Constructing Basic Algebras for the Principal Block of Sporadic Simple Groups

by

Thomas Rune Hoffman

A Dissertation Submitted to the Faculty of the

Department of Mathematics

In Partial Fulfillment of the Requirements
For the Degree of

Doctor of Philosophy

In the Graduate College

The University of Arizona

2 0 0 4

Get the official approval page
from the Graduate College
before your final defense.

Statement by Author

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Signed: _____

Acknowledgments

I would like to thank Larry Grove and Alex Ryba for reading this lengthy document and making suggestions, improving the final product. I also owe much thanks to Klaus Lux for struggling through the early versions of my work and helping me turn it into a comprehensible dissertation.

Finally, I thank my wife Molly J for all of her support over my six years at the University of Arizona and two years prior at Oregon State University. I know it wasn't always easy or fun, but she was always there for me. Thank you.

Table of Contents

TABLE OF CONTENTS—*Continued*

TABLE OF CONTENTS—*Continued*

List of Figures

List of Figures—*Continued*

List of Figures—*Continued*

List of Tables

Abstract

This dissertation describes an algorithm for constructing the basic algebra Morita equivalent to the principal block of certain group algebras. This algorithm uses the method of condensation as it is described in [Lux97]. Using an intermediate condensation subalgebra allows for the construction of the projective indecomposable modules required to realize the basic algebra.

The group algebras we are concerned with here are for sporadic groups in characteristic dividing the order of the group. In particular, the basic algebra for the principal block of the Higman-Sims group in characteristic 2 is completed and seven of the thirteen projective indecomposable modules for the Mathieu group M_{24} are constructed. In addition to these algebras, we have also computed the basic algebras for many alternating and symmetric groups.

Constructing Basic Algebras for the Principal Block of Sporadic Simple Groups

Thomas Rune Hoffman, Ph.D.
The University of Arizona, 2004

Director: Klaus Lux

This dissertation describes an algorithm for constructing the basic algebra Morita equivalent to the principal block of certain group algebras. This algorithm uses the method of condensation as it is described in [Lux97]. Using an intermediate condensation subalgebra allows for the construction of the projective indecomposable modules required to realize the basic algebra.

The group algebras we are concerned with here are for sporadic groups in characteristic dividing the order of the group. In particular, the basic algebra for the principal block of the Higman-Sims group in characteristic 2 is completed and seven of the thirteen projective indecomposable modules for the Mathieu group M_{24} are constructed. In addition to these algebras, we have also computed the basic algebras for many alternating and symmetric groups.

Introduction

The classification of finite simple groups stimulated the use of computations in the area of modular representation theory - working with groups as matrices over finite fields. Realizing finite simple groups as matrices was an important tool in showing the existence of previously unknown simple groups. This is seen in the construction by Z. Janko of the simple group, now denoted as J_1, as a subgroup of $\mathrm{GL}_{11}(7)$ in [Jan66]. Knowing all finite simple groups was required to complete the classification of finite simple groups.

After the completion of the classification of finite simple groups, efforts focused on investigating the 26 groups, called sporadic groups, which did not fit well into the classification. Unlike the alternating groups, for example, these groups don't have a set of unifying properties. The group theoretic properties of sporadic groups largely need to be explored on an individual basis. Computational methods were developed during the 1980's and 1990's for studying these groups and their representations. In this dissertation, we develop and implement an algorithm for computing the basic algebras Morita equivalent to the principal blocks of group algebras for sporadic groups.

The algorithm developed in this dissertation has two main steps. Firstly, we construct a subalgebra $eFGe$ of the group algebra FG. In general, the dimension of $eFGe$ is much smaller than the dimension of FG, making it more accessible to computer programs. The second step is to build generators of the basic algebra B for the principal block of FG as homomorphisms between projective indecomposable $eFGe$-modules.

The motivation for this dissertation is that for most sporadic groups G, the regular representation of FG is computationally inaccessible. Consider the Higman-Sims

group HS with order $44352000 = 2^9 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 11$. Storing a matrix for the regular representation of HS over $\mathbb{F}_2$ in *C-MeatAxe* format requires approximately 223 terabytes. Clearly, we can not work with the regular representation of HS. Fortunately, we can apply an observation of K. Morita to group algebras.

In the 1950's, Morita noticed that for an algebra A, there are algebras whose representations are categorically equivalent to the representations of A. Algebras whose representations are categorically equivalent are said to be Morita equivalent. The algebras Morita equivalent to A can be realized as $\mathrm{End}_A(P)$ where P is taken to have each projective indecomposable A-module, up to isomorphism, as a summand.

It appears that we need to know the projective indecomposable FG-modules in order to construct an algebra Morita equivalent to FG. This is not true. There are subalgebras of FG which are Morita equivalent to FG that can be constructed without realizing them as $\mathrm{End}_{FG}(P)$ for some projective FG-module P. Let H be a subgroup of G such that the characteristic of F does not divide the order of H and e the sum of the elements of H. If Se is not zero for all simple FG-modules S, $eFGe$ is Morita equivalent to FG. We show in this dissertation how to construct the regular representation of $eFGe$ using a permutation representation of G on the cosets of H.

In [Lux97], K. Lux used $eFGe$ to construct projective indecomposable FG-modules for several large sporadic groups. However, due to limitations of computers and the *C-MeatAxe*, he was restricted in which groups and characteristics he could consider. Recalling the example from above of HS, in characteristic 2, we take H to be a subgroup of order 21. The algebra $e\mathbb{F}_2HSe$ has dimension 100732, which is clearly much less than 44352000. In *C-MeatAxe* format, a matrix for an element of the regular representation of $e\mathbb{F}_2HSe$ requires approximately 1.2 gigabytes. This puts the regular representation of $e\mathbb{F}_2HSe$ out of the range of what Lux could consider in 1997 and is still outside the bounds of what would be reasonable to work with on a computer today. Fortunately, for $g \in G$, ege has only order of H nonzero entries per row. The author has implemented a row-sparse format in the *C-MeatAxe* to take advantage

of these special matrices. A matrix for $ege \in e\mathbb{F}_2HSe$ in row-sparse format requires only 8.5 megabytes. This improvement allows us to extend the results of Lux to both larger groups and more characteristics.

Now that the construction of the projective indecomposable $eFGe$-modules, up to isomorphism, is possible, we return to the original problem of building the basic algebra of FG. The basic algebra of FG is the smallest algebra Morita equivalent to FG. We realize the basic algebra Morita equivalent to $eFGe$ as $\mathrm{End}_{eFGe}(P)$, where P is a direct sum of projective indecomposable $eFGe$-modules such that, up to isomorphism, each projective indecomposable $eFGe$-module appears exactly once. Since e is chosen so that $eFGe$ is Morita equivalent to FG, $\mathrm{End}_{eFGe}(P)$ is also Morita equivalent to FG.

The implementation of the algorithm described in this dissertation has been used to construct basic algebras for several sporadic groups, including the Mathieu groups, except M_{24} in characteristic 2, the Janko groups, Higman-Sims group, McLaughlin group, except in characteristic 3, and the Held group in characteristic 7. While some of these group algebras have been studied, the basic algebra was not known for many of them, and for a few of them the projective indecomposable FG-modules were not known. Basic algebras for many alternating and symmetric groups have also been computed.

In the future, we will extend these results further in several ways. Firstly, we can improve the *C-MeatAxe* further by increasing the number of programs which work with row-sparse format matrices. Most importantly is the *C-MeatAxe* program `zmw` which makes words in generators for an algebra. Because of memory restrictions, the reduced size of row-sparse matrices would enable it to be used on larger representations. We would also like to add the functionality of applying a word to a vector without constructing the word to `zmw`. These improvements make constructing the basic algebras of M_{24} in characteristic 2 and McL in characteristic 3 feasible.

Another possibility for extending our results is to develop new methods for finding

the projective indecomposable $eFGe$-modules. Currently, our implementation constructs the projective indecomposable $eFGe$-modules as submodules of the regular representation of $eFGe$. However, for M_{24} in characteristic 2, we have found 7 of the 13 projective indecomposable $eFGe$-modules, up to isomorphism, as submodules of projective $eFGe$-modules corresponding to subgroups of M_{24} which are not the condensation subgroup H.

This dissertation is organized as follows.

Chapter 1 covers the background material necessary for understanding the terminology contained in this dissertation. Of particular importance are Sections 1.2 and 1.4: they contain the definitions of endomorphism rings, projective modules and radicals, all of which play major roles in this dissertation.

Chapter 2 covers the basics of Morita theory, including both the consequences of Morita equivalence and the necessary conditions to form a Morita equivalence. The second half focuses on the Morita equivalence of condensation subalgebras. We show that condensation subalgebras are Morita equivalent, followed by a discussion on finding an appropriate subgroup for condensation. The chapter ends with algorithms for building the matrix of an element in the condensation subalgebra and for finding a generating set for the condensation subalgebra.

Chapter 3 covers the construction of the basic algebra Morita equivalent to a group algebra. Basic algebras are discussed, as well as peakwords and blocks. An algorithm is given to construct basic algebras. At the end of this chapter, Ext-quivers are discussed which give another view of basic algebras.

Chapter 4 discusses the author's implementation of Algorithm 3.4.1 on page 87. This includes discussion of the programs the author has written in *GAP* as well as modifications to existing programs in the *C-MeatAxe*. Chapter 5 describes the results achieved by the implementation discussed in Chapter 4. Many Ext-quivers are displayed here.

Chapter 1
Background

The purpose of this chapter is fix the definitions and results which are used in this dissertation. Since the material covered here can be found in most books on representation theory of finite groups, we will only give proofs when they are instructive for the following, otherwise we just give references.

1.1 Algebras and Modules

The main objects of study in this dissertation are finite dimensional algebras and their representations. This section will give the basic definitions necessary for this study and can be found in references on representation theory of finite groups such as [CR62], [CR81], [Lux97], and [AF92].

We let $\mathbb{F}$ denote a finite field for the remainder of this dissertation. Also, all rings are associative.

Definition 1.1.1. *An algebra A over a field $\mathbb{F}$ is a ring A with an identity element, 1_A, which is at the same time an $\mathbb{F}$-vector space. Moreover, the scalar multiplication in the vector space and the ring multiplication are required to satisfy the axiom*

$$\alpha(ab) = (\alpha a)b = a(\alpha b),$$

for $\alpha \in \mathbb{F}$, and $a, b \in A$.

An algebra can be defined more generally over a ring instead of a field. We use Definition 1.1.1 since all of the algebras considered in this dissertation are over fields.

For a given algebra A, there is an algebra A^{op}, called the opposite algebra, which can be useful in the study of the representation theory of finite groups. As $\mathbb{F}$-vector spaces, $A = A^{op}$. We define multiplication, denoted by $*$, in A^{op} by $a_1 * a_2 = a_2 a_1 \in A$.

Definition 1.1.2. *Let A be an algebra with a subring B. If B is also an $\mathbb{F}$-vector subspace of A, we call B a subalgebra of A.*

Just as rings have special subrings called ideals, the following definition names the corresponding concept for algebras.

Definition 1.1.3. *Let A be an algebra over $\mathbb{F}$. A right ideal I in the algebra A is a subalgebra of A which is also a right ideal in the ring A. Left ideals are defined similarly. If I is both a right and left ideal in A, then we call it a two-sided ideal.*

In coming sections, we will see that ideals play an important role in the study of algebras. We notice that an algebra A always has at least two ideals since $\{0\}$ and A are both two-sided ideals in A. Algebras with only two two-sided ideals are addressed in the following definition.

Definition 1.1.4. *We call an algebra A simple if the only two-sided ideals in A are $\{0\}$ and A.*

Two-sided ideals are not the only ideals in an algebra with which we are concerned. The following definition is an example of the role right and left ideals play in the structure of an algebra.

Definition 1.1.5. *A descending chain $I_1 \supseteq I_2 \supseteq I_3 \supseteq \dots$ of right ideals in an algebra A is said to terminate if it is finite or if there is some index k such that $I_n = I_k$ for all $n \geq k$. An algebra for which every descending chain of right ideals terminates is called Artinian.*

Similarly an ascending chain $I_1 \subseteq I_2 \subseteq I_3 \subseteq \dots$ of right ideals in an algebra A is said to terminate if it is finite or if there is some index k such that $I_n = I_k$ for all $n \geq k$. An algebra for which every ascending chain of right ideals terminates is called Noetherian.

Proposition 1.1. *Finite dimensional algebras are both Artinian and Noetherian.*

Proof. The proof of this proposition follows from the fact that ideals in the finite dimensional $\mathbb{F}$-algebra A are $\mathbb{F}$-vector subspaces of A. □

For the rest of this dissertation, let A be a finite dimensional $\mathbb{F}$-algebra. Group algebras for finite groups are examples of finite dimensional algebras. Since group algebras play an important role in this dissertation, we describe them below.

***Example* 1.1.1.** *Let G be a finite group. Denote by $\mathbb{F}G$ the set of all formal sums $\sum_{g\in G}\alpha_g g$ with $\alpha_g \in \mathbb{F}$. With the obvious addition and scalar multiplication, we get that $\mathbb{F}G$ is a vector space over $\mathbb{F}$ with a basis corresponding to elements of G. With multiplication defined by*

$$\left(\sum_{x\in G}\alpha_x x\right)\left(\sum_{y\in G}\beta_y y\right) = \sum_{u\in G}\left(\sum_{x,y\in G:\, xy=u}\alpha_x\beta_y\right)u$$

for $\alpha_x, \beta_y \in \mathbb{F}$, $\mathbb{F}G$ is a ring. The ring homomorphism $\phi\colon \mathbb{F}\to\mathbb{F}G$ defined by $\phi(\alpha)\mapsto \alpha 1_G$ where 1_G is the identity in G, satisfies $\phi(1_{\mathbb{F}}) = 1_G$ and the image of ϕ is in the center of $\mathbb{F}G$. So $\mathbb{F}G$ is an $\mathbb{F}$-algebra which we call the group algebra of G over $\mathbb{F}$.

Now that we have covered the basic definitions for algebras, we move on to modules over algebras. The reader should notice in the next definition that we will primarily be using right modules.

Definition 1.1.6. *Let A be an algebra over $\mathbb{F}$. We say that M is a right A-module if it is an $\mathbb{F}$-vector space with a right action by A satisfying:*

1. $m\cdot(a_1a_2) = (m\cdot a_1)\cdot a_2$,
2. $m\cdot(a_1+a_2) = (m\cdot a_1)+(m\cdot a_2)$,
3. $(m_1+m_2)\cdot a = (m_1\cdot a)+(m_2\cdot a)$,
4. $m\cdot(1_A) = m$,
5. $\lambda(m\cdot a) = (\lambda m)\cdot a$,

for all $m, m_1, m_2 \in M$ and $a, a_1, a_2 \in A$. Left A-modules are defined similarly.

If M is a right A-module and a left B-module for two algebras A and B such that $b \cdot (m \cdot a) = (b \cdot m) \cdot a$ for all $b \in B$, $a \in A$ and $m \in M$, then we call M a B-A bimodule.

***Example* 1.1.2.** *An example of a right A-module which makes frequent appearances in representation theory is the $\mathbb{F}$-vector space A with right multiplication by the $\mathbb{F}$-algebra A. We call this A-module the right regular A-module and denote it by A_A. We similarly define the left regular A-module ${}_AA$, and the A-A bimodule ${}_AA_A$.*

The following definitions will classify a few types of modules with properties which are important in later chapters.

Definition 1.1.7. *Let M be an A-module and N be an $\mathbb{F}$-vector subspace of M. Then N is called an A-submodule of M if $na \in N$ for all $a \in A$ and $n \in N$.*

We suppress the prefix A and simply write N is a submodule of M when A is clear from the context.

Definition 1.1.8. *A descending chain $M_1 \geq M_2 \geq M_3 \geq \dots$ of submodules in an A-module is said to terminate if it is finite or if there is some index k such that $M_n = M_k$ for all $n \geq k$. An A-module for which every descending chain of submodules terminates, is called Artinian.*

Similarly, an ascending chain $M_1 \leq M_2 \leq M_3 \leq \dots$ of submodules in an A-module is said to terminate if it is finite or if there is some index k such that $M_n = M_k$ for all $n \geq k$. An A-module for which every ascending chain of submodules terminates is called Noetherian.

Let I be a right ideal in the algebra A. Then I is also a right A-module. In fact, the ideals of A are the submodules of A_A. With this observation, we see that A is Artinian (Noetherian) by Definition 1.1.5 if and only if A_A is Artinian (Noetherian) by

Definition 1.1.8. In particular, when A is finite dimensional, Proposition 1.1 implies that A_A is Artinian and Noetherian. The following lemma extends this idea to all finitely generated A-modules.

Lemma 1.2. *Let M be a finitely generated A-module. If A is Artinian (Noetherian), then so is M.*

Proof. A proof of this lemma is found in [Fei82] on page 6. □

Definition 1.1.9. *We call an A-module $M \neq \{0\}$ simple if the only right submodules of M are 0 and M.*

We call the algebra A simple if the A-module A_A is simple. This is equivalent to saying that the only right ideals of A are $\{0\}$ and A itself.

The reader will see why the trivial A-module is not allowed to be simple in the following definition.

Definition 1.1.10. *Let M be an A-module. A series*

$$0 = M_0 < M_1 < \cdots < M_n = M$$

of submodules of M is called a composition series for M if M_i/M_{i-1} is a simple A-module for $1 \leq i \leq n$. We call n the length of the composition series. The factor M_i/M_{i-1} is called a composition factor of M.

The previous definition brings up two questions, when does an A-module have a composition series and if it does, is the composition series unique. The next lemma answers the first question.

Lemma 1.3. *The A-module M has a composition series if and only if M is both Artinian and Noetherian.*

Proof. A proof of this lemma is found in [Fei82] on page 7. □

We stated earlier that we are only considering algebras A which are finite dimensional. By Proposition 1.1, A is both Artinian and Noetherian. So by Lemma 1.2, a finitely generated A-module M is also Artinian and Noetherian. Therefore, M has a composition series by Lemma 1.3. Now we consider the possibility that a composition series for M is unique. Instead of deciding the uniqueness of the series, we ask if the composition factors for M are unique. The Jordan-Hölder Theorem will answer this uniqueness question, but it will have to wait until the next section.

1.2 A-module Homomorphisms

Recall that the A-modules M and N are $\mathbb{F}$-vector spaces, so we can consider the $\mathbb{F}$-linear maps between M and N. We are more interested in the $\mathbb{F}$-linear maps which commute with the action of A on M and N. These maps are the A-homomorphisms and are defined as follows.

Definition 1.2.1. *Let M and N be A-modules. Then an $\mathbb{F}$-linear map $\phi\colon M \to N$ is an A-homomorphism if $\phi(m \cdot a) = \phi(m) \cdot a$ for all $m \in M$ and $a \in A$. We use $\mathrm{Hom}_A(M, N)$ to denote the $\mathbb{F}$-vector space of A-homomorphisms from M to N.*

The reader should notice that we are writing our homomorphisms on the left. While this may not seem crucial, it is important when we consider A-endomorphisms and composition.

We will use $\operatorname{Ker}\phi$ and $\operatorname{Im}\phi$ for the kernel and image of the A-homomorphism $\phi\colon M \to N$ without explanation. They are clearly submodules of M and N, respectively.

Definition 1.2.2. *Let M and N be A-modules and $\phi \in \mathrm{Hom}_A(M, N)$. We call ϕ an A-epimorphism if $\operatorname{Im}\phi = N$ and an A-monomorphism if $\operatorname{Ker}\phi = 0$. If ϕ is both an A-monomorphism and an A-epimorphism, then we call it an A-isomorphism.*

We drop the prefix A and simply write homomorphism, epimorphism, monomorphism, or isomorphism when A is clear from the context.

Definition 1.2.3. *Let $M_1, \ldots, M_n$ be A-modules with homomorphisms $f\colon M_1 \to M_2$ and $g\colon M_2 \to M_3$. If $\operatorname{Im} f = \operatorname{Ker} g$, then we call the sequence*

$$M_1 \xrightarrow{f} M_2 \xrightarrow{g} M_3$$

exact, or exact at M_2. If the sequence

$$M_1 \longrightarrow M_2 \longrightarrow \cdots \longrightarrow M_{n-1} \longrightarrow M_n$$

is exact at $M_2, \ldots, M_{n-1}$, then we call the sequence exact. If we have a sequence

$$0 \longrightarrow M_1 \xrightarrow{f} M_2 \xrightarrow{g} M_3 \longrightarrow 0$$

which is exact, then we call it a short exact sequence.

We notice that if we have a short exact sequence as in Definition 1.2.3, then f is a monomorphism and g is an epimorphism. This leads to writing a short exact sequence as

$$M_1 \xhookrightarrow{f} M_2 \xrightarrow{g} M_3$$

where $\hookrightarrow$ signifies that f is an injection and hence a monomorphism and $\twoheadrightarrow$ denotes that g is an epimorphism.

If there is an A-isomorphism from M to N, then we say that M and N are isomorphic as A-modules. This is the piece we needed to state the Jordan-Hölder Theorem and finish our discussion from the last section.

Theorem 1.4 (Jordan-Hölder)**.** *Let*

$$0 = M_0 < M_1 < \cdots < M_m = M$$

and

$$0 = N_0 < N_1 < \cdots < N_n = M$$

be two composition series for the A-module M. Then $m = n$ and there exists a permutation σ such that the composition factors M_i/M_{i-1} and $N_{\sigma(i)}/N_{\sigma(i)-1}$ are isomorphic as A-modules.

Proof. A proof of the Jordan-Hölder Theorem is found in [CR62] on page 79. □

When $\phi \in \mathrm{Hom}_A(M, N)$ and $M = N$, we call ϕ an A-endomorphism. We use $\mathrm{End}_A(M)$ to denote the $\mathbb{F}$-vector space $\mathrm{Hom}_A(M, M)$. Since the composition of two endomorphisms is again an endomorphism, $\mathrm{End}_A(M)$ is ring. With the ring homomorphism $\tau\colon \mathbb{F} \to \mathrm{End}_A(M)$ defined by $\tau(a) = a \cdot id_M$ for all $a \in \mathbb{F}$ and with id_M the identity endomorphism on M, $\mathrm{End}_A(M)$ is an algebra. For $\phi, \psi \in \mathrm{End}_A(M)$ and $m \in M$, we define $(\phi \circ \psi)(m) = \phi(\psi(m))$.

Proposition 1.5. *Let A be a finite dimensional $\mathbb{F}$-algebra. Then $A \cong \mathrm{End}_A(A_A)$ as $\mathbb{F}$-algebras.*

We include the proof of this proposition since in many treatments of this subject, $A^{op} \cong \mathrm{End}_A(A_A)$, and it should be pointed out that the notation and definitions we are using changes this result.

Proof. Let $\phi_a\colon A_A \to A_A$ by $m \mapsto a \cdot m$ for $a \in A$ and all $m \in A_A$ where $a \cdot m$ is multiplication in A. Clearly, ϕ_a is an $\mathbb{F}$-linear map, so we need to check that ϕ_a commutes with the action of A on A_A. Let $b \in A$, then by associativity of multiplication in A, we get $\phi_a(mb) = a\cdot(mb) = (a\cdot m)b = \phi_a(m)b$. So $\phi_a \in \mathrm{End}_A(A_A)$.

Let $f\colon A \to \mathrm{End}_A(A_A)$ by $a \mapsto \phi_a$. To prove f is $\mathbb{F}$-linear, we consider

$$f(ca + b) = \phi_{ca+b}$$

for all $c \in \mathbb{F}$ and $a, b \in A$. So,

$$\begin{aligned} f(ca+b)(m) =& \phi_{ca+b}(m) \\ =& (ca+b) \cdot m \\ =& (ca) \cdot m + b \cdot m \\ =& c(a \cdot m) + b \cdot m \\ =& c\phi_a(m) + \phi_b(m) \\ =& cf(a)(m) + f(b)(m), \end{aligned}$$

for all $m \in A_A$. Lastly, to show f is a ring isomorphism,

$$\begin{aligned} f(a \cdot b)(m) =& \phi_{a \cdot b}(m) \\ =& (a \cdot b) \cdot m \\ =& a \cdot (b \cdot m) \\ =& (\phi_a \circ \phi_b)(m) \\ =& (f(a) \circ f(b))(m). \end{aligned}$$

Therefore, $A \cong \mathrm{End}_A(A_A)$ as $\mathbb{F}$-algebras. □

1.3 Structure Theorems

Since the A-module M is an $\mathbb{F}$-vector space, we can write it as a sum of vector subspaces. We are interested in expressing M as a sum of submodules, not just $\mathbb{F}$-subspaces. This decomposition of M is tied to the structure of A. In this section we cover the basic definitions and results related to the structure of A and M.

The first definition of this allows us to realize an A-module as a sum of submodules.

Definition 1.3.1. *An A-module M is said to be an internal direct sum $M_1 \oplus \cdots \oplus M_r$ for r a positive integer with submodules $M_1, \cdots, M_r$ of M if the following conditions are satisfied:*

1. *As* $\mathbb{F}$*-vector spaces,* $\mathrm{Span}_{\mathbb{F}}(M_1, \cdots, M_r) = M$.

2. *For* $1 \leq i \leq r$, $M_i \cap (M_1 + \cdots + M_{i-1} + M_{i+1} + \cdots + M_r) = 0$.

We call $M_1 \oplus \cdots \oplus M_r$ *a decomposition of* M. *The submodules* M_i *are called direct summands of* M. *In the case* $r > 1$ *and* $M_i \neq \{0\}$ *for all* $i \in \{1, 2, \ldots, r\}$, *we call* M *decomposable. If there is no decomposition of* M *with* $r > 1$ *and* $M_i \neq \{0\}$ *for all* $i \in \{1, 2, \ldots, r\}$, *we call* M *indecomposable.*

While the next definition is similar to the previous, we use it to construct an A-module out of other A-modules.

Definition 1.3.2. *Let* $M_1, \ldots, M_r$ *be* A*-modules for some positive integer* r. *We define the external direct sum* $M = M_1 \dot{+} \cdots \dot{+} M_r$ *to be* r*-tuples where the* i^{th} *entry is in* M_i *for* $1 \leq i \leq r$. *With* A *acting component-wise,* M *is an* A*-module. We write* $M_i \mid M$ *to indicate that* M_i *is isomorphic to a direct summand of* M.

It is possible for an A-module M to have a submodule N such that N is not a direct summand but $N \mid M$. This distinction is made clear in the following example from [Sző98], page 6.

***Example* 1.3.1.** *Consider the group algebra* $\mathbb{F}_2C_2$ *of the cyclic group* $C_2 =< g >$ *of order* 2 *over* $\mathbb{F}_2$, *the field with* 2 *elements. Let* M *be a three dimensional vector space over* $\mathbb{F}_2$ *with basis* v_1, v_2, v_3. *The vector space* M *is an* $\mathbb{F}_2C_2$*-module with the action of* g *on* M *given by the matrix*

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then $M =< v_1, v_2 > \oplus < v_3 >$ *is a decomposition of the* $\mathbb{F}_2C_2$*-module* M. *Let* S *be the submodule of* M *generated by* v_2. *Then* S *is isomorphic to the submodule generated by* v_3, *so* $S \mid M$. *But,* S *is not a direct summand of* M *since* $v_1 \notin S$ *and* S *is contained in every submodule containing* v_1.

Lemma 1.6 (Schur's Lemma)**.** *Let M_1 and M_2 be simple A-modules. If $M_1 \ncong M_2$, then $\mathrm{Hom}_A(M_1, M_2) = 0$, and $\mathrm{Hom}_A(M_1, M_1) = \mathrm{End}_A(M_1)$ is a division ring.*

Proof. A straightforward proof is found in [Gro83] on page 173. □

Before we can continue discussing the structure of A, we need to introduce some special ideals in A which will play an important role.

Definition 1.3.3. *Let A be an algebra and M an A-module. We define the annihilator of m, $\mathrm{Ann}_A(m)$, for some $m \in M$ to be the set of all $a \in A$ such that $ma = 0$. The annihilator of M, $\mathrm{Ann}_A(M)$, is the intersection of the annihilators of all the elements of M.*

From the definition we see that $\mathrm{Ann}_A(m)$ is a right ideal in A, which is maximal if and only if the A-module generated by m is simple. It is also clear that $\mathrm{Ann}_A(M)$ is a two-sided ideal in A.

Definition 1.3.4. *If $\mathrm{Ann}_A(M) = \{0\}$ for an A-module M, then we call M a faithful A-module.*

We have seen that annihilators for A-modules are ideals in A. The following definition considers a particular ideal in A.

Definition 1.3.5. *Let A be an algebra. Then we define $\mathrm{J}(A)$, the Jacobson radical of A, to be the intersection of the annihilators of all simple A-modules.*

Since the annihilator of a simple A-module is a maximal two-sided ideal in A, $\mathrm{J}(A)$ is the intersection of maximal two-sided ideals of A.

We define $\mathrm{J}^n(A)$ as $\mathrm{J}(\mathrm{J}^{n-1}(A))$.

The following lemma proves to be very important. At first glance, it may not be clear how it is useful. But Lemma 1.7 tells us where we should look to find generators of an algebra A. This lemma is crucial in the proof of Theorem 3.6.

Lemma 1.7. *Suppose A is an Artinian algebra and A' is a subalgebra of A such that $A' + \mathrm{J}^2(A) = A$. Then $A' = A$.*

Proof. A proof of this lemma can be found in [Ben98] on page 5. □

An important class of algebras A have the property that $J(A) = 0$. We call these algebras semisimple.

The following theorem associates with an A-module M a semisimple algebra E, for which M is also a module, such that the actions of A and E on M commute.

Theorem 1.8. *Let M be a A-module such that $M = M_1 \oplus \cdots \oplus M_r$ where for each $1 \leq i \leq r$, M_i is the direct sum of n_i A-modules isomorphic to the simple A-module S_i and $S_i \ncong S_j$ when $i \neq j$. Let $D_i = \mathrm{End}_A(S_i)$. Then D_i is a division ring, $\mathrm{End}_A(M_i) \cong Mat_{n_i}(D_i)$, and $E = \mathrm{End}_A(M) \cong \dot{+}_{i=1}^{r} \mathrm{End}_A(M_i)$ is semisimple.*

Proof. A proof of Theorem 1.8 is found in [Ben98] on page 6. □

Denote by $\mathrm{Mat}_n(R)$ the algebra of $n \times n$ matrices over the ring R.

The following theorem shows that a semisimple algebra is a sum of simple algebras and the simple summands are isomorphic to matrix algebras.

Theorem 1.9 (Wedderburn-Artin Structure Theorem)**.** *Let A be a semisimple algebra with r isomorphism classes of simple modules S_i, with $i = 1, \ldots, r$. Then $A \cong \dot{+}_{i=1}^{r} Mat_{n_i}(D_i)$, where D_i is a division ring such that $D_i \cong \mathrm{End}_A(S_i)$ and $n_i = \dim_{D_i}(S_i)$. If A is simple then $A \cong Mat_n(D)$.*

Proof. A proof of the Wedderburn-Artin Structure Theorem is found in [Ben98] on page 6. □

Since semisimple algebras decompose into matrix algebras, questions about them can generally be answered using linear algebra. The following theorem characterizes semisimple group algebras.

Theorem 1.10 (Maschke)**.** *The group algebra $\mathbb{F}G$ is semisimple if and only if the characteristic of $\mathbb{F}$ does not divide the order of G.*

Proof. A proof of Maschke's Theorem is found in [Car96] on page 4. □

Maschke's Theorem points out that at least for group algebras, the characteristic of the field plays a large role in the structure of the algebra. Because of this, we briefly consider which fields we will use for this dissertation. The following theorem helps us understand the division rings D_i appearing in the Wedderburn-Artin Structure Theorem (1.9).

Theorem 1.11 (Wedderburn)**.** *Every finite division ring is a field.*

Proof. A proof of this theorem can be found in [Hun87] on page 462. □

In the setting of Theorem 1.9, the field $\mathbb{F}$ is contained in the center of the division rings D_i since $\mathbb{F}$ commutes with A. If $\mathbb{F}$ is finite, and A is finite dimensional, then D_i is a finite division ring containing $\mathbb{F}$. Thus, by Theorem 1.11, D_i is a finite extension of $\mathbb{F}$. The following definition characterizes the case where $D_i = \mathbb{F}$.

Definition 1.3.6. *Let A be an algebra over the field $\mathbb{F}$. We call $\mathbb{F}$ a splitting field for A if for every simple A-module S,* $\mathrm{End}_A(S) \cong \mathbb{F}$.

For the rest of this dissertation, we assume that $\mathbb{F}$ is a splitting field of A.

Definition 1.3.7. *An A-module M is said to have the unique decomposition property if*

1. *M is a finite direct sum of indecomposable A-modules, and*
2. *when $M = \oplus_{i=1}^{r} M_i = \oplus_{i=1}^{k} M_i'$ with each M_i and M_i' indecomposable and nonzero, $r = k$ and, possibly after reordering, $M_i \cong M_i'$.*

An algebra A is said to have the unique decomposition property if every finitely generated A-module has the unique decomposition property.

Theorem 1.12 (Krull-Schmidt)**.** *Suppose that A is Artinian, then A has the unique decomposition property.*

Proof. A proof of the Krull-Schmidt Theorem can be found in [Ben98] on page 8. □

The unique decomposition property brings up the question of how many non-isomorphic indecomposable A-modules exist for a given algebra A. This is a hard question, but if A is finite dimensional we can answer the related question about isomorphism classes of simple A-modules.

Proposition 1.13. *Let A be a finite dimensional $\mathbb{F}$-algebra. Then a complete set of representatives of the isomorphism classes of simple A-modules is finite.*

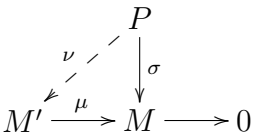
Proof. Since ideals of A are also A-modules, we have that every simple A-module is image of an A-homomorphism on the right regular A-module, A_A. Therefore the sum of the dimensions over $\mathbb{F}$ of the representatives must be less than the dimension of A_A over $\mathbb{F}$. Since the dimension of a representative is greater than or equal to 1, the set of representatives is finite. □

1.4 Projective Modules and Radicals

In this section we discuss projective modules. These modules will play a major role in the following chapters because of their properties described here.

Definition 1.4.1. *We call an A-module P projective if for all A-modules M, M' with an A-homomorphism $\sigma\colon P \to M$ and an A-epimorphism $\mu\colon M' \to M$, then there*

exists an A-homomorphism $\nu\colon P \to M'$ such that the following diagram commutes.



While this definition seems a little complex, the following lemma gives some equivalent characterizations of projective modules which will prove useful.

Lemma 1.14. *Let P and M be A-modules. Then the following are equivalent.*

1. *P is projective.*

2. *Every A-epimorphism $\tau\colon M \to P$ splits, i.e., P is isomorphic to a direct summand of M.*

3. *P is a direct summand of a free A-module.*

Proof. A proof of this lemma can be found in [Hun87] on pages 192-193. □

For each A-module, we can associate a particular projective A-module to it, but first we need to define a special type of A-epimorphism.

Definition 1.4.2. *Let M, N be A-modules and f an A-epimorphism in $\mathrm{Hom}_A(M, N)$. If for each sequence*

$$X \xrightarrow{g} M \xrightarrow{f} N$$

where X is an A-module and $g \in \mathrm{Hom}_A(X, M)$, where fg is an A-epimorphism, then g is also an A-epimorphism, we call f essential.

Definition 1.4.3. *Let M be an A-module. We call a projective A-module P a projective cover of M if there is an essential homomorphism from P to M.*

Comparing Definitions 1.4.1 and 1.4.3, we see that the essential homomorphism ensures that P both covers M, since it is an epimorphism, and that P is in some sense minimal.

Lemma 1.15. *The finitely-generated A-module M has a projective cover P which is unique up to isomorphism.*

Proof. This lemma is proven in [CR81], the uniqueness on page 131, and the existence on page 132. □

Definition 1.4.4. *The radical of an A-module M, denoted by* $\mathrm{Rad}(M)$*, is the intersection of all maximal A-submodules of M.*

The following lemma relates this definition of radicals for A-modules to the Jacobson radical of A, $\mathrm{J}(A)$, from Definition 1.3.5.

Lemma 1.16. *Let A be a finite dimensional algebra. If M is a finitely generated A-module then*

$$M\,\mathrm{J}(A) = \mathrm{Rad}(M).$$

Proof. A proof of this lemma is found in [Ben98] on page 4. □

Using the definition of radical of an A-module given above, we can define an important series of submodules of M.

Definition 1.4.5. *We recursively define the Loewy series, or radical series, of an A-module M as* $\mathrm{Rad}^0(M) = M$*, and* $\mathrm{Rad}^n(M) = \mathrm{Rad}(\mathrm{Rad}^{n-1}(M))$*. Let M be an A-module. Then the n^{th} Loewy layer, or radical layer, of the Loewy series of M is* $\mathrm{Rad}^{n-1}(M)/\,\mathrm{Rad}^n(M)$.

The first Loewy layer of an A-module M is also called the head of M.

Note that for an A-module M, $M/\,\mathrm{Rad}(M)$ is semisimple. Since $\mathrm{Rad}(M)$ is again an A-module, $\mathrm{Rad}^i(M)/\,\mathrm{Rad}^{i+1}(M)$ is semisimple for every positive integer i.

1.5 Idempotents

This section introduces special elements of algebras called idempotents. Idempotents with particular properties will also be discussed.

Definition 1.5.1. *We call a nonzero element e of an algebra A an idempotent if*

$$e^2 = e.$$

We notice that every algebra A has at least one idempotent, namely 1_A.

The following proposition describes the ineraction of the radical of an algebra and an idempotent of the algebra.

Proposition 1.17. *Let $e \in A$ be an idempotent. Then*

$$\mathrm{Rad}(eAe) = e\,\mathrm{J}(A)e.$$

Proof. A proof of this proposition is found in [CR81] on page 108. □

If $e \neq 1_A$ is an idempotent of A, then $1_A - e$ is an idempotent, since

$$(1_A - e)^2 = 1_A - 2e + e^2 = 1_A - e.$$

The idempotents e and $(1_A - e)$ have the property that

$$e(1_A - e) = (1_A - e)e = 0.$$

We define this property for arbitrary idempotents in the following definition.

Definition 1.5.2. *Two idempotents e_1 and e_2 in A are called orthogonal if*

$$e_1e_2 = e_2e_1 = 0.$$

Definition 1.5.3. *We call an idempotent $e \in A$ primitive if e can not be expressed as the sum of two orthogonal idempotents.*

Primitive idempotents play an important in the study of algebras since they correspond with indecomposable ideals as in the following proposition.

Proposition 1.18. *An idempotent $e \in A$ is primitive if and only if eA is an indecomposable right ideal of A.*

Proof. A proof of this proposition can be found in [CR81] on page 119. □

We call a set of orthogonal primitive idempotents in A, $\{e_1, \ldots, e_s\}$ for a positive integer s, complete if $1_A = e_1 + \cdots + e_s$. Complete sets of primitive idempotents are useful since, with Proposition 1.18, they give a decomposition of A into indecomposable right ideals by

$$1_A A = e_1 A + \cdots + e_s A.$$

If e is in the center of A, we say it is central. A centrally primitive idempotent is a central idempotent which can not be written as the sum of two nonzero central orthogonal idempotents. Extending the proposition above we see that if $e \in A$ is a centrally primitive idempotent, then eA is an indecomposable two-sided ideal.

Theorem 1.19. *If $e \in A$ is an idempotent and M is an A-module, then*

(*a*) $\mathrm{Hom}_A(eA, M) \cong Me$ *as $\mathbb{F}$-vector spaces, and*

(*b*) $\mathrm{End}_A(eA) \cong eAe$ *as $\mathbb{F}$-algebras.*

The following proof of this theorem is along the lines of the one found in [Pah03] on page 55. The importance of including this proof here is that our result is $\mathrm{End}_A(eA) \cong eAe$, instead of $\mathrm{End}_A(eA) \cong (eAe)^{op}$.

Proof. (a) An isomorphism from $\mathrm{Hom}_A(eA, M)$ to Me is given by

$$\phi \mapsto \phi(e) = \phi(e^2) = \phi(e)e \in Me$$

for $\phi \in \mathrm{Hom}_A(eA, M)$, with inverse given by

$$me \mapsto \phi_{me}$$

with

$$\phi_{me}(ea) = meea = mea$$

for $a \in A$ and $m \in M$.

(b) The same map given in (a) gives an isomorphism of rings when $M = eA$, since

$$(\phi \circ \psi)(e) = \phi(\psi(e)) = \phi(e\psi(e)) = \phi(e)\psi(e)$$

for $\phi, \psi \in \mathrm{End}_A(eA)$.

□

We can strengthen part (a) of Theorem 1.19 by defining an eAe-module structure on $\mathrm{Hom}_A(eA, M)$. We do this using part (b) to realize eAe as $\mathrm{End}_A(eA)$. For $\phi \in \mathrm{Hom}_A(eA, M)$ and $\psi \in \mathrm{End}_A(eA)$, we define the right action of $\mathrm{End}_A(eA)$ on $\mathrm{Hom}_A(eA, M)$ by $\phi \cdot \psi = \phi \circ \psi$. Checking Definition 1.1.6, we see that $\mathrm{Hom}_A(eA, M)$ is a right $\mathrm{End}_A(eA)$-module and hence a right eAe-module.

Corollary 1.20. *If $e \in A$ is an idempotent and M is an A-module, then*

$$\mathrm{Hom}_A(eA, M) \cong Me$$

as eAe-modules.

Proof. This result follows from using part (b) of Theorem 1.19 with the isomorphism given in the proof of part (a) of Theorem 1.19. □

1.6 Category Theory

In this section we define the basic ideas of category theory which for the basis form the discussion of Morita equivalence in Chapter 2. An in-depth discussion of this material can be found in [HS97].

Definition 1.6.1. *A category* $\mathfrak{C}$ *has three pieces of data:*

1. *a class of objects* $\mathrm{Obj}(\mathfrak{C})$,

2. *for each pair* $M, N \in \mathrm{Obj}(\mathfrak{C})$, *a set* $\mathfrak{C}(M, N)$ *of morphisms from* M *to* N,

3. *for each triple* $M_1, M_2, M_3 \in \mathrm{Obj}(\mathfrak{C})$, *a law of composition*

$$\mathfrak{C}(M_1, M_2) \times \mathfrak{C}(M_2, M_3) \to \mathfrak{C}(M_1, M_3),$$

which satisfy the following axioms:

A 1. *the sets* $\mathfrak{C}(M_1, N_1)$ *and* $\mathfrak{C}(M_2, N_2)$ *are disjoint unless* $M_1 = M_2$ *and* $N_1 = N_2$,

A 2. *the morphisms* $f \in \mathfrak{C}(M_1, M_2)$, $g \in \mathfrak{C}(M_2, M_3)$, *and* $h \in \mathfrak{C}(M_3, M_4)$ *satisfy the associative law of composition, i.e.,*

$$h(gf) = (hg)f,$$

A 3. *there is a morphism* $1_M \colon M \to M$ *such that, for any* $f \colon M \to N_1$, $g \colon N_2 \to M$,

$$f1_M = f$$

and

$$1_M g = g,$$

where $M, M_1, \ldots, M_4, N_1, N_2$ *are all in* $\mathrm{Obj}(\mathfrak{C})$.

The following example describes the categories that we will be concerned with in this dissertation.

***Example* 1.6.1.** *For an algebra* A, *we denote the category of finitely generated right* A*-modules as* Mod_A. *The objects we take are finitely generated* A*-modules,* M, N, *and the morphisms are* A*-homomorphisms,* $\mathrm{Hom}_A(M, N)$. *For the law of composition,*

$$(f \circ g)(m) = f(g(m))$$

for A-modules M_1, M_2, M_3, $m \in M_1$, and $f \in \mathrm{Hom}_A(M_1, M_2)$, $g \in \mathrm{Hom}_A(M_2, M_3)$.

We similarly define the category of finitely generated left A-modules and denote it by ${}_A\mathrm{Mod}$. This is the same as the category of finitely generated right A^{op}-modules since every left A-module is a right A^{op}-module.

Now that we have introduced categories, we want to consider relationships between them. To study relationships between objects, we need maps between them. The following definition gives us transformations between categories.

Definition 1.6.2. *Let $\mathfrak{C}$ and $\mathfrak{D}$ be categories. A functor $F\colon \mathfrak{C} \to \mathfrak{D}$ is a rule which assigns to each object $M \in \mathrm{Obj}(\mathfrak{C})$ an object $FM \in \mathrm{Obj}(\mathfrak{D})$ and to each morphism $f \in \mathfrak{C}(M, N)$ a morphism $Ff \in \mathfrak{D}(FM, FN)$, such that*

$$F(fg) = (Ff)(Fg),$$

for $M, N, O \in \mathrm{Obj}(\mathfrak{C})$, $f \in \mathfrak{C}(M, N)$ and $g \in \mathfrak{C}(O, M)$, and

$$F1_M = 1_{FM}.$$

***Example* 1.6.2.** *For each category $\mathfrak{C}$, we have a functor $I\colon \mathfrak{C} \to \mathfrak{C}$ defined by $IM = M$ for every object $M \in \mathfrak{C}$ and $If = f$ for every morphism $f \in \mathfrak{C}(M, N)$ and every pair of objects $M, N \in \mathfrak{C}$. We call I the identity functor on $\mathfrak{C}$.*

Let A and B be algebras, M an A-B bimodule and N a B-A bimodule. Also, let $V, W \in \mathrm{Mod}_A$ and $\alpha \in \mathrm{Hom}_A(V, W)$.

1. *The tensor functor $- \otimes_A M\colon \mathrm{Mod}_A \to \mathrm{Mod}_B$ is defined by*

$$V \mapsto V \otimes_A M$$

and

$$\alpha \mapsto \alpha_{-\otimes M}$$

where $\alpha_{-\otimes M}\colon V \otimes_A M \to W \otimes_A M$ is defined by

$$v \otimes_A m \mapsto \alpha(v) \otimes_A m$$

for $m \in M$ and $v \in V$.

2. *The* Hom *functor* $\mathrm{Hom}_A(N, -)\colon \mathrm{Mod}_A \to \mathrm{Mod}_B$ *is defined by*

$$V \mapsto \mathrm{Hom}_A(N, V)$$

and

$$\alpha \mapsto \alpha_*$$

where $\alpha_*\colon \mathrm{Hom}_A(N, V) \to \mathrm{Hom}_A(N, W)$ *is defined by*

$$\beta \mapsto \alpha\beta$$

for $\beta \in \mathrm{Hom}_A(N, V)$.

Definition 1.6.3. *Let A and B be algebras. The functor* $F\colon \mathrm{Mod}_A \to \mathrm{Mod}_B$ *is called left exact if for every exact sequence of A-modules,*

$$0 \longrightarrow M_1 \xrightarrow{f} M_2 \xrightarrow{g} M_3$$

the sequence

$$0 \longrightarrow FM_1 \xrightarrow{Ff} FM_2 \xrightarrow{Fg} FM_3$$

is exact.

***Example* 1.6.3.** *The* $\mathrm{Hom}_A(N, -)$ *functor defined in Example* 1.6.2 *is a left exact functor. A proof of this is found in* [*HS97*] *on page* 17.

Proposition 1.21. *Let*

$$0 \longrightarrow M_1 \xrightarrow{f} M_2 \xrightarrow{g} M_3 \longrightarrow 0$$

be a short exact sequence of A-modules with monomorphism f and epimorphism g. If P is a projective A-module then the sequence

$$0 \longrightarrow \mathrm{Hom}_A(P, M_1) \xrightarrow{f_*} \mathrm{Hom}_A(P, M_2) \xrightarrow{g_*} \mathrm{Hom}_A(P, M_3) \longrightarrow 0$$

is a short exact sequence. So the functor $\mathrm{Hom}_A(P, -)$ *is exact.*

Proof. Since $\mathrm{Hom}_A(P,-)$ is a left exact functor, we just need to show that g_* is surjective. Consider the diagram

$$\begin{array}{ccccccccc} & & & & & & P & & \\ & & & & & \swarrow^{\phi} & \downarrow{\scriptstyle \psi} & & \\ 0 & \longrightarrow & M_1 & \xrightarrow{f} & M_2 & \xrightarrow{g} & M_3 & \longrightarrow & 0 \end{array}$$

where $\psi \in \mathrm{Hom}_A(P, M_3)$. The homomorphism $\phi\colon P \to M_2$ exists and $g\phi = \psi$ since P is projective. Therefore, $\psi \in \mathrm{Im}(g_*)$ for all $\psi \in \mathrm{Hom}_A(P, M_3)$ proving that g_* is an epimorphism. □

Definition 1.6.4. *Let F and G be functors from the category $\mathfrak{C}$ to the category $\mathfrak{D}$. Then a natural transformation t from F to G is a rule assigning to each object $M \in \mathfrak{C}$ a morphism $t_M\colon FM \to GM$ in $\mathfrak{D}$ such that for any morphism $f \in \mathfrak{C}(M,N)$, the diagram*

$$\begin{array}{ccc} FM & \xrightarrow{t_M} & GM \\ {\scriptstyle Ff}\downarrow & & \downarrow{\scriptstyle Gf} \\ FN & \xrightarrow{t_N} & GN \end{array}$$

commutes. If t_M is an isomorphism for every $M \in \mathfrak{C}$, then t is called a natural equivalence, and the functors F and G are said to be naturally equivalent.

Definition 1.6.5. *Let $\mathfrak{C}$ and $\mathfrak{D}$ be two categories. We call $\mathfrak{C}$ and $\mathfrak{D}$ equivalent if there exist functors*

$$F\colon \mathfrak{C} \to \mathfrak{D}$$

and

$$G\colon \mathfrak{D} \to \mathfrak{C}$$

such that $(F \circ G)$ and $(G \circ F)$ are naturally equivalent to the identity functors of $\mathfrak{D}$ and $\mathfrak{C}$, respectively.

***Example* 1.6.4.** *Consider the categories of finite dimensional $\mathbb{F}$-row spaces and finite dimensional $\mathbb{F}$-column spaces. The functor corresponding to the transpose operation forms an equivalence between these two categories.*

There are objects in module categories which are important in the following chapters. These objects will be used to define categorical equivalences. We name these objects in the following definition.

Definition 1.6.6. *For a module category Mod_A for the algebra A, an element $P \in \mathrm{Mod}_A$ is called a progenerator if it satisfies the following:*

1. *P is a projective A-module,*

2. *every A-module is a homomorphic image of a direct sum of copies of P.*

It is important to notice that when A is a finite dimensional algebra, then Mod_A has a progenerator, namely A_A.

1.7 Homological Algebra

In Chapter 3, we will be constructing basic algebras. Some applications of this process are the computation of projective resolutions, group cohomology and Ext-groups. These applications are all based in homological algebra, which we introduce in this section for completeness. This material can be found in [HS97], [Car96], and [Ben98]. Some proofs have been included to aid understanding of the material.

Definition 1.7.1. *A chain complex $\mathbf{D}$ over the algebra A is a collection of right A-modules D_n indexed by integers with homomorphisms $\partial_n \colon D_n \to D_{n-1}$ satisfying $\partial_n \circ \partial_{n+1} = 0$.*

A cochain complex $\mathbf{C}$ over A is a collection of right A-modules C^n indexed by integers with homomorphisms $\delta^n \colon C^n \to C^{n+1}$ satisfying $\delta^n \circ \delta^{n-1} = 0$.

Definition 1.7.2. *Let $\mathbf{C}$ and $\mathbf{D}$ be chain complexes. A chain map $\mathbf{f} \colon \mathbf{C} \to \mathbf{D}$ consists*

of A-module homomorphisms $f_n \colon C_n \to D_n$, $n \in \mathbb{Z}$, *such that the diagram*

$$\begin{array}{ccc} C_n & \xrightarrow{\partial_n} & C_{n-1} \\ \downarrow{\scriptstyle f_n} & & \downarrow{\scriptstyle f_{n-1}} \\ D_n & \xrightarrow{\partial_n} & D_{n-1} \end{array}$$

commutes.

Let $\mathbf{C}$ *and* $\mathbf{D}$ *be cochain complexes. A cochain map* $\mathbf{f} \colon \mathbf{C} \to \mathbf{D}$ *consists of A-module homomorphisms* $f_n \colon C^n \to D^n$, *with* $n \in \mathbb{Z}$, *such that the diagram*

$$\begin{array}{ccc} C^n & \xrightarrow{\delta^n} & C^{n+1} \\ \downarrow{\scriptstyle f_n} & & \downarrow{\scriptstyle f_{n+1}} \\ D^n & \xrightarrow{\delta^n} & D^{n+1} \end{array}$$

commutes.

Definition 1.7.3. *Let* $\mathbf{C}$ *and* $\mathbf{D}$ *be cochain complexes with cochain maps* $\mathbf{f}, \mathbf{g} \colon \mathbf{C} \to \mathbf{D}$. *We call* $\mathbf{f}$ *and* $\mathbf{g}$ *cochain homotopic if there exist homomorphisms* $h_n \colon C^n \to D^{n-1}$, *with* $n \in \mathbb{Z}$, *such that* $f_n - g_n = \delta^{n-1} \circ h_n + h_{n+1} \circ \delta^n$.

$$\begin{array}{ccccccccc} \cdots & \longrightarrow & C^{n-1} & \xrightarrow{\delta^{n-1}} & C^n & \xrightarrow{\delta^n} & C^{n+1} & \longrightarrow & \cdots \\ & & \downarrow{\scriptstyle f_{n-1}} \downarrow{\scriptstyle g_{n-1}} & \swarrow{\scriptstyle h_n} & \downarrow{\scriptstyle f_n} \downarrow{\scriptstyle g_n} & \swarrow{\scriptstyle h_{n+1}} & \downarrow{\scriptstyle f_{n+1}} \downarrow{\scriptstyle g_{n+1}} & & \\ \cdots & \longrightarrow & D^{n-1} & \xrightarrow{\delta^{n-1}} & D^n & \xrightarrow{\delta^n} & D^{n+1} & \longrightarrow & \cdots \end{array}$$

The cochain complexes $\mathbf{C}$ *and* $\mathbf{D}$ *are called cochain homotopy equivalent if there are cochain maps* $\mathbf{f} \colon \mathbf{C} \to \mathbf{D}$ *and* $\mathbf{g} \colon \mathbf{D} \to \mathbf{C}$ *such that* $\mathbf{f} \circ \mathbf{g}$ *and* $\mathbf{g} \circ \mathbf{f}$ *are cochain homotopic to the cochain maps* $\mathbf{id_D}$ *and* $\mathbf{id_C}$, *respectively.*

We notice that cochain homotopy equivalent is an equivalence relation on cochain complexes.

Definition 1.7.4. *Let* $x \in C^n$ *with* $\delta^n(x) = 0$, *then we call* x *a cocycle. We define* $Z^n(\mathbf{C})$ *as the kernel of* δ^n.

Let $y \in C^{n-1}$ and $x = \delta^{n-1}(y)$, then we call x a coboundary. We define $B^n(\mathbf{C})$ as the image of δ^{n-1}.

Definition 1.7.5. *The cohomology of a cochain complex $\mathbf{C}$ is defined by*

$$H^n(\mathbf{C}) = H^n(\mathbf{C}, \delta^\bullet) = \frac{Ker(\delta^n \colon C^n \to C^{n+1})}{Im(\delta^{n-1} \colon C^{n-1} \to C^n)} = \frac{Z^n(\mathbf{C})}{B^n(\mathbf{C})}.$$

The A-module $H^n(\mathbf{C})$ is called the n^{th} cohomology module.

A cochain map $\mathbf{f} \colon \mathbf{C} \to \mathbf{D}$ induces a well defined map $\mathbf{f}^* \colon H^n(\mathbf{C}) \to H^n(\mathbf{D})$ defined by $\mathbf{f}^*([x]) \mapsto [\mathbf{f}(x)]$ for $x \in Z^n(\mathbf{C})$.

Proposition 1.22. *Let $\mathbf{C}$ and $\mathbf{D}$ be cochain complexes. If $\mathbf{f}, \mathbf{g} \colon \mathbf{C} \to \mathbf{D}$ are cochain homotopic then $\mathbf{f}^* = \mathbf{g}^* \colon H^n(\mathbf{C}) \to H^n(\mathbf{D})$. Thus a cochain homotopy equivalence of $\mathbf{C}$ and $\mathbf{D}$ induces isomorphisms $H^n(\mathbf{C}) \cong H^n(\mathbf{D})$ for all $n \in \mathbb{Z}$.*

Proof. Let x be a cocycle, so $x \in C^n$ for some $n \in \mathbb{Z}$ and $\delta^n(x) = 0$. Then

$$\mathbf{f}^*[x] - \mathbf{g}^*[x] = [\mathbf{f}(x)] - [\mathbf{g}(x)] = [f^n(x) - g^n(x)]$$

from the definition of $\mathbf{f}^*$ and $\mathbf{g}^*$, and

$$[f^n(x) - g^n(x)] = [\delta^{n-1} \circ h_n(x) + h_{n+1} \circ \delta^n(x)]$$

by the definition of cochain homotopy. Since x is a cocycle,

$$[h_{n+1} \circ \delta^n(x)] = 0,$$

and $\delta^{n-1} \circ h_n(x)$ is a boundary so

$$[\delta^{n-1} \circ h_n(x)] = 0.$$

□

Definition 1.7.6. *Let M be an A-module. A projective resolution of M is a long exact sequence*

$$\cdots \longrightarrow P_2 \xrightarrow{\partial_2} P_1 \xrightarrow{\partial_1} P_0$$

where each P_i, for i a non-negative integer, is a projective A-module, such that the sequence

$$\cdots \longrightarrow P_2 \longrightarrow P_1 \longrightarrow P_0 \longrightarrow M \longrightarrow 0$$

is exact.

We notice here that since every A-module M is the image of a free A-module, projective resolutions always exist.

One nice property of projective resolutions is covered by the following proposition.

Proposition 1.23. *Two projective resolutions for an A-module M are canonically of the same homotopy type.*

Proof. This proof of this proposition is found in [HS97] on page 129. □

Let M_1 and M_2 be A-modules, and suppose

$$\cdots \longrightarrow P_2 \xrightarrow{\partial_2} P_1 \xrightarrow{\partial_1} P_0$$

is a projective resolution for M_1. We can form the cochain complex

$$\mathrm{Hom}_A(P_0, M_2) \xrightarrow{\partial_*^0} \mathrm{Hom}_A(P_1, M_2) \xrightarrow{\partial_*^1} \mathrm{Hom}_A(P_2, M_2) \longrightarrow \cdots$$

where ∂_*^i is defined as postcomposition with ∂_{i+1}, as in part 2 of Example 1.6.2. By Propositions 1.22 and 1.23, the cohomology groups of this sequence do not depend on the choice of projective resolution for M. With this, we make the following definition.

Definition 1.7.7. *With the setting as above, we define*

$$\mathrm{Ext}_A^n(M_1, M_2) = H^n(\mathrm{Hom}_A(\mathbf{P}, M_2)).$$

Since $\mathrm{Ext}_A^n(M_1, M_2)$ does not depend on the projective resolution for M_1, we introduce a specific resolution in the following definition which simplifies our calculations.

Definition 1.7.8. *Let* $\mathbf{P}$ *be a projective resolution for the* A*-module* M*. We call* $\mathbf{P}$ *a minimal projective resolution for* M *if for every projective resolution* $\mathbf{Q}$ *of* M*, there exist chain maps* $\mathbf{f}\colon \mathbf{P} \to \mathbf{Q}$ *and* $\mathbf{g}\colon \mathbf{Q} \to \mathbf{P}$ *such that* f_i *is injective and* g_i *is surjective for all nonnegative integers* i*, and both* $\mathbf{f}$ *and* $\mathbf{g}$ *are lifts of* id_M*, i.e. every square in the diagrams*

$$\begin{array}{ccccccccc}
\cdots & \longrightarrow & P_2 & \longrightarrow & P_1 & \longrightarrow & P_0 & \longrightarrow & M & \longrightarrow & 0 \\
 & & \downarrow{\scriptstyle f_2} & & \downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f_0} & & \downarrow{\scriptstyle id_M} & & \\
\cdots & \longrightarrow & Q_2 & \longrightarrow & Q_1 & \longrightarrow & Q_0 & \longrightarrow & M & \longrightarrow & 0
\end{array}$$

and

$$\begin{array}{ccccccccc}
\cdots & \longrightarrow & P_2 & \longrightarrow & P_1 & \longrightarrow & P_0 & \longrightarrow & M & \longrightarrow & 0 \\
 & & \uparrow{\scriptstyle g_2} & & \uparrow{\scriptstyle g_1} & & \uparrow{\scriptstyle g_0} & & \uparrow{\scriptstyle id_m} & & \\
\cdots & \longrightarrow & Q_2 & \longrightarrow & Q_1 & \longrightarrow & Q_0 & \longrightarrow & M & \longrightarrow & 0
\end{array}$$

commute.

Minimal projective resolutions have a straightforward construction. Let M be an A-module. Then M has a projective cover $P(M)$ with an essential homomorphism $\partial_0\colon P(M) \to M$. We will use $\Omega_n(M)$ to denote the kernel of ∂_{n-1}, so $\Omega_1(M)$ is an A-submodule of $P(M)$ and we will use ι_1 to denote the injection of $\Omega_1(M)$ into $P(M)$. Since $\Omega_1(M)$ is also an A-module, it has a projective cover, $P(\Omega_1(M))$ with an essential homomorphism $\omega_1\colon P(\Omega_1(M)) \to \Omega_1(M)$. We now define $\partial_1\colon P(\Omega_1(M)) \to P(M)$ as the composition $\iota_1 \circ \omega_1$. Continuing this process gives us the following picture

$$\begin{array}{ccccccccccc}
\cdots & \longrightarrow & P(\Omega_2(M)) & & \xrightarrow{\partial_2} & & P(\Omega_1(M)) & & \xrightarrow{\partial_1} & P(M) & \xrightarrow{\partial_0} M \\
 & & & \searrow^{\omega_2} & & \nearrow_{\iota_2} & & \searrow^{\omega_1} & & \nearrow_{\iota_1} & \\
 & & & & \Omega_2(M) & & & & \Omega_1(M) & &
\end{array}$$

of a projective resolution of M. It can be checked that this is a minimal projective resolution of M.

Lemma 1.24. *With* M *defined as above,* $\iota_n(\Omega_n(M)) \subseteq \mathrm{Rad}(P(\Omega_{n-1}(M)))$.

Proof. This is clear from the picture above. Let $w \in \Omega_n(M)$. Then

$$0 = \partial_{n-1}(\iota_n(w)) = \iota_{n-1} \circ \omega_{n-1}(\iota_n(w)).$$

Since ι_{n-1} is a monomorphism, $\omega_{n-1}(\iota_n(w)) = 0$. Since ω_{n-1} is essential, $\iota_n(w) \in \mathrm{Rad}(P(\Omega_{n-1}(M)))$. □

Corollary 1.25. *In the setting given above,*

$$\partial_n(P(\Omega_n(M))) = \iota_n(P(\Omega_n(M))) \subseteq \mathrm{Rad}(P(\Omega_{n-1}(M))).$$

Proof. This statement is a direct result of Lemma 1.24 □

Lemma 1.26. *Let M and N be A-modules with N simple, and $\mathbf{P}$ a minimal projective resolution for M. Then*

$$\mathrm{Ext}^n_A(M, N) \cong \mathrm{Hom}_A(P(\Omega_n(M)), N)$$

for all positive integers n.

Proof. To prove this we need to show that for all positive integers n, $f \circ \partial_n = 0$ for all $f \in \mathrm{Hom}_A(P(\Omega_{n-1}(M)), N)$. Since we defined δ^{n-1} as composition with ∂_n, this will imply both that

$$Im(\delta^{n-1} \colon \mathrm{Hom}_A(P(\Omega_{n-1}(M)), N) \to \mathrm{Hom}_A(P(\Omega_n(M)), N)) = \{0\}$$

and

$$Ker(\delta^n \colon \mathrm{Hom}_A(P(\Omega_n(M)), N) \to \mathrm{Hom}_A(P(\Omega_{n+1}(M)), N)) \cong \mathrm{Hom}_A(P(\Omega_n(M)), N).$$

If f is the zero homomorphism, then we are done. So assume that the image of f, $Im(f)$, is nonzero, so it must be N. By Corollary 1.25, the image of ∂_n is contained in the radical of $P(\Omega_{n-1}(M))$ which is in the kernel of f since N is simple. Thus, we have $f \circ \partial_n = 0$ for all $f \in \mathrm{Hom}_A(P(\Omega_{n-1}(M)), N)$ and all positive integers n. □

1.8 Representations and Characters

The purpose of this section is to introduce the concepts of representations, characters, and Brauer characters for finite groups. This material can be found in many books on representation theory of finite groups, in particular, [Isa76] and [Ser96].

For this section, we continue with $\mathbb{F}$ denoting a field of characteristic $p > 0$. We also use F to denote a field of arbitrary characteristic and $\mathbb{C}$ to denote the field of complex numbers.

We use $\mathrm{GL}(V)$ to denote the general linear group for a vector space V, and $\mathrm{GL}_n(F)$ for the $n \times n$ matrices over F with nonzero determinant. If $V = F^n$, then it is obvious that the difference between these two notations is that a basis of V has been chosen for $\mathrm{GL}_n(F)$, and not for $\mathrm{GL}(F^n)$.

Definition 1.8.1. *Let G be a group. Then a representation of G is a homomorphism $\rho\colon G \to \mathrm{GL}(F^n)$ for some positive integer n. We call n the degree of ρ.*

A representation ρ of G affords a representation of the group algebra FG by the linear extension

$$\rho(\sum_{g\in G} a_g g) = \sum_{g\in G} a_g \rho(g).$$

If ρ has degree n, and V is an n-dimensional row space over $\mathbb{F}$, then any subspace of V generates a right $\mathbb{F}G$-module.

We call a representation of G irreducible if the corresponding FG-module is simple.

In order to do calculations, it is convenient to fix a basis for F^n which leads to the following definition.

Definition 1.8.2. *Let G be a group. Then a matrix representation of G is a homomorphism $\varrho\colon G \to \mathrm{GL}_n(F)$ for some positive integer n. We call n the degree of ϱ.*

We use $\mathrm{Tr}\colon \mathrm{GL}_n(F) \to F$ to denote the trace map defined by

$$a \mapsto \sum_{i=1}^{n} a_{ii}$$

with $a \in \mathrm{GL}_n(F)$. With this definition we can apply Tr to any representation $\varrho(g)$ for g in a group G. For a representation ρ, we have $\mathrm{Tr}(\rho(g))$ as the sum of the eigenvalues of $\rho(g)$. This statement follows from results of linear algebra which are not covered here.

Definition 1.8.3. *Let ρ be a (matrix) representation of G. The map $\chi_\rho\colon G \to \mathbb{C}$ defined by $\chi_\rho(g) \mapsto \mathrm{Tr}(\rho(g))$ is called the character of ρ.*

Since we defined characters by using the trace map which can be applied to both representations and matrix representations, we will drop the distinction between them and just say representation.

***Example* 1.8.1.** *For any group G, we have the trivial representation $\rho : G \to \mathrm{GL}(\mathbb{C})$ defined by $g \mapsto 1$. This representation affords the trivial character, denoted 1_G, which clearly has value 1 for every $g \in G$.*

Proposition 1.27. *Let χ_ρ be the character for a representation ρ of G of degree n. Then the following are true.*

1. $\chi_\rho(1) = n$,
2. $\chi_\rho(s^{-1}) = \overline{\chi_\rho(s)}$, *for* $s \in G$,
3. $\chi_\rho(tst^{-1}) = \chi_\rho(s)$, *for* $s, t \in G$,

where $\overline{z}$ is the complex conjugate of z,

Proof. A proof of this proposition is found in [Ser96] on page 10. □

An important result of this proposition is that the character χ of G is constant on conjugacy classes of G. This aids in storage of character information and also in calculations.

We call a character irreducible if it comes from an irreducible representation. We denote by $\mathrm{Irr}(G)$ the set of irreducible characters of the group G.

Proposition 1.28. *Let G be a group. Then $|\mathrm{Irr}(G)|$ equals the number of conjugacy classes of G.*

Proof. A proof of this proposition is found in [Isa76] on page 16. □

It is clear that the sum of two characters is again a character. However, the difference of two characters may or may not be a character, but it is still constant the conjugacy classes of G, which leads to the following definition.

Definition 1.8.4. *A class function on a group G is a function $\varphi\colon G \to \mathbb{C}$ which is constant on the conjugacy classes of G.*

Clearly, characters are class functions, but not all class functions are characters. For instance, the zero function is a class function and not a character. The reason for considering class functions is expressed in the following theorem.

Theorem 1.29. *Every class function φ can be expressed as*

$$\varphi = \sum_{\chi \in \mathrm{Irr}(G)} \alpha_\chi \chi,$$

where $\alpha_\chi \in \mathbb{C}$. Furthermore, φ is a character if and only if all of the α_χ are nonnegative integers and $\varphi \neq 0$.

Proof. A proof of this theorem is found in [Isa76] on page 16. □

This theorem is important since it says that we can express any character in terms of irreducible characters. This allows us to focus on irreducible characters

in our study of characters. Studying irreducible characters leads to the following orthogonality conditions.

Proposition 1.30 (First Orthogonality Condition)**.** *Let χ_i and χ_j be in* $\mathrm{Irr}(G)$*. Then*

$$\frac{1}{|G|}\sum_{g\in G}\chi_i(g)\chi_j(g^{-1}) = \delta_{ij}.$$

Proof. A proof of the First Orthogonality Condition for characters is found in [Isa76] on page 20. □

Combining this result with part 2 of Proposition 1.27 leads to the following definition.

Definition 1.8.5. *Let φ_1 and φ_2 be class functions on G. Then*

$$(\varphi_1, \varphi_2) = \frac{1}{|G|}\sum_{g\in G}\varphi_1(g)\overline{\varphi_2(g)}$$

is called the scalar product of φ_1 and φ_2.

Some obvious properties of the scalar product are

(a) $(\varphi_1, \varphi_2) = \overline{(\varphi_2, \varphi_1)}$;

(b) $(\varphi_1, \varphi_1) > 0$ unless $\varphi_1 = 0$;

(c) $(c_1\varphi_1 + c_2\varphi_2, \varphi_3) = c_1(\varphi_1, \varphi_3) + c_2(\varphi_2, \varphi_3)$;

(d) $(\varphi_1, c_1\varphi_2 + c_2\varphi_3) = \overline{c_1}(\varphi_1, \varphi_2) + \overline{c_2}(\varphi_1, \varphi_3)$

where φ_1, φ_2 and φ_3 are class functions on G and c_1, c_2 are in $\mathbb{C}$.

Let $\chi \in Irr(G)$, then a couple of not so obvious properties are

(a) $(\chi, \chi) = 1$;

(b) $(\chi, \vartheta) = n$,

where n is the coefficient of χ when the character ϑ is expressed as a sum of irreducible characters of G.

By summing over the irreducible characters instead of over the elements of G, we get a second orthogonality condition.

Proposition 1.31 (Second Orthogonality Condition)**.** *Let $g, h \in G$. Then*

$$\sum_{\chi \in \mathrm{Irr}(G)} \chi(g)\overline{\chi(h)} = 0$$

if g is not conjugate to h in G.

Proof. A proof of the Second Orthogonality Condition for characters is found in [Isa76] on page 21. □

When studying groups, we often learn a lot by looking at the subgroups. If we have a representation ρ for a group G, then we denote the restriction of ρ to the subgroup H by ρ_H. This is clearly a representation for H. If χ is the character corresponding to ρ, then the character corresponding to ρ_H is denoted by χ_H and is called the restriction of χ to H.

Conversely, the following definition gives a character for G which comes from a character of H.

Definition 1.8.6. *Let H be a subgroup of G and let χ be a character of H. Then χ^G, called the induced character, is defined by*

$$\chi^G(g) = \frac{1}{|H|} \sum_{x \in G} \chi^\circ(xgx^{-1}),$$

where χ° is defined by $\chi^\circ(g) = \chi(g)$ if $g \in H$ and $\chi^\circ(g) = 0$ otherwise.

A proof that χ^G is a character for G is found in [Isa76] on page 63.

We have introduced both induction and restriction, the following lemma gives a connection between the two ideas.

Lemma 1.32 (Frobenius Reciprocity)**.** *Let H be a subgroup of G and suppose that χ is a character for H and ϑ is a character for G. Then*

$$(\chi, \vartheta_H) = (\chi^G, \vartheta).$$

Proof. A proof of this lemma is found in [Isa76] on page 62. □

Let Ω be a G-set. Then define $\vartheta(g) = |\{\alpha \in \Omega | \alpha^g = \alpha\}|$. Let M be a $\mathbb{C}G$-module with a $\mathbb{C}$-basis corresponding to Ω and define the action of G on M as permuting the basis as it permutes the set Ω. Then M is called a permutation module. The character afforded by M is ϑ. We call this the permutation character of Ω.

Lemma 1.33. *Let G act transitively on Ω, let $\alpha \in \Omega$ and let H be the stabilizer in G of α. Then $(1_H)^G$ is the permutation character of the action.*

Proof. A proof of this lemma is found in [Isa76] on page 68. □

We would like to define characters for representations of G over $\mathbb{F}$, but the trace map does not work. In characteristic p, the representation which is the direct sum of $n \cdot p$ copies of the trivial representation, has trace zero for all conjugacy classes of G, for all positive integers n. In this case, the trace map does not distinguish between these representations. A solution to this problem was developed by R. Brauer and a thorough treatment can be found in [Isa76], starting on page 262.

Definition 1.8.7. *We define the exponent of a group G, denoted by $exp(G)$, to be the smallest integer n such that $g^n = 1$ for all $g \in G$.*

Let U be the group of $exp(G)^{th}$ roots of unity in $\mathbb{C}$. Then we define a group monomorphism $\mu \colon U \to \mathbb{F}^\times$. The monomorphism μ is not unique and while this choice does not affect the theory, it is a choice that needs to be made in practice. A discussion of this choice is found in [JLPW95] in the section on Conway polynomials.

Definition 1.8.8. *Let ρ be a representation of degree f for G over $\mathbb{F}$. Define a function ϕ from the p-regular elements of G, the elements whose order is not divisible by p, into $\mathbb{C}$ as follows. Let x be a p-regular element of G with $\varepsilon_1, \ldots, \varepsilon_f \in \mathbb{F}$ the eigenvalues of $\rho(x)$ counted with multiplicity. Let $\phi(x) = \sum_{i=1}^{f} \mu^{-1}(\varepsilon_i)$. The function ϕ is called the Brauer character of G afforded by ρ.*

Let $\rho_1, \ldots, \rho_r$ be a set of representatives for similarity classes of irreducible representations of G over $\mathbb{F}$ and let ϕ_i be the Brauer character afforded ρ_i. We say that the set of ρ_i for $1 \leq i \leq r$ are the irreducible Brauer characters and we denote this set by $\mathrm{IBr}(G)$. Similar to the result for ordinary characters, every Brauer character is of the form $\sum n_i \phi_i$ where the n_i are nonnegative integers and not all zero for $1 \leq i \leq r$. A proof of this is found in [Isa76] on pages 264 and 265.

Theorem 1.34. *Let χ be an ordinary character of G and let $\widehat{\chi}$ denote the restriction of χ to the p-regular elements of G. Then $\widehat{\chi}$ is a Brauer character of G.*

Proof. A proof of this theorem is found in [Isa76] on page 266. □

Since we can restrict ordinary characters to get Brauer characters, the following definition looks at the decomposition of the Brauer characters which come from restricting irreducible characters.

Definition 1.8.9. *Let $\chi \in \mathrm{Irr}(G)$ and let $\widehat{\chi}$ be the restriction of χ to the p-regular elements of G. We write*

$$\widehat{\chi} = \sum_{\phi \in \mathrm{IBr}(G)} d_{\chi\phi}\phi.$$

The nonnegative integers $d_{\chi\phi}$ are uniquely defined and are called the decomposition numbers of G for a prime p.

We use the decomposition numbers of G to define a special set of characters.

Definition 1.8.10. *For each $\phi \in \mathrm{IBr}(G)$ we define*

$$\Phi_\phi = \sum_{\chi \in \mathrm{Irr}(G)} d_{\chi\phi}\chi.$$

The Φ's are called projective indecomposable characters of G.

The following proposition ensures that there is a projective character for each $\phi \in \mathrm{IBr}(G)$.

Proposition 1.35. *If $\phi \in \mathrm{IBr}(G)$, then there exists $\chi \in \mathrm{Irr}(G)$ with $d_{\chi\phi} \neq 0$.*

Proof. A proof of this proposition is found in [Isa76] on page 268. □

The projective characters provide a link between ordinary characters and Brauer characters that are necessary in this dissertation. In order to see this link, we need the following lemma.

Lemma 1.36. *Let $x, y \in G$ be such that p does not divide the order of x. Then*

$$\sum_{\chi \in \mathrm{Irr}(G)} \chi(x)\overline{\chi(y)} = \sum_{\phi \in \mathrm{IBr}(G)} \phi(x)\overline{\Phi_\phi(y)}.$$

Proof. Since p does not divide the order of x, for all $\chi \in \mathrm{Irr}(G)$,

$$\chi(x) = \widehat{\chi}(x) = \sum_{\phi \in \mathrm{IBr}(G)} d_{\chi\phi}\phi(x).$$

Then

$$\begin{aligned}
\sum_{\chi \in \mathrm{Irr}(G)} \chi(x)\overline{\chi(y)} &= \sum_{\chi \in \mathrm{Irr}(G), \phi \in \mathrm{IBr}(G)} d_{\chi\phi}\phi(x)\overline{\chi(y)} \\
&= \sum_{\chi \in \mathrm{Irr}(G), \phi \in \mathrm{IBr}(G)} \phi(x)\overline{d_{\chi\phi}\chi(y)} \\
&= \sum_{\phi \in \mathrm{IBr}(G)} \phi(x)\overline{\Phi_\phi(y)}.
\end{aligned}$$

□

The following lemma gives the value of projective indecomposable characters of G on elements of G whose order is divisible by p.

Lemma 1.37. *Let $y \in G$ such that p divides the order of y in G. Then $\Phi_\phi(y) = 0$ for all $\phi \in \mathrm{IBr}(G)$.*

Proof. By Lemma 1.36, we have

$$\sum_{\chi \in \mathrm{Irr}(G)} \chi(id_G)\overline{\chi(y)} = \sum_{\phi \in \mathrm{IBr}(G)} \phi(id_G)\overline{\Phi_\phi(y)}$$

with id_G the identity element of G. By the Second Orthogonality Condition (Proposition 1.31),

$$\sum_{\chi \in \mathrm{Irr}(G)} \chi(id_G)\overline{\chi(y)} = 0$$

since id_G and y are not conjugates in G. We also know that $\phi(id_G) > 0$ and $\Phi_\phi(y) \geq 0$ for all $\phi \in \mathrm{IBr}(G)$. So, in order for

$$\sum_{\phi \in \mathrm{IBr}(G)} \phi(id_G)\overline{\Phi_\phi(y)} = 0,$$

$\Phi_\phi(y) = 0$ for all $\phi \in \mathrm{IBr}(G)$. □

Given a character χ of G, it is easy to construct the Brauer character $\widehat{\chi}$. However, given an arbitrary Brauer character, we can not construct a character from it. By Lemma 1.37, projective characters are 0 on elements of G with order divisible by p. This means that given $\widehat{\Phi_\phi}$ for some $\phi \in \mathrm{IBr}(G)$, we can construct Φ_ϕ by setting $\Phi_\phi(g) = \widehat{\Phi_\phi}(g)$ for all p-regular elements in G, and $\Phi_\phi(g) = 0$ for all other elements in G.

Theorem 1.38. *The projective characters Φ_ϕ for $\phi \in \mathrm{IBr}(G)$ form a basis for the class functions on G which are zero on the set of elements of G which are not p-regular.*

Proof. The Φ_ϕ for $\phi \in \mathrm{IBr}(G)$ are linearly independent since

$$0 = \sum_{\phi \in \mathrm{IBr}(G)} \alpha_\phi \Phi_\phi = \sum_{\phi \in \mathrm{IBr}(G)} \alpha_\phi \sum_{\chi \in \mathrm{Irr}(G)} d_{\chi\phi}\chi = \sum_{\phi \in \mathrm{IBr}(G)} \sum_{\chi \in \mathrm{Irr}(G)} \alpha_\phi d_{\chi\phi}\chi$$

with $\alpha_\phi \in \mathbb{C}$ for $\phi \in \mathrm{IBr}(G)$, implies that $\alpha_\phi = 0$ for $\phi \in \mathrm{IBr}(G)$.

The Φ_ϕ for $\phi \in \mathrm{IBr}(G)$ span this space since it is isomorphic to the space of class functions on the p-regular classes of G which is spanned by $\mathrm{IBr}(G)$. $\square$

Chapter 2
Morita Theory

The purpose of this chapter is to consider the equivalence of module categories. In particular, we want to consider the relationship between the categories of right modules Mod_A and Mod_B where B is a subalgebra of A. In Section 2.1, we will introduce Morita equivalence and Section 2.2 we consider sufficient conditions for two algebras to be Morita equivalent. Development of the material in these sections can be found in [AF92] and [Lux97]. Section 2.3 will introduce condensation algebras which are particular subalgebras of A Morita equivalent to A. In Section 2.4, we discuss construction of condensation subalgebras for group algebras. Discussions and applications of condensation can be found in [Lux97], [Ryb90], [Sző98], and [Tha81].

2.1 Morita Equivalence

In Definition 1.6.5, we introduced equivalence of categories. We now use that idea to define Morita equivalence.

Definition 2.1.1. *We call the algebras A and B Morita equivalent if the categories* Mod_A *and* Mod_B *are equivalent.*

The following lemma summarizes some consequences of Morita equivalence.

Lemma 2.1. *Let A and B be Morita equivalent algebras with $F\colon \mathrm{Mod}_A \to \mathrm{Mod}_B$ forming this equivalence. Then the following hold for M, M', M'' in* Mod_A.

1. *The sequence*

$$0 \longrightarrow M' \xrightarrow{f} M \xrightarrow{g} M'' \longrightarrow 0$$

is (split) exact if and only if the sequence

$$0 \longrightarrow F(M') \xrightarrow{F(f)} F(M) \xrightarrow{F(g)} F(M'') \longrightarrow 0$$

is (split) exact.

2. *M is projective if and only if $F(M)$ is projective.*

3. *M is a progenerator if and only if $F(M)$ is a progenerator.*

4. *$f\colon M \to M'$ is a projective cover if and only if $F(f)\colon F(M) \to F(M')$ is a projective cover.*

5. *M is simple (semisimple) if and only if $F(M)$ is simple (semisimple).*

6. *M is indecomposable if and only if $F(M)$ is indecomposable.*

Furthermore, the lattice of submodules of M is isomorphic to the lattice of submodules of $F(M)$. This implies that $F(\mathrm{Rad}(M)) = \mathrm{Rad}(F(M))$.

Proof. Proofs of the statements in this lemma can be found in [AF92] on pages 254-258. □

Corollary 2.2. *Let A and B be Morita equivalent algebras. Then the number of isomorphism classes of simple modules is the same for A and B.*

Proof. This follows directly from part 5 of Lemma 2.1. □

2.2 Sufficient Conditions for Morita Equivalence

In the last section, we saw some of the consequences of two algebras A and B being Morita equivalent. In this section, we outline sufficient conditions for A and B to be Morita equivalent.

The reader should recall the definition of the tensor functor given in Example 1.6.2. The following theorem states a criterion for a pair of tensor functors to form a Morita equivalence.

Theorem 2.3. *Let A and B be $\mathbb{F}$-algebras with P an A-B bimodule and Q a B-A bimodule. If $P \otimes_B Q \cong A$ as A-A bimodules and $Q \otimes_A P \cong B$ as B-B bimodules, then the functors $(- \otimes_A P)$ and $(- \otimes_B Q)$ define a Morita equivalence between A and B.*

Proof. By Definitions 2.1.1 and 1.6.5 we need to show that $(- \otimes_A P)(- \otimes_B Q)$ is naturally equivalent to id_{Mod_B} and $(-\otimes_B Q)(-\otimes_A P)$ is naturally equivalent to id_{Mod_A}. Let $V \in \mathrm{Mod}_B$, then

$$\begin{aligned}(- \otimes_A P)(- \otimes_B Q)(V) &= (- \otimes_A P)(V \otimes_B Q) \\ &= (V \otimes_B Q) \otimes_A P \\ &\cong V \otimes_B (Q \otimes_A P) \\ &\cong V \otimes_B B \\ &\cong V.\end{aligned}$$

Let $f \in \mathrm{Hom}_B(V, W)$ for objects V, W in Mod_B, and $v \in V$, then

$$\begin{aligned}((- \otimes_A P)(- \otimes_B Q)(f))((v \otimes_B q) \otimes_A p) &= ((f_{(-\otimes_B Q)})_{(-\otimes_A P)})((v \otimes_B q) \otimes_A p) \\ &= (f_{(-\otimes_B Q)})(v \otimes_B q) \otimes_A p \\ &= f(v) \otimes_B q \otimes_A p\end{aligned}$$

for all $q \in Q$ and all $p \in P$. From these calculations, it is clear that the diagram

$$\begin{array}{ccc}(- \otimes_A P)(- \otimes_B Q)(V) & \xrightarrow{\cong} & V \\ \Big\downarrow{\scriptstyle (f_{(-\otimes_B Q)})_{(-\otimes_A P)}} & & \Big\downarrow{\scriptstyle f} \\ (- \otimes_A P)(- \otimes_B Q)(W) & \xrightarrow{\cong} & W\end{array}$$

commutes. We have shown that there exists a natural transformation between the functors $(- \otimes_A P)(- \otimes_B Q)$ and id_{Mod_B} where every component is an isomorphism, which gives us a natural equivalence. A similar argument works for the functors $(- \otimes_B Q)(- \otimes_A P)$ and id_{Mod_A}. □

Corollary 2.4. *Let P be a progenerator for Mod_A and $Q = \mathrm{Hom}_A(P, A)$. Then $E = \mathrm{End}_A(P)$ is Morita equivalent to A and a categorical equivalence is given by $(- \otimes_A Q)$ and $(- \otimes_E P)$.*

Proof. A proof of this corollary can be found in [Lux97] on page 43. □

Theorem 2.3 and Corollary 2.4 describe the conditions under which the algebras A and B are Morita equivalent and express the equivalence in terms of progenerators. The following theorem shows that given a Morita equivalence of A and B, we can use progenerators with Hom and tensor functors to define Morita equivalences.

Theorem 2.5 (Morita's Theorem)**.** *Let A and B be Morita equivalent algebras with the functors $F\colon \mathrm{Mod}_A \to \mathrm{Mod}_B$ and $G\colon \mathrm{Mod}_B \to \mathrm{Mod}_A$ forming a Morita equivalence between A and B. Then $P = F(A)$ is an A-B bimodule and $Q = G(B)$ is an B-A bimodule such that $P \otimes_B Q \cong A$ as A-A bimodules and $Q \otimes_A P \cong B$ as B-B bimodules. Furthermore, the following statements hold.*

1. *P is a progenerator for Mod_B and ${}_A\mathrm{Mod}$.*

2. *Q is a progenerator for Mod_A and ${}_B\mathrm{Mod}$.*

3. *There are bimodule isomorphisms*

$$Q \cong \mathrm{Hom}_A(P, A) \cong \mathrm{Hom}_B(P, B),$$

$$P \cong \mathrm{Hom}_A(Q, A) \cong \mathrm{Hom}_B(Q, B).$$

4. *There are ring isomorphisms*

$$A \cong \mathrm{End}_B(P) \cong \mathrm{End}_B(Q),$$

$$B \cong \mathrm{End}_A(P) \cong \mathrm{End}_A(Q).$$

5. *The functors* $- \otimes_A P$ *and* $- \otimes_B Q$ *define a Morita equivalence between* A *and* B.

6. *The functors* $\mathrm{Hom}_A(Q, -)$ *and* $\mathrm{Hom}_B(P, -)$ *define a Morita equivalence between* A *and* B.

Proof. A proof of these statements can be found in [AF92] on page 262. □

2.3 Condensation and Idempotents

We know that A and $\mathrm{End}_A(P)$ are Morita equivalent when P is a progenerator for Mod_A, by Corollary 2.4. Our task now is to construct P for a given algebra A. We don't necessarily know many of the A-modules in Mod_A, but we do know that the right ideals of A are in Mod_A. In Section 1.5, we discussed the idempotent elements in A. Idempotents give us ideals in A which prove to be useful in this dissertation.

By Theorem 1.19, $\mathrm{End}_A(eA) \cong eAe$, which leads to the following definition.

Definition 2.3.1. *Let* $e \in A$ *be an idempotent. We call the subalgebra* eAe *of* A *the condensation subalgebra of* A *corresponding to* e.

In Theorem 2.3 and Corollary 2.4, we saw a relationship between progenerators and tensor functors forming a Morita equivalence. The following theorem realizes this relationship in terms of idempotents.

Theorem 2.6. *Let* $e \in A$ *be an idempotent,* $e_1A, \ldots, e_kA$ *be representatives of the isomorphism classes of projective indecomposable* A*-modules, and* $S_1, \ldots, S_k$ *be representatives of the isomorphism classes of simple* A*-modules such that* $e_iA/\operatorname{Rad}(e_iA) \cong S_i$. *Then the following statements are equivalent.*

1. *The functors* $(- \otimes_{eAe} eA)$ *and* $(- \otimes_A Ae)$ *form a Morita equivalence between* A *and* eAe.

2. AeA *is equal to* A.

3. *For all* $1 \leq i \leq k$, *the* A*-module* e_iA *is a direct summand of* eA.

4. $S_ie \neq 0$ *for all* $1 \leq i \leq k$.

Proof. A proof of this theorem can be found in [Lux97] on page 46. □

We name the idempotents satisfying this theorem in the following definition.

Definition 2.3.2. *Let* e *be an idempotent in* A. *If* e *satisfies one of the conditions in the Theorem* 2.6, *then* e *is called a faithful idempotent of* A.

2.4 Condensation of Group Algebras

We have introduced the idea of condensation for finitely generated algebras. We are interested in applying condensation to group algebras. More specifically, we consider the case where G is a finite group and $\mathbb{F}$ is a field with characteristic p dividing the order of G.

The first step to take is finding an idempotent to use for condensation. Fortunately, idempotents can be constructed easily in group algebras, as in the following lemma.

Lemma 2.7. *Let* H *be a subgroup of* G *such that* p *does not divide the order of* H. *Then*

$$e_H := \frac{1}{|H|} \sum_{h \in H} h$$

is an idempotent in $\mathbb{F}G$.

Proof. Clearly, e_H is in $\mathbb{F}G$, so we just need to check that $e_H^2 = e_H$. Squaring e_H gives

$$\begin{aligned} e_H^2 &= (\frac{1}{|H|}\sum_{h_1 \in H} h_1)(\frac{1}{|H|}\sum_{h_2 \in H} h_2) \\ &= \frac{1}{|H|^2}(\sum_{h_1 \in H} h_1)(\sum_{h_2 \in H} h_2) \\ &= \frac{1}{|H|^2}\sum_{h_1 \in H}\sum_{h_2 \in H} h_1 h_2 \\ &= \frac{|H|}{|H|^2}\sum_{h \in H} h = e_H \end{aligned}$$

which proves the lemma. □

Not every idempotent e_H in $\mathbb{F}G$ is faithful. The subgroups which give faithful idempotents are dealt with in the following definition.

Definition 2.4.1. *We call H a condensation subgroup for $\mathbb{F}G$ if $e_H\mathbb{F}Ge_H$ is Morita equivalent to $\mathbb{F}G$.*

In the next section we will focus on using the idempotent e_H to condense $\mathbb{F}G$-modules. For this to be worthwhile, we need to determine which subgroups are condensation subgroups of G. We could construct e_H and then check if it is faithful for all subgroups H of G. This is clearly not an efficient use of time. Fortunately, the following theorem gives a character theoretic condition on H to ensure e_H is faithful.

Theorem 2.8. *The tensor functors $(- \otimes_{e_H\mathbb{F}Ge_H} e_H\mathbb{F}G)$ and $(- \otimes_{\mathbb{F}G} \mathbb{F}Ge_H)$ form a Morita equivalence between $Mod_{\mathbb{F}G}$ and $Mod_{e_H\mathbb{F}Ge_H}$ if and only if for all $\phi \in \mathrm{IBr}(G)$, the character theoretic scalar product of the restriction of ϕ to H with the trivial character 1_H is nonzero.*

Proof. A proof of this theorem is found in [Lux97] on page 53. □

The following corollary lifts the calculation of Theorem 2.8 up to a calculation using ordinary characters. We notice that since 1_H^G is zero on the classes of G which are not p-regular, we can express it in terms of the projective characters Φ_ϕ for $\phi \in \mathrm{IBr}(G)$ by Theorem 1.38. Corollary 2.9 is implemented in the *GAP* routine `FaithfulKondBurn` documented in [Lux97] on page 135.

Corollary 2.9. *Write*

$$1_H^G = \sum_{\phi \in \mathrm{IBr}(G)} n_\phi \Phi_\phi.$$

If the integers $n_\phi > 0$ for all $\phi \in \mathrm{IBr}(G)$ then the tensor functors $(- \otimes_{e_H \mathbb{F}G e_H} e_H \mathbb{F}G)$ and $(- \otimes_{\mathbb{F}G} \mathbb{F}G e_H)$ form a Morita equivalence between $Mod_{\mathbb{F}G}$ and $Mod_{e_H \mathbb{F}G e_H}$.

Proof. By Theorem 2.8, we need to show that $(\phi_H, 1_H) > 0$ for all $\phi \in \mathrm{IBr}(G)$.

Using Frobenius reciprocity, Lemma 1.32, we consider

$$\begin{aligned}
(\phi, \widehat{1_H^G}) &= (\phi, \sum_{\phi \in \mathrm{IBr}(G)} n_\phi \widehat{\Phi_\phi}) \\
&= \sum_{\phi \in \mathrm{IBr}(G)} n_\phi (\phi, \widehat{\Phi_\phi}) \\
&= \sum_{\phi \in \mathrm{IBr}(G)} n_\phi (\phi, \sum_{\chi \in \mathrm{Irr}(G)} d_{\chi\phi} \widehat{\chi}) \\
&= \sum_{\phi \in \mathrm{IBr}(G)} n_\phi \sum_{\chi \in \mathrm{Irr}(G)} d_{\chi\phi} (\phi, \widehat{\chi}) \\
&= \sum_{\phi \in \mathrm{IBr}(G)} n_\phi \sum_{\chi \in \mathrm{Irr}(G)} d_{\chi\phi} (\phi, \sum_{\phi' \in \mathrm{IBr}(G)} d_{\chi\phi'} \phi') \\
&= \sum_{\phi \in \mathrm{IBr}(G)} n_\phi \sum_{\chi \in \mathrm{Irr}(G)} d_{\chi\phi} \sum_{\phi' \in \mathrm{IBr}(G)} d_{\chi\phi'} (\phi, \phi')
\end{aligned}$$

which is greater than zero when $\phi = \phi'$ by Proposition 1.35. Thus $n_\phi > 0$ for all $\phi \in \mathrm{IBr}(G)$ implies $(\phi_H, 1_H) > 0$ for all $\phi \in \mathrm{IBr}(G)$. □

2.5 Calculating with e_H

We now know how to find H such that e_H is a faithful idempotent in $\mathbb{F}G$. Summing over all elements of a subgroup is not always efficient, particularly if we are working with a representation of large dimension. The following lemma gives a realization of Me_H which will help in our calculations.

Lemma 2.10. *Let H be a condensation subgroup of G. Then for all $M \in \mathrm{Mod}_{\mathbb{F}G}$, $Me_H = M^H$.*

Proof. Clearly, $Me_H \subseteq M^H$ since $e_H \cdot h = e_H$ for any $h \in H$. Conversely, if we have $m \in M^H$, we see that

$$me_H = \frac{1}{|H|}\sum_{h\in H} mh = \frac{1}{|H|}\sum_{h\in H} m = \frac{|H|}{|H|}m = m,$$

so $m \in Me_H$. □

In this section, we let H be a condensation subgroup for G. Then our condensation subalgebra of $\mathbb{F}G$ is $e_H\mathbb{F}Ge_H$. From Corollary 1.20, $\mathrm{Hom}_{\mathbb{F}G}(e_H\mathbb{F}G, -)$ is a functor from $\mathbb{F}G$ to $e_H\mathbb{F}Ge_H$ as defined in Example 1.6.2. So for an $\mathbb{F}G$-module M, we study the $e_H\mathbb{F}Ge_H$-module Me_H. But the functor, $\mathrm{Hom}_{\mathbb{F}G}(e_H\mathbb{F}G, -)$, does not tell us how $e_H\mathbb{F}Ge_H$ acts on Me_H. We need to study e_H to get the action of $e_H\mathbb{F}Ge_H$.

Instead of an arbitrary $\mathbb{F}G$-module, we restrict to the case that M is an $\mathbb{F}G$-permutation module. In particular, this means that M has a basis indexed by a G-set Ω. Let $O_1, \ldots, O_r$ be the H-orbits of Ω, and $\overline{O_i} := \sum_{k\in O_i} b_k$ for $1 \le i \le r$. In this setting we get the following lemma.

Lemma 2.11. *The set of $\overline{O_i} := \sum_{k\in O_i} b_k$ for $1 \le i \le r$ form a basis for Me_H.*

Proof. It is clear that $\overline{O_i} \in Me_H$ since $\overline{O_i}h = \overline{O_i}$ and $Me_H = M^H$, by Lemma 2.10.

Let m be an element in Me_H, then $m = \sum_{k\in\Omega} \alpha_k b_k$ with $\alpha_k \in \mathbb{F}$. Since $mh = m$, we have that $\alpha_{k_1} = \alpha_{k_2}$ when $k_1, k_2 \in O_i$, thus $m = \sum_{i=1}^{r} \alpha_i \overline{O_i}$. So the set of $\overline{O_i}$ for

$1 \leq i \leq r$ spans Me_H. Linear independence follows from the fact that the $O_1, \ldots, O_r$ are a partition of the basis for M. □

Now that we have a basis for Me_H, we would like to know how $e_H g e_H$ with $g \in G$ acts on this basis.

Lemma 2.12. *The action of $e_H g e_H$ for $g \in G$ on Me_H with respect to the basis from Lemma 2.11 is given by*

$$\overline{O_i} e_H g e_H = \sum_{j=1}^{r} c_{ij} \frac{1}{|O_j|} \overline{O_j}$$

where $c_{ij} = |\{k \in O_i | kg \in O_j\}|$. Furthermore, for a fixed i,

$$|\{c_{ij} | c_{ij} \neq 0, 1 \leq j \leq r\}| \leq |H|.$$

Proof. We notice that if $k \in O_j$, then

$$b_k e_H = \frac{1}{|H|} \sum_{h \in H} b_k h = \frac{1}{|H|} |Stab_H(b_k)| \overline{O_j} = \frac{1}{|O_j|} \overline{O_j}.$$

From this we get

$$\begin{aligned} \overline{O_j} e_H g e_H &= \overline{O_j} g e_H \\ &= \sum_{k \in O_j} b_k g e_H \\ &= \sum_{k \in O_j} b_{kg} e_H \\ &= \sum_{j=1}^{r} c_{ij} \frac{1}{|O_i|} \overline{O_i}. \end{aligned}$$

To finish the proof, we notice that if $c_{ij} \neq 0$, then there is a k in O_i such that $k^g \in O_j$. But there are at most $|H|$ elements in O_i, so the size of the set $\{c_{ij} | c_{ij} \neq 0, 1 \leq j \leq r\}$ is bounded by the order of H. □

The following example explicitly shows how to calculate the c_{ij}'s and construct the matrix for $e_H g e_H$ for $g \in G$.

***Example* 2.5.1.** *Let $G = A_5$ and $\mathbb{F}$ the field with 3 elements. We take the permutations $(2,4)(3,5), (1,2,5)$ as generators for G and let H be the subgroup generated by $(2,3)(4,5), (2,4)(3,5)$.*

Take Ω to be the set $\{1,2,3,4,5\}$. The subgroup H has 2 orbits in Ω, $O_1 = \{1\}$, and $O_2 = \{2,3,4,5\}$. These orbits are contiguous, so we read off

$$c_{11} = 0 \quad c_{12} = 1$$
$$c_{21} = 1 \quad c_{22} = 1$$

for the permutation $(1,2,5)$. Now, weighting c_{ij} by $|O_i|^{-1} \mod 3$, we get the matrix

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

for $e_H(1,2,5)e_H$.

We would like to repeat the above process with an $\mathbb{F}G$-module which is important in the following, namely, $e_H\mathbb{F}G$. Before we do this, we need the following results.

Lemma 2.13. *Let H and e_H be defined as above, and suppose $g, g' \in G$. Then $e_Hg = e_Hg'$ if and only if $Hg = Hg'$.*

Proof. Suppose $Hg = Hg'$. Then there exists an $h \in H$ such that $hg = g'$ and $e_Hg = e_Hhg = e_Hg'$.

Suppose $e_Hg = e_Hg'$. Then $e_Hg'g^{-1} = e_H$. This implies that $g'g^{-1} \in H$ and so $Hg = Hg'$. □

Let $\{g_j | 1 \leq j \leq [G:H]\}$ be a complete set of representatives for the right cosets of H in G.

Proposition 2.14. *The set $\{e_Hg_j | 1 \leq j \leq [G:H]\}$ is a basis for $e_H\mathbb{F}G$ with H and e_H defined as above.*

Proof. From Lemma 2.13, we get that $\{e_H g_j | 1 \leq j \leq [G : H]\}$ is a generating set for $e_H \mathbb{F}G$. To show that this set is linearly independent, suppose

$$\sum_{j=1}^{[G:H]} \alpha_{g_j} e_H g_j = 0.$$

We can rewrite this sum with the basis of $\mathbb{F}G$ to get

$$\sum_{j=1}^{[G:H]} \alpha_{g_j} e_H g_j = \sum_{j=1}^{[G:H]} \frac{1}{|H|} \sum_{h \in H} \alpha_{g_j} h g_j = \sum_{g' \in G} \alpha_{g_j} g'$$

and since $H\backslash G$ is a disjoint partition of G we get that $\alpha_{g_j} = 0$ for $1 \leq j \leq [G : H]$. $\square$

Notice that the basis elements of $\mathbb{F}G$ permute the basis elements of $e_H \mathbb{F}G$, so $e_H \mathbb{F}G$ is the $\mathbb{F}G$ permutation module corresponding to G acting on the cosets of H.

***Example* 2.5.2.** *Again let $G = A_5$ and $\mathbb{F}$ the field with 5 elements. Take the permutations $(2,4)(3,5), (1,2,5)$ as generators for G and let H be the subgroup generated by $(3,5,4)$.*

This time we want to see how the permutation $(1,2,5)$ acts on $e_H \mathbb{F}G$. So we need Ω to be a complete set of representatives of right cosets for H in G. We take

$$\begin{aligned}\Omega = \{&(), (2,3)(4,5), (2,4,3), (2,5,3), (1,5,4,3,2), (1,5,4,2,3), (1,5,2,4,3),\\ &(1,5,3), (1,3,2), (1,3)(2,5), (1,3)(4,5), (1,3)(2,4), (1,2,5,4,3), (1,2,3),\\ &(1,2,4,5,3), (1,2)(4,5), (1,4,5,2,3), (1,4,2,5,3), (1,4,3), (1,4,5,3,2)\}.\end{aligned}$$

Now we need to calculate the orbits of H on Ω. We get

$$O_1 = \{()\},$$

$$O_2 = \{(2,3)(4,5), (2,5,3), (2,4,3)\},$$

$$O_3 = \{(1,5,4,3,2), (1,4,5,3,2), (1,3,2)\},$$

$$O_4 = \{(1,5,4,2,3), (1,4,2,5,3), (1,3)(2,4)\},$$

$O_5 = \{(1,5,2,4,3),(1,4,5,2,3),(1,3)(2,5)\},$

$O_6 = \{(1,5,3),(1,4,3),(1,3)(4,5)\},$

$O_7 = \{(1,2,5,4,3),(1,2,4,5,3),(1,2,3)\},$

$O_8 = \{(1,2)(4,5)\}.$

The action of $(1,2,5)$ *on* Ω *is represented by the permutation*

$$(1,13,5)(2,14,6)(3,15,7)(4,16,8)(9,11,10)(18,20,19).$$

We can calculate the c_{ij}*'s by processing this permutation. First we see that* $1 \mapsto 13$*, which means that* $()$ *in* O_1 *is mapped to* $(1,2,5,4,3)$ *in* O_7*, and we add* 1 *to* $c_{1,7}$*. Next,* $13 \mapsto 5$*, and we now add* 1 *to* $c_{7,3}$*. Continuing this process, we get the following matrix of values for* c_{ij} *where* $1 \leq i, j \leq 8$*.*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Since orbits 2 *through* 7 *have length* 3*, we multiply columns* 2 *through* 7 *by* $2 = 3^{-1}$ *in* $\mathbb{F}$ *to get the matrix*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 1 \\ 1 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

for $e_H(1,2,5)e_H$ *acting on* $e_H\mathbb{F}G$*.*

Similarly, we get the matrix

$$\begin{pmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

for $e_H(2,4)(3,5)e_H$ *acting on* $e_H\mathbb{F}G$.

In Example 2.5.2, we were looking at how an element of $e_H\mathbb{F}Ge_H$ acts on $e_H\mathbb{F}G$. The reason for considering this particular example is that we can write

$$e_H\mathbb{F}G = e_H\mathbb{F}Ge_H \oplus e_H\mathbb{F}G(1 - e_H)$$

since $1 = e_H + (1 - e_H)$. Now, $e_H\mathbb{F}Ge_H$ annihilates $e_H\mathbb{F}G(1 - e_H)$, so as a right $e_H\mathbb{F}Ge_H$-module, $e_H\mathbb{F}G = e_H\mathbb{F}Ge_H$.

2.6 Generating $e_H\mathbb{F}Ge_H$

In the previous section, we saw how to condense an $\mathbb{F}G$-module M and construct the elements e_Hge_H as they act on M. We now have to ask ourselves, how do we generate the algebra $e_H\mathbb{F}Ge_H$. The group algebra $\mathbb{F}G$ is generated as an algebra by elements corresponding to a generating set of G. We note that the matrices given in Example 2.5.2 for $e_H(2,4)(3,5)e_H$ and $e_H(1,2,5)e_H$ do not generate $e_H\mathbb{F}A_5e_H$ even though $(2,4)(3,5)$ and $(1,2,5)$ generate A_5. So $e_H\mathbb{F}A_5e_H$ is not a group algebra.

The following algorithm allows us to check that a subset of $e_H\mathbb{F}Ge_H$ is a generating set for the algebra.

Algorithm 2.6.1 (Spinning)**.** INPUT*: The identity element* $id_{e_H\mathbb{F}Ge_H}$ *in the right regular* $e_H\mathbb{F}Ge_H$*-module* $e_H\mathbb{F}Ge_H$ *and a set of elements* $\Upsilon = \{e_Hg_1e_H, \ldots, e_Hg_ne_H\}$ *in the algebra* $e_H\mathbb{F}Ge_H$.

COMPUTATION: *Let B be an $\mathbb{F}$-basis for $e_H\mathbb{F}Ge_H$, c a pointer to the current position in B, and C a temporary list of elements of $e_H\mathbb{F}Ge_H$.*

1. *Initialize B to be a list with the single element id_M. Set c to point at id_M in B. Initialize Υ to be an empty list.*

2. *For each element a of Υ, add the element $B[c]a$ to C.*

3. *Increment c.*

4. *For each u in C, check if u is linearly dependent on B. If not, add u to B.*

5. *Reset $C = []$.*

6. *If c is less than the length of B, then go to Step 2. Otherwise, stop.*

OUTPUT: *The $\mathbb{F}$-basis B for $e_H\mathbb{F}Ge_H$.*

Instead of using a temporary list C, we could have checked for linear dependence after every multiplication. We have not done this since for large matrices we do want to minimize the number of times we read them into memory.

To use the Spinning Algorithm (2.6.1) to verify that a subset of $e_H\mathbb{F}Ge_H$ generates the algebra, we run the algorithm on the subset and check the dimension of the $\mathbb{F}$-basis which is returned. The following proposition gives the dimension of $e_H\mathbb{F}Ge_H$ over $\mathbb{F}$.

Proposition 2.15. *Let H be a condensation subgroup of G. Then*

$$dim_{\mathbb{F}}(e_H\mathbb{F}Ge_H) = (1_H^G, 1_H^G).$$

Proof. The group G is a basis for $\mathbb{F}G$. It is then clear that coset representatives of H in G form a basis of $e_H\mathbb{F}G$. Now we see that the double coset representatives of H in G form a basis for $e_H\mathbb{F}Ge_H$. This leads to the fact that

$$dim_{\mathbb{F}}(e_H\mathbb{F}Ge_H) = |H\backslash G/H|,$$

the number of double H cosets in G. This tells us that the dimension of $e_H\mathbb{F}Ge_H$ is independent of the field $\mathbb{F}$.

So instead of calculating

$$dim_{\mathbb{F}}(e_H\mathbb{F}Ge_H) = dim_{\mathbb{F}}(\mathrm{End}_{\mathbb{F}G}(e_H\mathbb{F}G)),$$

we can just calculate

$$dim_{\mathbb{C}}(\mathrm{End}_{\mathbb{C}G}(e_H\mathbb{C}G)).$$

This is an easier calculation since by Maschke's Theorem (1.10), the group algebra $\mathbb{C}G$ is semisimple. So

$$\mathrm{End}_{\mathbb{C}G}(e_H\mathbb{C}G) \cong \dot{+}_{i=1}^{n} \dot{+}_{j=1}^{n} \mathrm{Hom}_{\mathbb{C}G}(S_i, S_j)$$

for some positive integer n, and for $1 \leq i \leq n$, the S_i's are simple $\mathbb{C}G$-modules. Each of the vectors spaces in this sum are either 0 or 1-dimensional. Therefore,

$$dim_{\mathbb{C}}(\mathrm{End}_{\mathbb{C}G}(e_H\mathbb{C}G))$$

is the number of simple $\mathbb{C}G$-modules appearing in the permutation module $e_H\mathbb{C}G$, which is measured by $(1_H^G, 1_H^G)$. □

The following theorem allows us to find a generating set $e_H\mathbb{F}Ge_H$ using a representation with dimension smaller than $dim_{\mathbb{F}}(e_H\mathbb{F}Ge_H)$.

Theorem 2.16. *Let N be a subgroup of G which normalizes H and suppose that N/H is a p-group generated by $\{n_1H, \ldots, n_lH\}$, with $l \in \mathbb{N}$. Let $f = \sum_{n\in N} n \in \mathbb{F}G$ and denote by $V = f\mathbb{F}G$ the permutation module of G on the cosets of N. Given elements $g_1, \ldots, g_k$, for some $k \in \mathbb{N}$, consider the subalgebra B of $e_H\mathbb{F}Ge_H$ generated by $e_Hg_1e_H, \ldots, e_Hg_ke_H$. Then the following statement holds. Suppose $f \cdot B = Ve_H$, then*

$$\{e_Hn_1e_H, \ldots, e_Hn_le_H, e_Hg_1e_H, \ldots, e_Hg_ke_H\}$$

is a generating set for the condensation algebra $e_H\mathbb{F}Ge_H$.

Proof. A proof of this theorem is found in [LW01] on page 169, or in [Lux97] on page 53. □

Note that by Theorem 2.16, we can replace the $e_H\mathbb{F}Ge_H$-module $e_H\mathbb{F}Ge_H$ with $f\mathbb{F}Ge_H$ in the Spinning Algorithm (2.6.1).

Chapter 3

Constructing the Basic Algebra

In the last chapter we looked at the condensation subalgebras of $\mathbb{F}G$ which are Morita equivalent to $\mathbb{F}G$. These algebras are important since we can construct them directly from G. There are algebras which are more suitable for working with on a computer called basic algebras. The purpose of this chapter is to introduce basic algebras and develop an algorithm for constructing basic algebras Morita equivalent to principal block of group algebras. The methods used in this chapter are valid for the entire group algebra, but we will specialize to the principal block, which is defined in Section 3.3.

For the rest of this chapter, we will let A be a finite dimensional algebra over a finite field $\mathbb{F}$. To simplify some statements, we assume that $\mathbb{F}$ is a splitting field of A. The first obvious consequence is that $\mathrm{End}_A(S) = \mathbb{F}$ for all simple A-modules S, see page 29.

Since A is finite dimensional it is also Artinian, by Proposition 1.1. Let r be the number of isomorphism classes of projective indecomposable A-modules. Then since A is Artinian, the right regular A-module can be decomposed as

$$A_A = \bigoplus_{i=1}^{r} \bigoplus_{j=1}^{n_i} P_{ij}, \tag{3.1}$$

for positive integers n_i and P_{ij} projective indecomposable A-modules with $P_{ij} \cong P_{kl}$ if and only if $i = k$. We will denote by P_i representatives of isomorphism classes of projective indecomposable A-modules. Let S_i denote $P_i/\operatorname{Rad}(P_i)$. By Corollary 1.7.4 in [Ben98] on page 13, the S_i's are representatives of isomorphism classes of simple A-modules.

By the Krull-Schmidt Theorem (1.12), r and $\{n_i \in \mathbb{Z}^+ : 1 \leq i \leq r\}$ in the

decomposition of A_A given by Equation 3.1 are unique, up to the order of summation. This allows us to use this decomposition of A_A for the remainder of this chapter.

3.1 Basic Algebras

As mentioned above, we want to construct the smallest algebra which is Morita equivalent to the $\mathbb{F}$-algebra A. By smallest, we mean in terms of the dimension as an $\mathbb{F}$-vector space. Since $A/J(A)$ is a semisimple algebra, we can use the Wedderburn-Artin Structure Theorem (1.9) to get

$$A/\,\mathrm{J}(A) \cong \dot{+}_{i=1}^{r} Mat_{n_i}(\mathbb{F}) \tag{3.2}$$

where r and the n_i are the same as in Equation 3.1. Since r is tied to the number of isomorphism classes of simple A-modules, Corollary 2.2 tells us that r is also the number of isomorphism classes of simple A'-modules for any algebra A' which is Morita equivalent to A. We see in Equation 3.2, that the sum of the integers n_i is the dimension of $A/\,\mathrm{J}(A)$. We show in this section that constructing an algebra with $n_i = 1$, for $1 \leq i \leq r$, gives the smallest algebra Morita equivalent to A.

Definition 3.1.1. *An algebra B over the field $\mathbb{F}$ is called a basic algebra if for every simple B-module S,* $\dim_{\mathrm{End}_B(S)}(S) = 1$.

Generally, we start with an algebra A which is not a basic algebra and we want to construct a basic algebra B which is Morita equivalent to A. Since it is not obvious how to do this from the definition of B, we examine the relationship between basic algebras and Morita equivalence.

The following lemma relates A-modules and basic algebras.

Lemma 3.1. *Let A be an algebra over a field $\mathbb{F}$ and P an A-module. Then* $\mathrm{End}_A(P)$ *is a basic algebra if and only if P decomposes into pairwise non-isomorphic indecomposable A-modules.*

Proof. A proof of this lemma is found in [Erd90] on page 5. □

The following corollary gives us a criterion for an algebra to be a basic algebra.

Corollary 3.2. *The algebra A is a basic algebra if and only if A_A decomposes into pairwise non-isomorphic indecomposable A-modules.*

Proof. This corollary follows directly from Lemma 3.1 since A_A is a projective A-module and $A \cong \mathrm{End}_A(A_A)$. □

We now use Lemma 3.1 to prove a uniqueness condition for basic algebras.

Proposition 3.3. *Two basic algebras are Morita equivalent if and only if they are isomorphic.*

Proof. Let A and B be Morita equivalent basic algebras. By Morita's Theorem (2.5), there is a progenerator $P \in \mathrm{Mod}_A$ such that $B \cong \mathrm{End}_A(P)$. Combining Lemma 3.1, Corollary 3.2, and the fact that P generates Mod_A, we see that $P \cong A_A$. Therefore

$$B \cong \mathrm{End}_A(P) \cong \mathrm{End}_A(A_A) \cong A.$$

The converse is trivial. □

This proposition proved that for an algebra A, a basic algebra B which is Morita equivalent to A is unique up to isomorphism. We will abuse language and say that B is the basic algebra Morita equivalent to A.

Corollary 2.4 tells us which algebras are Morita equivalent to A using progenerators. The following theorem describes the progenerator which gives the basic algebra Morita equivalent to A.

Theorem 3.4. *Let A be a finite dimensional algebra over $\mathbb{F}$. Then using the notation of Equation* 3.1*, $P = \dot{+}_{i=1}^{r} P_i$ is a progenerator for the category* Mod_A*, and* $\mathrm{End}_A(P)$ *is the basic algebra Morita equivalent to A.*

Proof. What we need to prove is that P is a progenerator for Mod_A. By Lemma 3.1, $\mathrm{End}_A(P)$ is a basic algebra and the Morita equivalence will follow from Corollary 2.4.

Let M be an A-module. By Lemma 1.15, M has a projective cover Q and there is an essential homomorphism $\phi\colon Q \twoheadrightarrow M$. We can use the Krull-Schmidt Theorem (1.12) to write $Q = \bigoplus_{i=1}^{r} m_i P_i$ where m_i is a nonnegative integer for $1 \leq i \leq r$. Let $k = \max\{m_i | 1 \leq i \leq r\}$. It is clear that there is an A-module epimorphism

$$\alpha\colon kP = \dot{+}_{i=1}^{r} kP_i \twoheadrightarrow \bigoplus_{i=1}^{r} m_i P_i = Q.$$

So $(\phi \circ \alpha)\colon kP \to M$ is an epimorphism and thus P is a progenerator for Mod_A. $\square$

Theorem 3.4 tells us that knowing the projective indecomposable A-modules is enough to realize the basic algebra Morita equivalent to A as an endomorphism ring. Let P be the progenerator $\dot{+}_{i=1}^{r} P_i$ for Mod_A, then

$$\begin{aligned} \mathrm{End}_A(P) &= \mathrm{End}_A(\dot{+}_{i=1}^{r} P_i) \\ &= \mathrm{Hom}_A(\dot{+}_{i=1}^{r} P_i, \dot{+}_{j=1}^{r} P_j) \\ &\cong \dot{+}_{i=1}^{r} \dot{+}_{j=1}^{r} \mathrm{Hom}_A(P_i, P_j). \end{aligned} \tag{3.3}$$

To make the isomorphism explicit, consider the natural projections $\pi_i\colon P \twoheadrightarrow P_i$ and natural injections $\iota_i\colon P_i \hookrightarrow P$. Let $\varphi_{ij} \in \mathrm{Hom}_A(P_i, P_j)$. Then we will use $\widehat{\varphi_{ij}}$ to denote the element $\iota_j \circ \varphi_{ij} \circ \pi_i$ in $\mathrm{End}_A(P)$.

Recall that we want to construct a generating set for the basic algebra Morita equivalent to A. The approach we take here is to find a basis for the complement of $\mathrm{Rad}^2(\mathrm{End}_A(P))$ in $\mathrm{End}_A(P)$ as an $\mathbb{F}$-vector space using Equation 3.3. We then use Lemma 1.7 to show that this basis will generate $\mathrm{End}_A(P)$ as an algebra. In order to do this, we need the following lemma.

Lemma 3.5. *Let A be a finitely generated algebra, and e a faithful idempotent of A with $P \cong eA$. Then* $\mathrm{Rad}^n(\mathrm{Hom}_A(P,P))$ *are the A-homomorphisms in* $\mathrm{Hom}_A(P,P)$

whose image is contained in $\mathrm{Rad}^n(P)$*, i.e.,* $\mathrm{Rad}^n(\mathrm{Hom}_A(P,P))$ *is isomorphic to* $\mathrm{Hom}_A(P, \mathrm{Rad}^n(P))$.

Proof. We know from Proposition 1.17 that

$$\mathrm{Rad}(eAe) = e\,\mathrm{J}(A)e.$$

We extend this to $\mathrm{Rad}^n(eAe)$ by induction on n. Suppose we have

$$\mathrm{Rad}^{n-1}(eAe) = e\,\mathrm{J}^{n-1}(A)e.$$

Then

$$\mathrm{Rad}^n(eAe) = \mathrm{Rad}^{n-1}(eAe)\,\mathrm{Rad}(eAe) = e\,\mathrm{J}^{n-1}(A)ee\,\mathrm{J}(A)e.$$

Since $\mathrm{J}(A)$ is a two-sided ideal in A,

$$e\,\mathrm{J}^{n-1}(A)ee\,\mathrm{J}(A)e = e\,\mathrm{J}^{n-1}(A)AeA\,\mathrm{J}(A)e.$$

By Theorem 2.6,

$$e\,\mathrm{J}^{n-1}(A)AeA\,\mathrm{J}(A)e = e\,\mathrm{J}^{n-1}(A)A\,\mathrm{J}(A)e = e\,\mathrm{J}^n(A)e.$$

Now we consider $\mathrm{Rad}^n(\mathrm{Hom}_A(P,P))$. By Theorem 1.19,

$$\mathrm{Rad}^n(\mathrm{Hom}_A(P,P)) = \mathrm{Rad}^n(\mathrm{Hom}_A(eA,eA)) \cong \mathrm{Rad}^n(eAe).$$

By the above argument,

$$\begin{aligned}\mathrm{Rad}^n(eAe) = e\,\mathrm{J}^n(A)e &\cong \mathrm{Hom}_A(eA, e\,\mathrm{J}^n(A)) \\ &= \mathrm{Hom}_A(eA, eA\,\mathrm{J}^n(A)) \cong \mathrm{Hom}_A(P, P\,\mathrm{J}^n(A)).\end{aligned}$$

Applying Lemma 1.16, we have

$$\mathrm{Hom}_A(P, P\,\mathrm{J}^n(A)) = \mathrm{Hom}_A(P, \mathrm{Rad}^n(P)),$$

which finishes the proof. □

We now examine $\mathrm{Hom}_A(P_i, P_j)$ for $1 \leq i, j \leq r$. First, we consider the short exact sequence

$$\mathrm{Rad}(P_j) \overset{\mu_j}{\hookrightarrow} P_j \overset{\tau_j}{\twoheadrightarrow} P_j/\mathrm{Rad}(P_j)$$

where μ_j is the natural injection and τ_j is the natural epimorphism. Since, by Proposition 1.21, the homomorphism functor $\mathrm{Hom}_A(P_i, -)$ is exact since P_i is a projective A-module, we get the sequence

$$\mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)) \overset{\mu_{j*}}{\hookrightarrow} \mathrm{Hom}_A(P_i, P_j) \overset{\tau_{j*}}{\twoheadrightarrow} \mathrm{Hom}_A(P_i, P_j/\mathrm{Rad}(P_j))$$

where μ_{j*} and τ_{j*} are postcomposition with μ_j and τ_j, respectively, is exact. Clearly, $\mathrm{Hom}_A(P_i, P_j/\mathrm{Rad}(P_j))$ is $\{0\}$ when $i \neq j$, since $P_j/\mathrm{Rad}(P_j)$ is simple. This leaves us with the exact sequence

$$\mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)) \overset{\mu_{j*}}{\hookrightarrow} \mathrm{Hom}_A(P_i, P_j) \overset{\tau_{j*}}{\twoheadrightarrow} 0$$

and μ_{j*} is an isomorphism. Let id_i denote the identity homomorphism on P_i. Then, when $i = j$, $\mathrm{Hom}_A(P_i, P_i/\mathrm{Rad}(P_i))$ is a 1-dimensional space generated by $\tau_{i*}(id_i)$.

We also look at the short exact sequence

$$\mathrm{Rad}^2(P_j) \hookrightarrow \mathrm{Rad}(P_j) \overset{\nu_j}{\twoheadrightarrow} \mathrm{Rad}(P_j)/\mathrm{Rad}^2(P_j)$$

where ν_j is the natural epimorphism. Again, we apply the functor $\mathrm{Hom}_A(P_i, -)$ to get the short exact sequence

$$\mathrm{Hom}_A(P_i, \mathrm{Rad}^2(P_j)) \hookrightarrow \mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)) \overset{\nu_{j*}}{\twoheadrightarrow} \mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)/\mathrm{Rad}^2(P_j)) \tag{3.4}$$

where ν_{j*} is postcomposition with ν_j. We let $n_{ij} \in \mathbb{N}$ denote the dimension of the vector space $\mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)/\mathrm{Rad}^2(P_j))$. Notice that n_{ij} is also the multiplicity of S_i in the semisimple quotient $\mathrm{Rad}(P_j)/\mathrm{Rad}^2(P_j)$. We choose a basis for

$$\mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)/\mathrm{Rad}^2(P_j))$$

and denote it by f_{ij}^k for $1 \leq k \leq n_{ij}$. Then we use $[f_{ij}^k]$ to denote a representative of the preimage $\nu_{j*}^{-1}(f_{ij}^k)$.

Using this notation we have set up, we now state the following theorem.

Theorem 3.6. *Let A be a finite dimensional algebra over $\mathbb{F}$, a splitting field for A, and $P = \dot{+}_{i=1}^r P_i$. Then $B = \mathrm{End}_A(P)$ is generated by the set*

$$\{e_j = \widehat{id_j} | 1 \leq j \leq r\} \bigcup \{b_{ij}^k = \widehat{[f_{ij}^k]} | 1 \leq k \leq n_{ij}, 1 \leq i, j \leq r\}.$$

Proof. Let B' be the subalgebra of B generated by the set

$$\{e_j = \widehat{id_j} | 1 \leq j \leq r\} \bigcup \{b_{ij}^k = \widehat{[f_{ij}^k]} | 1 \leq k \leq n_{ij}, 1 \leq i, j \leq r\}.$$

The approach we take in this proof is to show

$$B' + J^2(B) = B.$$

Then by Lemma 1.7, $B = B'$ and we will have a generating set for B as an algebra. We proceed in two steps, first we show that $B'/\operatorname{Rad}(B) = B/\operatorname{Rad}(B)$ and then $B'/\operatorname{Rad}^2(B) = B/\operatorname{Rad}^2(B)$.

Consider the quotient $B/\operatorname{Rad}(B)$. By Lemma 3.5, we have

$$B/\operatorname{Rad}(B) = \operatorname{Hom}_A(P,P)/\operatorname{Rad}(\operatorname{Hom}_A(P,P)) \cong \operatorname{Hom}_A(P,P)/\operatorname{Hom}_A(P, \operatorname{Rad}(P)).$$

Combining this with Equation 3.3, we get

$$B/\operatorname{Rad}(B) \cong \dot{+}_{j=1}^r \dot{+}_{i=1}^r \operatorname{Hom}_A(P_i, P_j)/\operatorname{Hom}_A(P_i, \operatorname{Rad}(P_j)).$$

By their construction the homomorphisms id_j for $1 \leq j \leq r$ are bases for the non-zero summands. The endomorphisms $e_j = \widehat{id_j}$ for $1 \leq j \leq r$ are linearly independent and so $\{e_j = \widehat{id_j} | 1 \leq j \leq r\}$ is a basis for the complement of $\operatorname{Rad}(B)$ in B.

Next, we show that

$$\{b_{ij}^k = \widehat{[f_{ij}^k]} | 1 \leq k \leq n_{ij}, 1 \leq i, j \leq r\}$$

is a basis for the complement of $\mathrm{Rad}^2(B)$ in $\mathrm{Rad}(B)$. By Lemma 3.5, these spaces are isomorphic to $\mathrm{Hom}_A(P, \mathrm{Rad}^2(P))$ and $\mathrm{Hom}_A(P, \mathrm{Rad}(P))$, respectively. Combining this with Equation 3.3, we get

$$\mathrm{Rad}(B)/\mathrm{Rad}^2(B) \cong \dot{+}_{j=1}^{r} \dot{+}_{i=1}^{r} \mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j))/\mathrm{Hom}_A(P_i, \mathrm{Rad}^2(P_j)).$$

By the construction of

$$\{f_{ij}^k | 1 \leq k \leq n_{ij}\}$$

for fixed $i, j \in \{1, ..., r\}$, this set is a basis for

$$\mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j)/\mathrm{Rad}^2(P_j))$$

which we see is isomorphic to

$$\mathrm{Hom}_A(P_i, \mathrm{Rad}(P_j))/\mathrm{Hom}_A(P_i, \mathrm{Rad}^2(P_j))$$

by considering the short exact sequence in Equation 3.4. Putting these sets together with the decomposition above, we get that

$$\{b_{ij}^k = \widehat{[f_{ij}^k]} | 1 \leq k \leq n_{ij}, 1 \leq i, j \leq r\}$$

is a basis for the complement of $\mathrm{Rad}^2(B)$ in $\mathrm{Rad}(B)$.

We now see that

$$\{e_j = \widehat{id_j} | 1 \leq j \leq r\} \bigcup \{b_{ij}^k = \widehat{[f_{ij}^k]} | 1 \leq k \leq n_{ij}, 1 \leq i, j \leq r\}$$

is an $\mathbb{F}$-vector space basis for the complement of $\mathrm{Rad}^2(B)$ in B. □

Theorem 3.6 proves that in order to realize B as an endomorphism algebra, we need to know the projective indecomposable A-modules. We will address how to gather this information in the next section.

3.2 Peakwords

We saw in the last section that constructing the basic algebra Morita equivalent to an algebra A requires the projective indecomposable A-modules. We use peakwords to construct the projective A-modules. In this section, we define peakwords and explain how they are used in this construction. A more thorough treatment of peakwords can be found in [Lux97] on pages 54 through 57.

Recall that $S_i = P_i/\operatorname{Rad}(P_i)$ is a simple A-module and let $D_i = \operatorname{End}_A(S_i)$ for $1 \leq i \leq r$.

Definition 3.2.1. *An element a_i in the $\mathbb{F}$-algebra A is called a peakword of A corresponding to S_i relative to $\{S_1, \ldots, S_r\}$ if the following conditions hold:*

1. $\operatorname{Ker}_{S_j}(a_i) = \{0\}$ *for every* $j \neq i$, $1 \leq j \leq r$,
2. $\dim_{\mathbb{F}}(\operatorname{Ker}_{S_i}(a_i)) = \dim_{\mathbb{F}}(D_i)$, *and*
3. $\operatorname{Ker}_{S_i}(a_i) = \operatorname{Ker}_{S_i}(a_i^2)$.

Finding peakwords takes some work since while a given algebra A has a decomposition, it is not always apparent from the representation of A. Fortunately, the program `pwkond` in the *C-MeatAxe* ([Rin00]) finds peakwords for A.

Recall that a projective indecomposable A-module P has the property that

$$P/\operatorname{Rad}(P) \cong S$$

where S is a simple A-modules. We call A-modules with this property S-local. The following theorem relates S-local A-modules to peakwords of A.

Theorem 3.7. *Let V be a faithful A-module and let $a \in A$. Then there exists a uniquely determined idempotent $e \in A$ that induces the Fitting decomposition of a on V, i.e., if $V = \operatorname{Ker}_V(a^k) \oplus \operatorname{Im}_V(a^k)$ is the Fitting decomposition of V with respect to the endomorphism induced by a, then $\operatorname{Ker}_V(a^k) = Ve$ and $\operatorname{Im}_V(a^k) = V(1-e)$. If a is*

a peakword of A corresponding to the composition factor S of V, then the A-module Ve is S-local.

Proof. A proof of this theorem is found in [Lux97] on page 57. □

If $e \in A$ is a primitive idempotent such that eA is S-local, we call e S-primitive. The idempotent e in Theorem 3.7 is S-primitive when a is a peakword in A. The following lemma shows us where to find $v \in V$ which will generate Ve.

Lemma 3.8. *Let V be an A-module and S a composition factor of V. Let π be an A-epimorphism from V onto the S-local A-module M and let e_S be an S-primitive idempotent. Then the image of $\pi(v)$ of $v \in Ve_S$ does not generate M if and only if $\pi(v) \in \mathrm{Rad}_A(M) \cap Me_S = \mathrm{Rad}_{e_SAe_S}(Me_S)$. Hence the subspace of vectors in v in Ve_S, for which $\pi(v)$ does not generate M, has codimension $\dim_{\mathbb{F}}(\mathrm{End}_A(S)) = \dim_{\mathbb{F}}(Se_S)$. Furthermore, if $M \cong P(S)$, the projective cover of S, then the ratio of vectors in Ve_S that do not generate an A-submodule isomorphic to $P(S)$ to the vectors in Ve_S is at most $\frac{1}{|\mathrm{End}_A(S)|}$.*

Proof. A proof of this theorem is found in [Lux97] on page 68. □

By this lemma, at most $\frac{1}{2}$ of the vectors in Ve_S won't generate an A-submodule isomorphic to $P(S)$.

Combining Theorem 3.7 and Lemma 3.8, we get a method for constructing a projective cover of the simple A-module S. Let a_S be a peakword in A for S, and let V be a faithful projective A-module. Often, we choose $V = A_A$, but that is not important for this discussion. Then for a random $v \in \mathrm{Ker}_V(a_S)$, $vA \cong P(S)$ with probability at least $\frac{1}{2}$. Since $\dim_{\mathbb{F}}(P(S))$ can be calculated independently, we can verify that $vA \cong P(S)$.

Now that we know how to construct the projective indecomposable $e_H\mathbb{F}Ge_H$-modules, we use the program `rad` in the *C-MeatAxe* ([Rin00]) to calculate their first and second Loewy layers. This allows us to find vectors in the complement of the

radical and radical squared. These vectors are used to construct the generating set described in Theorem 3.6 for the basic algebra which is Morita equivalent to $\mathbb{F}G$.

3.3 Blocks

As stated in the introduction to this chapter, we are interested mainly in group algebras. The reason for working with finite dimensional algebras in the last sections is that we can write a group algebra as a direct sum of finite dimensional subalgebras called blocks, which are not necessarily group algebras. This sum will help reduce the work required since the basic algebra Morita equivalent to a group algebra is the sum of the basic algebras Morita equivalent to these blocks. This section will define blocks for a finite dimensional algebra A and explore their relation to Morita equivalence.

Recall the decomposition of A_A given in Equation 3.1 on page 73. For each summand P_{ij} of A_A there is a primitive idempotent e_{ij} such that $e_{ij}A = P_{ij}$. The set

$$E = \{e_{ij} : 1 \leq i \leq r, 1 \leq j \leq n_i\}$$

is a complete set of pairwise orthogonal primitive idempotents for A, i.e.,

$$\sum_{e_{ij} \in E} e_{ij} = 1_A.$$

Definition 3.3.1. *Let P_1 and P_2 be two projective indecomposable A-modules. There are primitive idempotents e_1 and e_2 such that $e_1A \cong P_1$ and $e_2A \cong P_2$. We say P_1 and P_2 are linked if there exists a sequence of primitive idempotents $f_1, f_2, \ldots, f_n \in E$ such that $e_1 = f_1$, $e_2 = f_n$ and for each $1 \leq i \leq n-1$, f_iA and $f_{i+1}A$ share a composition factor. We call two primitive idempotents e_1 and e_2 linked if e_1A and e_2A are linked.*

The relation of being linked forms an equivalence relation on E. We use E_ℓ for $1 \leq \ell \leq m$ to denote the equivalence classes of E. The idea of being linked leads to the definition of blocks.

Definition 3.3.2. *The direct sum of the projective indecomposable A-modules $e_{ij}A$ for all e_{ij} belonging to an equivalence class E_ℓ of E, is called a block of A.*

The sum of two pairwise orthogonal primitive idempotents is again an idempotent. This leads to the following definition.

Definition 3.3.3. *Let E_ℓ be an equivalence class of E. We use u_ℓ to denote the sum of the idempotents in E_ℓ. The sums $u_1, \ldots, u_m$ of these partitions are called block idempotents.*

It is clear from the above definitions that $u_\ell A$ is a block of A for $1 \leq \ell \leq m$. We say an A-module M is in the block $u_\ell A$ if $Mu_\ell = M$.

The following theorem describes important properties of block idempotents and blocks of A.

Theorem 3.9. *Let A be a finite dimensional algebra. Then the block idempotents $u_1, \ldots, u_m$ of A are pairwise orthogonal centrally primitive idempotents with*

$$1_A = u_1 + \cdots + u_m.$$

Moreover, each block $u_i A$, with $1 \leq i \leq m$, is an indecomposable two-sided ideal and

$$A = u_1 A \oplus \cdots \oplus u_m A$$

is a unique decomposition of A into indecomposable two-sided ideals.

Proof. A proof of this theorem is found in [AF92] on page 100. □

Now that we have covered the basics of blocks, we can look at how this will help us to construct the basic algebra B Morita equivalent to a finite dimensional algebra A. Recall that we are realizing B as the endomorphism algebra of a progenerator P of Mod_A. From Theorem 3.4, we know that the progenerator we are interested in is $P = \dot{+}_{i=1}^{r} P_i$, where the $P_i's$ are representatives of the isomorphism classes of

projective indecomposable A-modules. We can now partition this sum according to the equivalence classes E_ℓ. By doing this we get $P = \dot{+}_{i=1}^{m} Q_i$, where $Q_i = Pu_i$ are projective A-modules. If $i \neq j$, Q_i and Q_j are in different blocks of A.

Suppose Q_i is in the block u_iA, then it follows that Q_i is a progenerator for Mod_{u_iA}. And by Theorem 3.4, $\mathrm{End}_{u_iA}(Q_i)$ is the basic algebra equivalent to u_iA. Clearly, $\mathrm{End}_{u_iA}(Q_i) = \mathrm{End}_A(Q_i)$ since $Q_i = Q_iu_i$.

The last piece we need for our calculation is that $\mathrm{Hom}_A(Q_i, Q_j) = 0$ for $i \neq j$. This follows from Theorem 3.9 since

$$\mathrm{Hom}_A(Q_i, Q_j) = \mathrm{Hom}_A(u_iP, u_jP) \cong u_jPu_i = u_ju_iP = 0$$

for $i \neq j$.

Now the following calculation,

$$\begin{aligned} B &= \mathrm{End}_A(P) \\ &= \mathrm{End}_A(\dot{+}_{i=1}^{m} Q_i) \\ &\cong \dot{+}_{i=1}^{m} \mathrm{End}_A(Q_i) \\ &= \dot{+}_{i=1}^{m} \mathrm{End}_{u_iA}(Q_i) \end{aligned}$$

shows us that the basic algebra B can be decomposed into blocks, which are also basic algebras, corresponding to the blocks of A. From this we see that we can construct the blocks of the basic algebra independently of each other.

3.4 Constructing Basic Algebras

We now focus on group algebras instead of arbitrary finite dimensional algebras. As before, $\mathbb{F}$ is a finite field of characteristic p, and p divides the order of the finite group G. We also assume that $\mathbb{F}$ is the splitting field of $\mathbb{F}G$. Lastly, $\mathbb{F}G$ is a finite dimensional algebra over $\mathbb{F}$, so everything we have done in this chapter applies to $\mathbb{F}G$.

Before we get started, it should be noted that every group algebra $\mathbb{F}G$ has a one-dimensional simple $\mathbb{F}G$-module with a trivial action by G which is called the trivial $\mathbb{F}G$-module. We use this module to make the following definition.

Definition 3.4.1. *Let u_1 be the block idempotent such that $u_1\mathbb{F}G$ is the block of $\mathbb{F}G$ which contains the trivial $\mathbb{F}G$-module. We call this block the principal block of $\mathbb{F}G$.*

In the last section, we saw that we can calculate the basic algebra Morita equivalent to the principal block of $\mathbb{F}G$ independently of any other blocks of $\mathbb{F}G$. For the purposes of this dissertation, we restrict our calculations to principal blocks. We do this partly because the principal block tends to be the most complex block of $\mathbb{F}G$, but also partly as a matter of convenience. To collect the projective indecomposable $e_H\mathbb{F}Ge_H$-modules in a block B, we pick one projective indecomposable $e_H\mathbb{F}Ge_H$-module P in B, and find all the projective indecomposable $e_H\mathbb{F}Ge_H$-modules which are linked to P. For the principal block, we let P be the projective cover of the trivial $e_H\mathbb{F}Ge_H$-module, which is easy to identify. Identifying an $e_H\mathbb{F}Ge_H$-module in another block is more difficult, and has not yet been implemented.

We started this project under the assumption that $\mathbb{F}G$ is too large as an $\mathbb{F}$-vector space to work with efficiently. Generally, $u_1\mathbb{F}G$ is still large. However, we defined condensation for finite dimensional algebras, which $u_1\mathbb{F}G$ clearly is. One of the benefits of using condensation with group algebras is that the idempotent used in condensation can be defined purely in terms of G. By Corollary 2.9, we find possible condensation subgroups by decomposing their permutation characters, 1_H^G, in terms of G's indecomposable projective characters. While $u_1\mathbb{F}G$ may not be a group algebra, since u_1 is a sum of primitive idempotents in $\mathbb{F}G$, $u_1\mathbb{F}G$ is a sum of projective indecomposable $\mathbb{F}G$-modules. So we can still find condensation subgroups for $u_1\mathbb{F}G$ using the projective indecomposable characters of G. Our requirement now becomes that the permutation character 1_H^G contains projective indecomposable summands corresponding to the projective covers of the simple $u_1\mathbb{F}G$-modules. Since we have reduced the

requirements on the subgroup H, we generally get a larger selection.

Combining the results of this chapter with condensation from the previous chapter, we get the following algorithm for computing the basic algebra Morita equivalent to the principal block of a group algebra.

Algorithm 3.4.1 (Constructing the Basic Algebra of the block $u\mathbb{F}G$ of $\mathbb{F}G$)**.** INPUT*: The group G and prime number p which divides the order of G.*
COMPUTATION*:*

1. *Find a condensation subgroup H for G which gives a Morita equivalence between $u\mathbb{F}G$ and $e_H u\mathbb{F}Ge_H$.*

2. *Find a generating set for the condensation subalgebra $e_H\mathbb{F}Ge_H$.*

3. *Construct the projective indecomposable $e_H u\mathbb{F}Ge_H$-modules using peakwords.*

4. *Construct a generating set for the basic algebra Morita equivalent to $e_H u\mathbb{F}Ge_H$ using Theorem* 3.6*.*

OUTPUT*: The generators of the basic algebra of $u\mathbb{F}G$ as matrices acting on representatives of the isomorphism classes of projective indecomposable modules.*

The implementation of this algorithm is explained in detail in Section 4.2, beginning on page 95.

Consider the following example where G is A_5 and $\mathbb{F}$ is the field with 4 elements.

***Example* 3.4.1.** *The only possible condensation subgroup for $\mathbb{F}A_5$ is the trivial subgroup. So we do not need condensation in this example. Since $\mathbb{F}A_5$ is a group algebra, it is generated by elements corresponding to a generating set of A_5. The principal block of $\mathbb{F}A_5$ has 3 projective indecomposable modules up to isomorphism. Their first and second Loewy layers are*

$$\begin{array}{ccc} 2a & 1a & 2b \\ 1a & 2a \quad 2b & 1a \end{array}$$

If we label these modules by their heads, then the generating set returned by Theorem 3.6 *is*

$$\{e_{2a}, e_{1a}, e_{2b}, b_{2a1a}, b_{2b1a}, b_{1a2a}, b_{1a2b}\}.$$

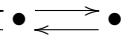
3.5 Ext-Quivers

The generating set given by Theorem 3.6 for a basic algebra B gives a nice graphical presentation related to B. In this section, we will see how to construct these graphs from the generating set and describe their connection with the functor Ext^1_B.

Definition 3.5.1. *A quiver is a directed graph, that means a set of vertices and a list of ordered pairs of vertices describing arrows between them.*

The following is a simple example of a quiver.

***Example* 3.5.1.**

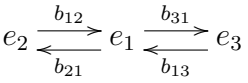


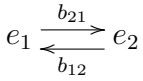
Recall that a generating set for a basic algebra Morita equivalent to a group algebra as given by Theorem 3.6 is of the form

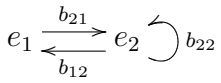
$$\{e_j = \widehat{id_j} | 1 \leq j \leq r\} \bigcup \{b_{ij}^k = \widehat{[f_{ij}^k]} | 1 \leq k \leq n_{ij}, 1 \leq i, j \leq r\}.$$

We can construct a quiver from this set by taking $\{e_j = \widehat{id_j} | 1 \leq j \leq r\}$ as our vertex set and for each b_{ij}^k in our generating set, put the ordered pair (e_j, e_i) into our list of arrows. The following example makes this idea clear.

***Example* 3.5.2.** *Quivers for the principal block of the alternating group A_5 are:*

characteristic 2 

characteristic 3 

characteristic 5 

We have drawn the arrows in the direction opposite to the homomorphisms. Recall that $b_{ij} \in \mathrm{Hom}_A(P_i, P_j)$ for P_i and P_j projective A-modules. This allows us to read off the labels of a path in a quiver to get an element in B. Examining characteristic 2 of Example 3.5.2, we have the path $b_{12}b_{31}$. While this is a path from e_2 to e_3 in the quiver, it corresponds to an element in B which goes from e_3 to e_2.

These graphs store information about the basic algebra B. By Lemma 1.26, we see that

$$\begin{aligned}\dim_{\mathbb{F}}(\mathrm{Ext}^1_B(S_1, S_2)) &= \dim_{\mathbb{F}}(\mathrm{Hom}(P(\mathrm{Rad}(P(S_1))/\mathrm{Rad}^2(P(S_1))), S_2)) \\ &= \dim_{\mathbb{F}}(\mathrm{Hom}(\mathrm{Rad}(P(S_1))/\mathrm{Rad}^2(P(S_1)), S_2)) \\ &= \dim_{\mathbb{F}}(\mathrm{Hom}(S_2, \mathrm{Rad}(P(S_1))/\mathrm{Rad}^2(P(S_1))))\end{aligned}$$

for S_1, S_2 simple B-modules. This fact implies that $\dim_{\mathbb{F}}(\mathrm{Ext}^1_B(S_1, S_2))$ is the number of arrows from e_1 to e_2 in the quiver we constructed out of the generating set from Theorem 3.6. This observation leads to the following definition.

Definition 3.5.2. *Let $S_i \cong Ae_i/\mathrm{Rad}(A)e_i$ for $i = 1, \ldots, r$, be representatives for the isomorphism classes of simple A-modules. The* Ext*-quiver of A, $Q(A)$, is the quiver with vertices x_i corresponding to S_i, and the number of arrows from x_i to x_j is $\dim_{\mathbb{F}} \mathrm{Ext}^1_A(S_i, S_j)$.*

In addition to giving us a nice pictorial presentation of the generating set of B, Ext-quivers also lead to a presentation of B as the quotient of a path algebra.

Definition 3.5.3. *Let Q be a quiver and F a field. Then the path algebra FQ is the algebra over F which as a vector space has all paths in Q as a basis. Multiplication of basis elements in FQ corresponds to concatenation of paths in Q with $ab = 0$ if the path described by b does not end at the vertex which begins the path described by a for $a, b \in FQ$. We extend this multiplication by linearity to the entire space.*

We saw in Example 3.5.2 that we can realize the elements of the basic algebra B as paths in a quiver. This suggests that we can realize B as a quotient of a path algebra. This is proven in the following theorem..

Theorem 3.10. *Suppose B is a finite dimensional basic algebra over a splitting field F, and let $Q = Q(B)$ be its* Ext*-quiver. Then there is a surjective map of algebras $\phi\colon FQ \twoheadrightarrow B$ such that the kernel of ϕ is contained in the ideal of paths of length at least two.*

Proof. A proof of this theorem is found in [Ben98] on page 103. □

The proof of Theorem 3.10 given in [Ben98] is similar to the proof of Theorem 3.6 on page 79 except that by using the path algebra of the Ext-quiver, the setting is simplified.

Chapter 4

Implementation

So far we have seen several algorithms and the theory on which they are based. In this chapter, we discuss the implementation of these algorithms using the computer algebra system *GAP*, [GAP02], and the collection of programs known as the *C-MeatAxe*, [Rin00]. In addition to these two packages, we will be using programs written by Klaus Lux documented in [Lux97], and programs written by the author. We begin this chapter by giving brief descriptions of these programs.

4.1 *GAP* Programs

In this section, we describe a couple of the tools in *GAP* of which we make heavy use. Following that, we give brief descriptions of the main programs written by the author.

4.1.1 *GAP* Tools

The first tool we discuss is straight line programs. Straight line programs define an efficient way to evaluate a word on specific generators. The point is that at each step, we can use any of the subwords calculated in the previous steps. For example, consider the word $ababbab$. We can construct this word by first computing $c = ab$ and $d = cb$, then the word is cdc. This method only used 4 multiplications whereas the straightforward method requires 6 multiplications. With longer words, the savings are greater.

An important application of straight line programs is describing elements of G in terms of the generators. Table 5.2 on page 111 lists the standard words $z_1, \ldots, z_{11}$. These words are defined in *GAP* using straight line programs. Standard words are

useful when we search for generators of $e_H\mathbb{F}Ge_H$, since they allow us to construct elements of G, and hence $e_H\mathbb{F}Ge_H$, which are easy to describe and reconstruct.

Another very important feature of *GAP* is its implementation of Burnside's table of marks.

Definition 4.1.1. *Let G be a finite group.*

1. *Let X be a finite G-set and let U be a subgroup of G. The mark $\beta_X(U)$ of U on X is defined as*

$$\beta_X(U) = |\operatorname{Fix}_X(U)|$$

   where $\operatorname{Fix}_X(U) = \{x \in X | x \cdot u = x \text{ for all } u \in U\}$ is the set of fixed points of the subgroup U in the action of G on X.

2. *The table of marks of G is the square matrix*

$$M(G) = (\beta_{G_i \backslash G}(G_j))_{i,j}$$

   where both G_i and G_j run through the system of representatives of the conjugacy classes of subgroups of G.

In [Pfe97], Götz Pfeiffer describes an algorithm for constructing tables of marks and the information that can be deduced from the entries in the table. During the construction of the table, data is produced which is not stored in table, such as generators for representatives of the conjugacy classes of subgroups. As a theoretical object, the table of marks does not contain any information about how to construct representatives. Fortunately, the implementation of the table of marks in *GAP* does contain the extra data from the construction of the table. This means, in particular, that if we want to construct the subgroup with number n in the table of marks of G, the *GAP* command `RepresentativeTomByGeneratorsNC` will do this. We need to remember that the subgroups returned are representatives of conjugacy classes, and not every group has a table of marks in *GAP*'s database. The *GAP* command

`AllLibTomNames` returns a list containing the group names for groups with a table of marks in *GAP*'s database.

4.1.2 Program Descriptions

In this section, we give brief descriptions of the main programs written by the author which are used in the implementation of Algorithm 3.4.1. This collection of programs will be submitted to the *GAP* council for consideration to become a *GAP* share package. The code for some of the programs written by the author which are referenced here are included in Appendix A.

AutoCalcBasic is the automated calculation of the basic algebra for the principal block of a group algebra. The input is the name of a group and the characteristic of $\mathbb{F}$. It requires that *GAP* has access to the table of marks as well as the character table and Brauer character table for the group. This program returns both the information for the Ext-quiver as well as the matrices in $\mathrm{Hom}_{u_1\mathbb{F}G}(P_i, P_j)$ corresponding to the arrows in the quiver.

Condense constructs the matrix for the element $e_H g e_H$ for $g \in G$ as it acts on the basis for an $e_H\mathbb{F}Ge_H$-module M. The input required is permutations for the generators for G and H, and the elements to condense. All of these permutations need to be for the action on the cosets of G/H. This program returns the matrix in sparse format when applicable. The *GAP* code for this program is in Appendix A.4.

CondenseModuleTom constructs the generators of $e_H\mathbb{F}Ge_H$ as they act on a condensed permutation module given by a subgroup in the table of marks. The input is the name of the group, the number of the subgroup, and straight line programs for the generators of the condensation algebra. The *GAP* code for this program is in Appendix A.5.

FindAlgGens searches for a generating set of the condensation subalgebra. The input is the group name and permutations for the group and condensation subgroup. This program will take advantage of Theorem 2.16 when it is appropriate. This program returns a description of the generating set which can be used along with straight line programs to rebuild the generators when desired. The *GAP* code for this program is in Appendix A.3.

FindBasicHoms constructs the homomorphisms corresponding to the arrows in the Ext-quiver. The input is the group name and the *C-MeatAxe* representation name. The program requires that the projective indecomposable modules in the principal block of $e_H\mathbb{F}Ge_H$ have been constructed. The *GAP* code for this program is in Appendix A.6.

FindCondSubgroup finds the largest condensation subgroup. The input is the group name and the characteristic of $\mathbb{F}$. To make use of Theorem 2.16, this program finds the conjugate of the condensation subgroup inside the representative of the conjugacy class of its normalizer returned by the *GAP* command `NormalizerTom` and also returns the straight line programs to construct the normalizing p-elements described in Theorem 2.16. This program also uses the table of marks as well as the character table and Brauer character table for the group. The *GAP* code for this program is in Appendix A.1.

GetPerms finds the permutations for G acting on the cosets of the condensation subgroup H. These permutations are what we need for condensation. The input is the group name and condensation information returned by the commands `InfoKond` and `FindCondSubgroup`. This program will first try to use representations of G retrieved by the *GAP* package `atlasrep` along with the *C-MeatAxe* program `zvp`. The idea is to find a vector v fixed by H and moved by every subgroup containing H. The permutation of the matrix of $g \in G$ acting on v is

the permutation of g acting on the right cosets of H in G up to ordering.

If an appropriate vector is not found, `GetPerms` then switches to the program `ImprimPerm`, in [Lux97] on page 138. The *GAP* code for this program is in Appendix A.2.

4.2 Implementing Construction of the Basic Algebra of the Principal block of $\mathbb{F}G$

The structure of this section follows Algorithm 3.4.1 on page 87, since this is the algorithm which leads to the results in the following chapter. This algorithm takes a finite group G and a prime p which divides the order of G as input. Recall that $\mathbb{F}$ is a field of characteristic p which is a splitting field for $\mathbb{F}G$. For our examples, we use the alternating group on 5 letters, A_5, for G and the prime number 3 for p. In this situation, $\mathbb{F}$ is GF(3).

All of our calculations will be using the principal block of $\mathbb{F}G$, so we simplify our notation and denote the principal block of $\mathbb{F}G$ by B_0.

Step 1. Find a condensation subgroup H for G which gives a Morita equivalence between B_0 and $e_H B_0 e_H$. We need to find a condensation subgroup H of G. We use Corollary 2.9 to find the possible condensation subgroups of G. The program `FaithfulKondBurn` from [Lux97] on page 135 uses the character tables and table of marks given in *GAP* to find all subgroups of G which lead to faithful condensation. The program `InfoKond`, also written by Lux, returns the condensation information for the largest of the condensation subgroups. We are interested in the largest condensation subgroup since this leads to the smallest condensation subalgebra.

```
gap> grpinfo:=InfoKond("A5",3,1);
[ 6, 9 ]
rec( permchar := [ 15, 3, 0, 0, 0 ], norm := [ [ 6 ] ],
  numberinburn := 4, cartan := [ [ 2, 1 ], [ 1, 2 ] ],
  dimmodchars := [ 1, 4 ], splitf := 3,
```

```
  dimfixmod := [ 1, 1 ], dimfixproj := [ 3, 3 ] )
```

Notice from this output that the subgroup of G with number 4 in the table of marks for G is the largest condensation subgroup for G, up to conjugation. Since the table of marks does not distinguish between conjugate subgroups, we run into an issue when we try to use Theorem 2.16 on page 71. The Normalizer of H in G, call it N, can be found using the table of marks and the command `NormalizerTom`. The issue is that we don't know that H is contained in N, since they are both just representatives of conjugacy classes. To deal with this issue, the author has written a program, `FindCondSubgroup` on page 180, which finds words in the generators of N to generate a conjugate of H.

```
gap> subinfo:=FindCondSubgroup("A5",3);
[ 6, 9 ]
rec( SimpleRep := 5, Normalizer := 8, kdim := 2, knumber := 8,
  gens := [ [ <straight line program>, <straight line program> ],
      [ <straight line program> ] ],
  strline := [ [ "aba^{2}", "a^{2}ba" ], [ "a" ] ],
  horder := 4, korder := 12, script := true )
```

The first element of `gens` is a list of straight line programs in the generators of the subgroup corresponding to number `Normalizer` in the table of marks of G. The second element is a list of the normalizing elements described in Theorem 2.16. The entry `strline` is the same elements as `gens` except in human readable strings which are given purely for our benefit. All calculations should be done using the straight line programs for the reasons stated in Section 4.1.1.

The output from `FindCondSubgroup` gives us the information necessary to construct the condensation subgroup H and any elements of order p which normalize H, if they exist.

Step 2. Find a generating set for the condensation subalgebra $e_H\mathbb{F}Ge_H$. The idea for this step is to take a set of elements in G, construct the corresponding elements in

$e_H\mathbb{F}Ge_H$. Then we can use the Spinning Algorithm (2.6.1) to check if they generate the algebra $e_H\mathbb{F}Ge_H$. If they don't generate, we condense more elements of G, add them to our list of generators and start the spinning again.

To find a generating set, we need to know if there are normalizing elements as described by Theorem 2.16. This will affect the $e_H\mathbb{F}Ge_H$-module used to verify generation. Let K be the subgroup of G generated by H and any normalizing elements identified by `FindCondSubgroup`. If there are no normalizing elements, then $K = H$. Let $\bar{K}$ denote the sum of the elements in K. So the $e_H\mathbb{F}Ge_H$-module used to verify generation is $\bar{K}\mathbb{F}Ge_H$. If $K = H$, then

$$\bar{K}\mathbb{F}Ge_H = e_H\mathbb{F}Ge_H$$

since $\frac{1}{|H|} \in \mathbb{F}$. Note that $\bar{K}\mathbb{F}Ge_H$ is generated by $\bar{K} \in e_H\mathbb{F}Ge_H$.

Now that we have fixed our $e_H\mathbb{F}Ge_H$-module for spinning, we need to be able to condense an element g of G, with respect to its action on $\bar{K}\mathbb{F}Ge_H$. The first step is to construct a permutation describing the action of g on the cosets G/K. There are two programs for this purpose. The *GAP*-procedure `ImprimPerm`, in [Lux97] on page 138, uses transversals in a straight forward manner to build these permutations. While this is quick for small groups, this method is slow for long transversals. The author has written a *GAP*-procedure `GetPerms`, on page 183, which constructs permutations using representations from [W$^+$04] via the `atlasrep` package of *GAP*, and the *C-MeatAxe* program `zvp`.

```
gap> perms:=GetPerms("A5",grpinfo,subinfo);
Vector 1: Orbit size = 15
Vector 1: Orbit size = 5
[ [ 2,
      [ (2,4)(5,6)(7,10)(8,14)(11,12)(13,15), (1,2,5)(3,8,
            15)(4,10,14)(6,13,12)(7,11,9),
          (2,6)(4,5)(7,12)(8,13)(10,11)(14,15),
          (2,5)(4,6)(7,11)(8,15)(10,12)(13,14),
          (1,3,9)(2,7,8)(4,11,13)(5,12,14)(6,10,15) ] ],
  [ 2,
```

```
[ (2,3)(4,5), (1,2,4), (2,5)(3,4), (2,4)(3,5), (3,4,
      5) ] ] ]
```

Notice that `GetPerms` returned two list of permutations. The first set is for G/H and the second is for G/K to be used with Theorem 2.16. This is done since even though we check generation using $\bar{K}\mathbb{F}Ge_H$, we need to construct the projective indecomposable $e_H\mathbb{F}Ge_H$-modules later.

We use these permutations to condense elements of G. The *GAP*-procedure `Condense` on page 189, also written by the author, takes these permutations and a list of elements in G to be condensed. This version of condensation is specialized to take advantage of the row sparse property of the condensed matrices e_Hge_H for some $g \in G$. This property and its implications are discussed in Section 4.3. Since we want to minimize the number of elements condensed, the program `Condense` is called from within `FindAlgGens`, which is described below.

Knowing how to condense elements of G, we can address which elements of G are condensed. First, we condense the normalizing elements, if there are any, described by `FindCondSubgroup`. Since we want our results to be reproducible, we use the standard words $z_1, \ldots, z_{11}$ of G described in Table 5.2 evaluated on the generators of G. If this set of elements doesn't generate $e_H\mathbb{F}Ge_H$, we use the standard words $z_1, \ldots, z_{11}$ evaluated on previous on previous standard words. We continue making new elements in this manner until we find a generating set for $e_H\mathbb{F}Ge_H$. Once we have found a generating set, we use the Spinning Algorithm to remove redundant generators. This is accomplished by removing one element and checking if the rest of the set generates the algebra. If it does, throw away the removed element and start again, otherwise replace the element and try removing another. The author's *GAP*-procedure `FindAlgGens`, on page 185, implements this process of finding a generating set for the condensation algebra.

```
gap> alggens:=FindAlgGens("A5",3,perms[1],perms[2],
>       subinfo.kdim,Length(subinfo.gens[2]));
```

```
[ 6, 9 ]
[ rec( genlist := [ 2 ],
      basedim := [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
      resstrline := [ (2,3)(4,5), (1,2,4), (1,2,3,4,5),
          (1,3,5,4,2), (2,5,3), (1,5)(3,4), (1,2)(4,5),
          (1,2,4), (1,3)(2,4), (2,4)(3,5), (2,4,3) ],
      bigresstrline :=
        [ (2,4)(5,6)(7,10)(8,14)(11,12)(13,15),
          (1,2,5)(3,8,15)(4,10,14)(6,13,12)(7,11,9),
          (1,2,10,11,6)(3,8,4,5,13)(7,14,15,12,9),
          (1,14,6,5,8)(2,9,4,12,11)(3,10,13,15,7),
          (1,9,3)(2,13,10)(4,8,12)(5,15,11)(6,14,7),
          (1,13)(3,12)(4,15)(5,10)(6,9)(11,14),
          (1,7)(2,3)(5,12)(6,15)(8,9)(11,13),
          (1,2,5)(3,8,15)(4,10,14)(6,13,12)(7,11,9),
          (1,4)(2,15)(3,14)(5,7)(8,11)(9,10),
          (2,5)(4,6)(7,11)(8,15)(10,12)(13,14),
          (1,3,9)(2,11,14)(4,7,15)(5,10,8)(6,12,13) ] ) ]
```

The program `FindAlgGens` returns a list of records where each record contains two sets of 11 permutations corresponding to the words in Table 5.2 evaluated on the permutations returned by `GetPerms`. The records also include which of these elements are in the generating set in the variable `genlist`.

Step 3. Construct the projective indecomposable $e_H\mathbb{F}Ge_H$-modules using peakwords. Knowing generators for $e_H\mathbb{F}Ge_H$, we can begin to construct the projective indecomposable $e_H\mathbb{F}Ge_H$-modules using peakwords. The first step is to find a set of peakwords for the simple $e_H\mathbb{F}Ge_H$-modules. The regular representation of $e_H\mathbb{F}Ge_H$ contains all of the simple $e_H\mathbb{F}Ge_H$-modules and could be used to find peakwords. The problem is that we would have to use the *C-MeatAxe* program `chop` on the regular representation, which is time consuming, if not impractical, for large modules. The record returned by `FindCondSubgroup` has an entry named `SimpleRep`. This entry is the number in the table of marks of a subgroup L which corresponds to an $e_H\mathbb{F}Ge_H$-module, the permutation module of G on the cosets of L, which contains all of the

simple $e_H\mathbb{F}Ge_H$-modules as composition factors. This is the smallest $e_H\mathbb{F}Ge_H$-module which has this property and is easily constructed using the table of marks and the author's program `Condense`. We construct this module corresponding to L using the authors' *GAP*-procedure `CondenseModuleTom` found on page 191.

```
gap> CondenseModuleTom( "A5", 3, subinfo.SimpleRep,
>                grpinfo.numberinburn, alggens, subinfo);
Vector 1: Orbit size = 12
```

The procedure `CondenseModuleTom` does not return anything in *GAP*, but instead constructs the matrices for the module in the directory described by the variable `BasicWorkingSpace`.

```
gap> BasicWorkingSpace;
dir("/tmp/filepf2rKh.1/")
```

Now we call the *C-MeatAxe* commands `chop` and `pwkond` from within *GAP* to get peakwords for the simple $e_H\mathbb{F}Ge_H$-modules. Once we have these peakwords, we can use Theorem 3.7 and Lemma 3.8 to construct projective covers for the simple $e_H\mathbb{F}Ge_H$-modules. Applying the Spinning Algorithm (2.6.1) to a kernel vector of a peakword constructs the space mentioned in Theorem 3.7. The Spinning Algorithm is implemented in the *C-MeatAxe* program `zsp`. The problem we run into with this implementation is that all of the matrices for the generators must fit into memory. The author has modified this program to take matrices in sparse format, as discussed in Section 4.3.2, which are much smaller and so more generators fit into memory.

Step 4. Construct a generating set for the basic algebra Morita equivalent to $e_H\mathbb{F}Ge_H$ using Theorem 3.6. The last step is to construct a generating set for the basic algebra as described in Theorem 3.6. The author has written the *GAP*-procedure `FindBasicHoms`, on page 195, to build the quiver information.

```
gap> algebra:=FindBasicHoms("A5",[ "1a", "1b" ],"x","y",grpinfo);
rec( group := "A5",
```

```
generators := [ "1a", "1b", "1a1b1", "1b1a1" ],
npims := 2, pimnames := [ "1a", "1b" ],
cartan := [ [ 2, 1 ], [ 1, 2 ] ], field := GF(3),
dim := [ 3, 3 ], adjmat := [ [ 0, 1 ], [ 1, 0 ] ],
1a := rec( start := 1, ende := 1, name := "id1a",
    mat :=
      [ [ Z(3)^0, 0*Z(3), 0*Z(3) ], [ 0*Z(3), Z(3)^0,
            0*Z(3) ], [ 0*Z(3), 0*Z(3), Z(3)^0 ] ] ),
1b := rec( start := 2, ende := 2, name := "id1b",
    mat :=
      [ [ Z(3)^0, 0*Z(3), 0*Z(3) ], [ 0*Z(3), Z(3)^0,
            0*Z(3) ], [ 0*Z(3), 0*Z(3), Z(3)^0 ] ] ),
1a1b1 := rec( start := 2, ende := 1, name := "1a1b1",
    mat :=
      [ [ Z(3)^0, Z(3)^0, Z(3)^0 ], [ Z(3), Z(3)^0,
           0*Z(3) ], [ Z(3)^0, Z(3), 0*Z(3) ] ] ),
1b1a1 := rec( start := 1, ende := 2, name := "1b1a1",
    mat :=
      [ [ Z(3), Z(3)^0, 0*Z(3) ], [ Z(3), 0*Z(3), Z(3) ]
          , [ Z(3), Z(3)^0, 0*Z(3) ] ] ) )
```

The record returned above contains entries for each of the homomorphisms corresponding to generators of the basic algebra. Compare this data to Figure 5.2 on page 112. Notice that these elements are given in the path algebra, not the endomorphism ring. The `start` and `ende` entries give the information we need for composition in the path algebra. So in addition to containing the information needed to construct the Ext-quiver of the basic algebra of B_0, we have matrices for each of the arrows in the Ext-quiver.

4.3 Sparse Matrices

At several points in the previous section, sparse matrices were mentioned. In this section, we discuss the observation that the elements $e_H g e_H$ of the condensation subalgebra for some $g \in G$ are sparse. The implications of this observation will also be covered.

4.3.1 Observation and Implication

Recall the result of Lemma 2.12 on page 65. The last part of this lemma tells us that at most $|H|$ of the $c_{ij}\frac{1}{|O_j|}$ are nonzero for a fixed j. Since the order of H is, in general, much smaller than the dimension of $e_H\mathbb{F}Ge_H$, we call the matrices for e_Hge_H row-sparse matrices. We take advantage of this property in our practical implementation. Before we discuss how this is implemented, we briefly review how the *C-MeatAxe*, as in [Rin00], stores matrices.

Suppose $\mathbb{F}$ is a field of order $p^k = q$ with p a prime number. The *C-MeatAxe* uses a q-adic packing scheme to store finite field elements into bytes. First, we find a number n such that $q^n \leq 256$. We can store numbers of the form $a_{n-1}q^0+a_{n-2}q^1+\cdots+a_0q^{n-1}$, where $a_1 \in \mathbb{F}$ for all $1 \leq i \leq n$, in a single byte. To see that this number is less than 256, we take the largest possible value for each of the a_i's, which is $q-1$, and compute the sum

$$(q-1)(q^0+\cdots+q^{n-1}) = (q^1+\cdots+q^n)-(q^0+\cdots+q^{n-1}) = q^n-1 < 256.$$

Using this q-adic packing, n elements of $\mathbb{F}$ into each byte. So to store an $r \times c$ matrix using the $C-MeatAxe$, we need $r\lceil\frac{c}{n}\rceil$ bytes. Notice that the largest possible value for n is 8 which occurs for $q = 2$. For $q = 3$, n is down to 5. So for larger fields, $r\lceil\frac{c}{n}\rceil$ is close to rc.

Suppose we have a $r \times c$ row-sparse matrix with at most h nonzero entries per row. Instead of storing all the matrix entries, we just store the nonzero entries and their positions in the rows. To do this, we store the nonzero entries for a row packed, then follow this with a list of positions for each row of the matrix. The positions are integers, so we use a long integer for each of these. Thus our row-sparse matrix can be stored in $r(l*h+\lceil\frac{h}{n}\rceil)$ bytes, where l is the number of bytes used to store a long integer. We need to be careful here since l depends on the architecture of the machine being used. For our purposes, we assume $l = 4$, which is standard for 32-bit architecture.

Comparing these two formulas, we see that the row-sparse matrix will fit into fewer bytes as long as $h < \frac{c}{4n+1}$. In practice h is generally very small compared to c. Consider the Mathieu group M_{11} in characteristic 2. In this case, $q = 2$ so $n = 8$, $r = c = 112$ and $h = 9$. Checking our formulas, we get that for $g \in M_{11}$, the matrix for $e_H g e_H$ will require $112\lceil\frac{112}{8}\rceil = 112 * 14$ bytes and the row-sparse matrix will fit into $112(4 * 9 + \lceil\frac{9}{8}\rceil) = 112 * 38$. In this case, the full matrices actually take up less space than the sparse format matrices. This seems like it might be a problem, but considering a larger example will demonstrate the pay off of using sparse matrices. Consider the McLaughlin group McL in characteristic 3. For this group, $q = 9$ so $n = 2$, $r = c = 89941$ and $h = 100$. For $g \in McL$, the matrix for $e_H g e_H$ will require $89941\lceil\frac{89941}{2}\rceil = 89941 * 44971$ whereas the row-sparse version fits in only $89941(4 * 100 + \lceil\frac{100}{2}\rceil) = 89941 * 450$. The row-sparse matrix is almost 100 times smaller than the full matrix.

4.3.2 Applications

We have seen that storing row-sparse matrices in sparse format saves a lot of storage space. This format would be more useful if we did not have to write the matrices in full matrix format in order to apply our programs. Fortunately, the Spinning Algorithm (2.6.1), which we need for finding generators of $e_H \mathbb{F} G e_H$, depends mainly on vector by matrix multiplication. The author has adjusted the implementations of `zsp` and `zmu` to work for row-sparse matrices. Before explaining this further, we consider how the *C-MeatAxe* implements vector by matrix multiplication.

Let v be a vector and M a matrix. To calculate vM, we step through the entries of v. For each nonzero entry of v, we multiply the corresponding row of M by the entry and add it to a result vector. Not only is this a simple algorithm, but it is quickly modified to work for vector by row-sparse matrix. The only change required is that after we multiply entries of a row by a nonzero entry of v, we add these entries

to the corresponding positions in the result vector.

Looking back at the Spinning Algorithm on page 69, we see that having multiplication work for row-sparse matrices, the Spinning Algorithm automatically works for row-sparse matrices. This is important since the implementation of the Spinning Algorithm in the *C-MeatAxe*, `zsp`, requires that all of the algebra generators fit into memory. With the reduction in size the row-sparse format has over full size matrices, we can fit larger algebra generators into memory.

This has allowed us to find generating sets for $e_H\mathbb{F}Ge_H$ for large groups G. Let us go back to the example of the McLaughlin group in characteristic 3. By our calculations, a single element e_Hge_H would require $89941 * 44971$ bytes, or 3.77 gigabytes to store in its full matrix form. Generators of this size are, for all practical purposes, impossible to work with. However, in row-sparse format, an element would require only $89941 * 450$ bytes, or approximately 38.5 megabytes, so several of these elements could fit in memory on most contemporary desktop computers.

We have focused on storage and memory issues related to row-sparse matrices. Since implementations have been modified to accommodate sparse matrix format, we should also be interested in how the timings of the new programs compare to the old. Considering the implementation of vector by matrix multiplication discussed above, we assume that using a matrix in sparse format would be faster since we reduce the number of additions. We don't gain as much as one might expect since adding two full bytes together can be done without unpacking the individual entries in each byte. However, multiplying a vector by a 9-sparse square matrix with 5504 rows is about 3 times faster than using the full matrix format. So this sparse matrix format improves the speed of our calculations as well as solving our space issues.

Chapter 5

Results

In this chapter, we construct basic algebras for the principal block of specific groups over various characteristics. Since we are only concerned with principal blocks, for the rest of this chapter, all results are only for the principal block of $\mathbb{F}G$.

The first section will contain a brief summary of which algebras were previously known and which results are new. For many of these algebras, the set of relations we have is hundreds or thousands of words. Displaying these here would be useless for understanding so we do not include presentations for the basic algebras here. The basic algebras described in this dissertation are available at
`http://math.arizona.edu/~hoffmant/basicalg.html`.

5.1 Data Summary

There are not many sources where basic algebras have been recorded. The most notable is [Erd90] on pages 294 to 306. In [Lux97], while there are no explicit basic algebras, pages 71 through 107 contain Loewy series for several sporadic groups, which is enough information to construct Ext-quivers for several group algebras. Similarly, [Ben83a] and [Ben83b] contain Loewy series for A_8 and A_9 in characteristic 2, respectively. In [Sin96], the extensions of simple modules for the Janko group J_1 in characteristic 2 are calculated. We also mention that when the prime divides the order of G only once, the basic algebra can be read off the Brauer tree. Brauer trees for sporadic groups have been cataloged in [HL89].

The following tables contain the references for previously known algebras, as well as the page numbers for the algebras in this dissertation. These tables are complete

so far as the author knows. The groups are collected into four tables corresponding to their type as either alternating, symmetric, Mathieu, or other sporadic.

5.1.1 Alternating Groups

The results for alternating groups contained in this dissertation begin with A_5, since it is notable as the smallest simple group. Basic algebras have been computed by the author for the alternating groups up through A_{12} with the exception of A_{12} in characteristic 2.

Note that while Martin gives an Ext-quiver for A_{10} in characteristic 2 in [MR96], it does not agree with the Ext-quiver constructed by the author on page 121. Klaus Lux has verified the author's Ext-quiver using the software described in Chapter 5 of [Lux97].

5.1.2 Symmetric Groups

The results for symmetric groups contained in this dissertation begin with S_5, to correspond with the beginning of the alternating groups. Basic algebras have been computed by the author for the symmetric groups up through S_{11}. Stuart Martin, along with Karin Erdmann and Lee Russell, have studied Ext-quivers of group algebras of symmetric groups. Their approach covers many of the larger symmetric groups, but some of their results do overlap with those of the author.

5.1.3 Sporadic Groups

The author has constructed basic algebras for all the sporadic groups with order less than that of the Held group with a few exceptions noted in Tables 5.3 and 5.4.

We split the sporadic groups into two tables just for layout purposes. First the Mathieu groups, then the rest of the sporadic groups addressed in this dissertation.

Group	Prime	Reference	Page
A_5	2	[Erd90] page 295	111
	3		112
	5		112
A_6	2		113
	3		113
	5		113
A_7	2	[Erd90] page 295	114
	3		115
	5		115
	7		116
A_8	2	[Ben83a]	116
	3		117
	5		118
	7		118
A_9	2	[Ben83b]	119
	3		119
	5		120
	7		120
A_{10}	2	[MR96]*	121
	3		121
	5		122
	7		123
A_{11}	2		124
	3		124
	5		125
	7		125
	11		126
A_{12}	3		127
	5		127
	7		128
	11		129

TABLE 5.1. Alternating Groups

5.2 Data Description

For each of the groups in the following sections, we have included the data necessary to reproduce these results using Algorithm 3.4.1.

The first step is identifying the condensation subgroup. In most cases, we identify a subgroup N in the Table of Marks stored in *GAP*, see [GAP02]. Our condensation

Group	Prime	Reference	Page
S_5	2		129
	3		130
	5		130
S_6	2		131
	3	[Mar90]	131
	5		131
S_7	2		132
	3	[Mar90]	133
	5		133
	7		134
S_8	2		135
	3	[Mar90]	135
	5		136
	7		136
S_9	2		137
	3		137
	5		138
	7		139
S_{10}	2		139
	3		140
	5	[Mar89]	140
	7		141
S_{11}	2		142
	3		143
	5	[Mar90]	143
	7		144
	11		144

TABLE 5.2. Symmetric Groups

subgroup H is a normal subgroup of N. The main reason for constructing H in N is to use Theorem 2.16. The normalizing elements in the theorem are the generators of a Sylow-p subgroup of N.

Once we have the condensation subgroup, we need to find a generating set for $e_H\mathbb{F}Ge_H$. The idea is to construct elements e_Hge_H for "random" $g \in G$. Since we want our work to be reproducible, we use a script to construct the standard words in G described in Table 5.5 on page 111 from two elements a, b of G.

We find the generators of $e_H\mathbb{F}Ge_H$ as described in Section 4.2 and record them

Group	Prime	Reference	Page
M_{11}	2	[Erd90] page 301	146
	3		146
	5	[HL89] page 64	146
	11	[HL89] page 65	147
M_{12}	2		148
	3		148
	5	[HL89] page 66	149
	11	[HL89] page 67	149
M_{22}	2		152
	3		153
	5	[HL89] page 82	154
	7	[HL89] page 87	155
	11	[HL89] page 92	155
M_{23}	2		158
	3		159
	5	[HL89] page 104	159
	7	[HL89] page 105	160
	11	[HL89] page 106	160
	23	[HL89] page 107	161
M_{24}	2		168†
	3	[Lux97] page 77	170
	5	[HL89] page 126	170
	7	[HL89] page 128	174
	11	[HL89] page 130	175
	23	[HL89] page 131	176

TABLE 5.3. Mathieu Groups
† The basic algebra for M_{24} in characteristic 2 has not yet been constructed.

in terms of the standards words in the generators of G. When evaluating the standard words on the generators g_1, g_2 of G, we write zi instead of $zi(g_1, g_2)$, but we include the input when it differs from g_1, g_2, i.e., we write $z3(z4, z8)$ for the element $z3(z4(g_1, g_2), z8(g_1, g_2))$.

We use Theorem 2.16 when it is applicable. So for many of the following groups we also list the normalizing elements described by the theorem. These elements are recorded as words in the generators of the subgroup N. Note that these elements are listed here as words, not straight line programs. We list words here for documentation purposes only. The program `FindCondSubgroup` returns the straight line programs

Group	Prime	Reference	Page
J_1	2	[Sin96]	150
	3	[HL89] page 69	150
	5	[HL89] page 71	151
	7	[HL89] page 72	151
	11	[HL89] page 74	152
	19	[HL89] page 76	152
J_2	2	[Lux97] page 82	156
	3	[Lux97] page 92	156
	5	[Lux97] page 95	157
	7	[HL89] page 102	158
HS	2		162
	3		162
	5	[Lux97] page 71	163
	7	[HL89] page 110	164
	11	[HL89] page 112	164
J_3	2		165
	3	[Lux97] page 97	165
	5	[HL89] page 115	166
	17	[HL89] page 118	167
	19	[HL89] page 121	167
McL	2		177
	5		177
	7	[HL89] page 134	178
	11	[HL89] page 137	178
He	7		179

TABLE 5.4. Other Sporadic Groups

for computational purposes.

We have included the dimensions of both the condensation subgroup and the basic algebra. This information is useful as a check that the results are correct. It also points out the advantage the basic algebra has over the condensation subalgebra.

The basic algebras have not been included here. For most of these groups, the description of the basic algebra would be long and not useful to read. Instead, we include the Ext-quiver. While the Ext-quiver is not a complete description of the basic algebra, it does contain some information about the basic algebra. The first two Loewy layers of the projective indecomposable $\mathbb{F}G$-modules can also be read from

$z1(a,b)$	a,
$z2(a,b)$	b,
$z3(a,b)$	$z1(a,b) * z2(a,b)$,
$w(a,b)$	$z3(a,b) * z2(a,b)$,
$z4(a,b)$	$z3(a,b) * w(a,b)$,
$z5(a,b)$	$z3(a,b) * z4(a,b)$,
$z6(a,b)$	$z3(a,b) * z5(a,b)$,
$z7(a,b)$	$z4(a,b) * z5(a,b)$,
$z8(a,b)$	$z3(a,b) * z6(a,b)$,
$z9(a,b)$	$z7(a,b) * w(a,b)$,
$v(a,b)$	$z4(a,b) * w(a,b)$,
$z10(a,b)$	$v(a,b) * z5(a,b)$,
$z11(a,b)$	$v(a,b)$

TABLE 5.5. Standard Words in a Group

the Ext-quiver.

If condensation is used, then the Ext-quiver recorded here has vertices with two labels. The first label corresponds to the name of a simple $e_H\mathbb{F}Ge_H$-module and the name in parentheses is the corresponding simple $\mathbb{F}G$-module.

5.3 Alternating Groups

5.3.1 A_5

The order of A_5 is $2^2 \cdot 3 \cdot 5 = 60$.

Characteristic 2: The only possible condensation subgroup for A_5 in characteristic 2 is the trivial subgroup. So we skip condensation and work with the group algebra. We use the standard generators given by the Table of Marks in *GAP*. Since there is no generation issue, we also have no need for normalizing elements. The basic algebra has dimension 18 and its Ext-quiver is given in Figure 5.1.

$$2a \longleftarrow 1a \longrightarrow 2b$$

FIGURE 5.1. Ext-quiver of A_5 in characteristic 2

Characteristic 3: We take the subgroup N with number 8 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{aba^2, a^2ba\}.$$

The condensation subalgebra has dimension 6 and is generated by

$$z2$$

and the normalizing element

$$a.$$

It turns out that the condensation subalgebra is the basic algebra and its Ext-quiver is given in Figure 5.2.

$$1a(1a) \longleftarrow 1b(4a)$$

FIGURE 5.2. Ext-quiver of A_5 in characteristic 3

Characteristic 5: We take the subgroup N with number 6 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{aba\}.$$

The condensation subalgebra has dimension 8 and is generated by

$$z9, z1.$$

The basic algebra has dimension 7 and its Ext-quiver is given in Figure 5.3.

FIGURE 5.3. Ext-quiver of A_5 in characteristic 5

5.3.2 A_6

The order of A_6 is $2^3 \cdot 3^2 \cdot 5 = 360$.

Characteristic 2: The only possible condensation subgroup for A_6 in characteristic 2 is the trivial subgroup. So we skip condensation and work with the group algebra. We use the standard generators given by the Table of Marks in *GAP*. Since there is no generation issue, we also have no need for normalizing elements. The basic algebra has dimension 34 and its Ext-quiver is given in Figure 5.4.

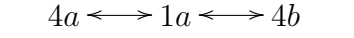


FIGURE 5.4. Ext-quiver of A_6 in characteristic 2

Characteristic 3: We take the subgroup N with number 11 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abab\}.$$

The condensation subalgebra has dimension 92 and is generated by

$$z7, z4, z3.$$

The basic algebra has dimension 7 and its Ext-quiver is given in Figure 5.5.

Characteristic 5: Let H be the subgroup with number 11 in the Table of Marks. The condensation subalgebra $e_H \mathbb{F}G e_H$ is generated by the words

$$z2, z4, z7$$

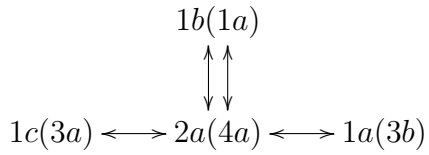


FIGURE 5.5. Ext-quiver of A_6 in characteristic 3

and has dimension 9. The basic algebra has dimension 7 and its Ext-quiver is given in Figure 5.6.

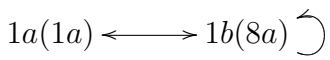


FIGURE 5.6. Ext-quiver of A_6 in characteristic 5

5.3.3 A_7

The order of A_7 is $2^3 \cdot 3^2 \cdot 5 \cdot 7 = 2520$.

Characteristic 2: We take the subgroup N with number 25 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a\}.$$

The condensation subalgebra has dimension 54 and is generated by

$$z3, z1, z2.$$

The basic algebra has dimension 19 and its Ext-quiver is given in Figure 5.7.

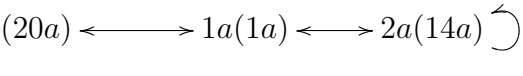


FIGURE 5.7. Ext-quiver of A_7 in characteristic 2

Characteristic 3: We take the subgroup N with number 25 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a\}.$$

The condensation subalgebra has dimension 54 and is generated by

$$z1, z2$$

and the element corresponding to

$$b.$$

The basic algebra has dimension 36 and its Ext-quiver is given in Figure 5.8.

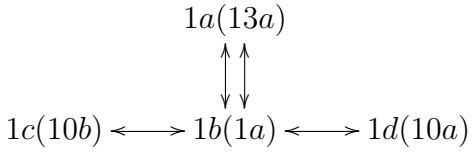


FIGURE 5.8. Ext-quiver of A_7 in characteristic 3

Characteristic 5: We take the subgroup N with number 27 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^3c, a^3, a^2\}.$$

The condensation subalgebra has dimension 27 and is generated by

$$z5, z4, z6.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.9.

$$1a(1a) \longleftrightarrow 1c(13a) \longleftrightarrow 1b(8a) \longleftrightarrow 2b(6a)$$

FIGURE 5.9. Ext-quiver of A_7 in characteristic 5

Characteristic 7: We take the subgroup N with number 35 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ca, b^2, ababab\}.$$

The condensation subalgebra has dimension 35 and is generated by

$$z3, z4, z5.$$

The basic algebra has dimension 11 and its Ext-quiver is given in Figure 5.10.

$$1a(1a) \longleftrightarrow 2b(5a) \longleftrightarrow 1c(10a) \circlearrowleft$$

FIGURE 5.10. Ext-quiver of A_7 in characteristic 7

5.3.4 A_8

The order of A_8 is $2^6 \cdot 3^2 \cdot 5 \cdot 7 = 20160$.

Characteristic 2: We take the subgroup N with number 90 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{baba\}.$$

The condensation subalgebra $e_H \mathbb{F}Ge_H$ is generated by the words

$$z9, z5, z1$$

and the elements corresponding to the words

$$bab^3, b^2a,$$

and has dimension 2248. The basic algebra has dimension 226 and its Ext-quiver is given in Figure 5.11.

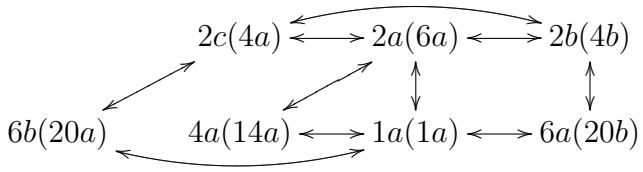


FIGURE 5.11. Ext-quiver of A_8 in characteristic 2

Characteristic 3: We take the subgroup N with number 85 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababc, bcbc, ba, abab\}.$$

The condensation subalgebra has dimension 93 and is generated by

$$z11, z2, z3, z8(z1, z3).$$

The basic algebra has dimension 46 and its Ext-quiver is given in Figure 5.12.

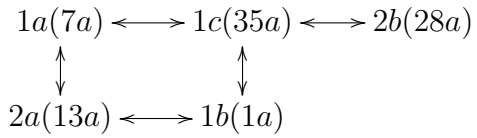


FIGURE 5.12. Ext-quiver of A_8 in characteristic 3

Characteristic 5: We take the subgroup N with number 106 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{acac, a, caba, cabcab\}.$$

The condensation subalgebra has dimension 30 and is generated by

$$z1, z4, z11.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.13.

$$1c(1a) \longleftrightarrow 1e(13a) \longleftrightarrow 1d(43a) \longleftrightarrow 1f(21a)$$

FIGURE 5.13. Ext-quiver of A_8 in characteristic 5

Characteristic 7 We take the subgroup N with number 114 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2baba^2b, a^3ba^3b, a^2, a^4ba^4b, aba^2ba\}.$$

The condensation subalgebra has dimension 17 and is generated by

$$z2, z5, z5(z1, z3).$$

The basic algebra has dimension 11 and its Ext-quiver is given in Figure 5.14.

$$1f(1a) \longleftrightarrow 1c(19a) \longleftrightarrow 1d(45a) \circlearrowleft$$

FIGURE 5.14. Ext-quiver of A_8 in characteristic 7

5.3.5 A_9

The order of A_9 is $2^6 \cdot 3^4 \cdot 5 \cdot 7 = 181440$.

Characteristic 2: We take the subgroup N with number 183 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{bab^2ab, b^2ab^2a\}.$$

The condensation subalgebra has dimension 2268 and is generated by

$$z6, z11, z5$$

and the element corresponding to the word

$$\{bab^2ab^2ca\}.$$

The basic algebra has dimension 296 and its Ext-quiver is given in Figure 5.15.

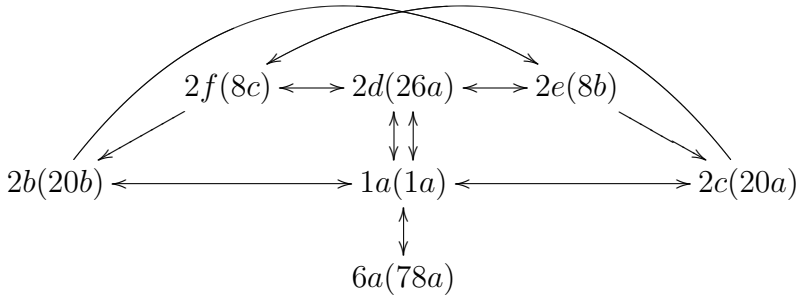


FIGURE 5.15. Ext-quiver of A_9 in characteristic 2

Characteristic 3: We take the subgroup N with number 163 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^3bab, a^3bab^3, a^2\}.$$

The condensation subalgebra has dimension 512 and is generated by

$$z5, z2, z5(z1, z3), z8(z1, z9).$$

The basic algebra has dimension 166 and its Ext-quiver is given in Figure 5.16.

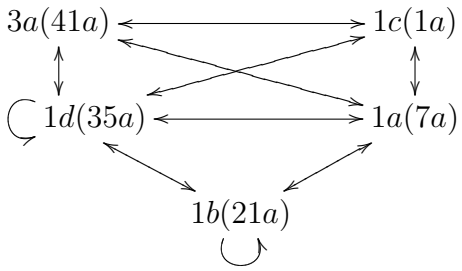


FIGURE 5.16. Ext-quiver of A_9 in characteristic 3

Characteristic 5: We take the subgroup N with number 183 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^2ab, b^2abab^2c, babca, abab^2abca\}.$$

The condensation subalgebra has dimension 44 and is generated by

$$z3, z7, z2, z7(z1, z3).$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.17.

$$2a(56a) \longleftrightarrow 1a(133a) \longleftrightarrow 1b(83a) \longleftrightarrow 1h(1a)$$

FIGURE 5.17. Ext-quiver of A_9 in characteristic 5

Characteristic 7 We take the subgroup N with number 162 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a, b\}.$$

The condensation subalgebra has dimension 63 and is generated by

$$z8, z10.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.18.

$$2e(8a) \longleftrightarrow 1b(19a) \longleftrightarrow 1c(101a) \longleftrightarrow 1a(115a) \longleftrightarrow 2a(47a) \longleftrightarrow 1e(1a)$$

FIGURE 5.18. Ext-quiver of A_9 in characteristic 7

5.3.6 A_{10}

The order of A_10 is $2^7 \cdot 3^4 \cdot 5^2 \cdot 7 = 1814400$.

Characteristic 2: We take the subgroup N with number 336 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{baba, abab^2ab^2ab\}.$$

The condensation subalgebra has dimension 22500 and is generated by

$$z11, z9,$$

and the element corresponding to the word

$$abab^2a.$$

The basic algebra has dimension 646 and its Ext-quiver is given in Figure 5.19.

Characteristic 3: We take the subgroup N with number 322 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ba^3cb, abacbc, aba^2, aba^3b, a^2b^2, a^2\}.$$

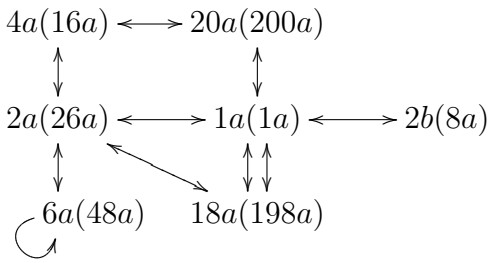


FIGURE 5.19. Ext-quiver of A_{10} in characteristic 2

The condensation subalgebra has dimension 526 and is generated by

$$z3, z9, z6.$$

The basic algebra has dimension 166 and its Ext-quiver is given in Figure 5.20.

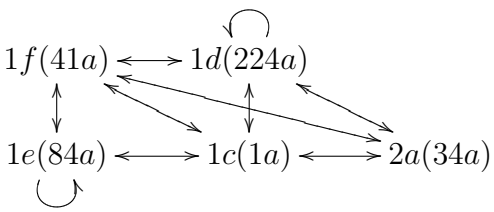


FIGURE 5.20. Ext-quiver of A_{10} in characteristic 3

Characteristic 5: We take the subgroup N with number 329 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{babcb, abab^3cbcb, abab, b^6\}.$$

The condensation subalgebra has dimension 1512 and is generated by

$$z5, z7, z6.$$

The basic algebra has dimension 121 and its Ext-quiver is given in Figure 5.21.

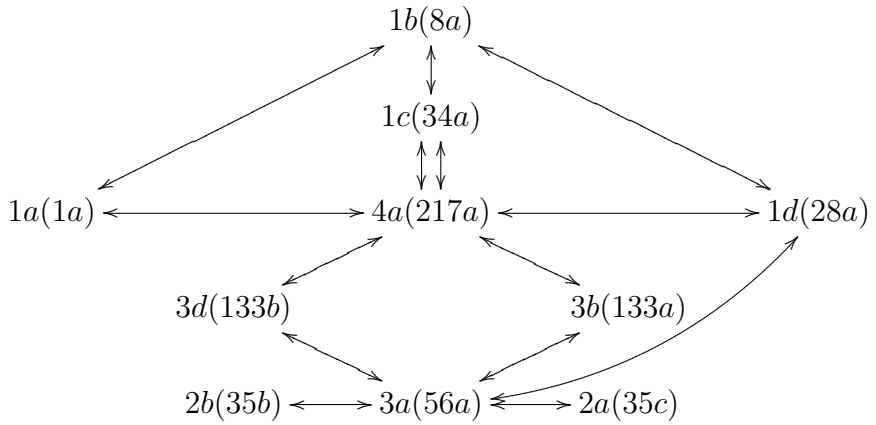


FIGURE 5.21. Ext-quiver of A_{10} in characteristic 5

Characteristic 7 We take the subgroup N with number 392 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababa^2ba^2, a^2ba^2baba^2ba^2, ababa^2bababab, a^2baba^2baba^2bab,$$
$$a^2bababa^2b, ababa^2baba^2baba, a^2ba^2babab\}.$$

The condensation subalgebra has dimension 55 and is generated by

$$z3, z1, z2.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.22.

$$1a(1a) \longleftrightarrow 1b(89a) \longleftrightarrow 1d(199a) \longleftrightarrow 1f(101a) \longleftrightarrow 1g(124a) \longleftrightarrow 1e(36a)$$

FIGURE 5.22. Ext-quiver of A_{10} in characteristic 7

5.3.7 A_{11}

The order of A_{11} is $2^7 \cdot 3^4 \cdot 5^2 \cdot 7 \cdot 11 = 19958400$.

Characteristic 2: We take the subgroup N with number 516 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^5b^2ab, abab, a^4b^3a^2\}.$$

The condensation subalgebra has dimension 5176 and is generated by

$$z6, z4, z3,$$

and the element corresponding to the word

$$ab^3a^2.$$

The basic algebra has dimension 562 and its Ext-quiver is given in Figure 5.23.

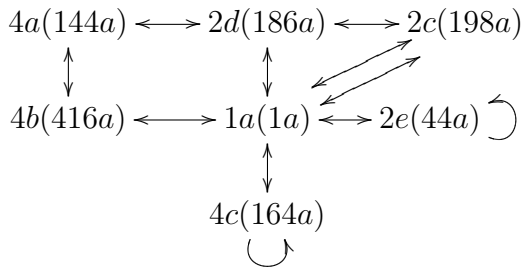


FIGURE 5.23. Ext-quiver of A_{11} in characteristic 2

Characteristic 3: We take the subgroup N with number 551 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2ba^2bca, a^3b, a^3cac, baba, aba^2ba\}.$$

The condensation subalgebra has dimension 3264 and is generated by

$$z7, z6, z9, z5.$$

The basic algebra has dimension 372 and its Ext-quiver is given in Figure 5.24.

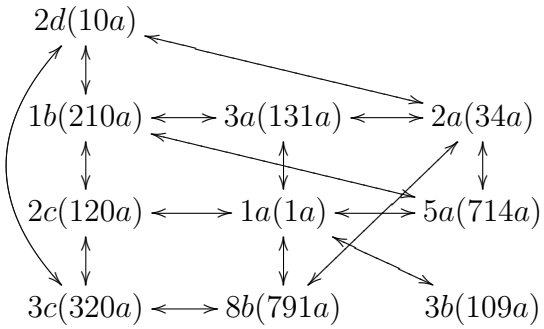


FIGURE 5.24. Ext-quiver of A_{11} in characteristic 3

Characteristic 5: Let H be the subgroup with number 562 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z10, z2, z4, z1,$$

and has dimension 736 over F. The basic algebra has dimension 121 and its Ext-quiver is given in Figure 5.25.

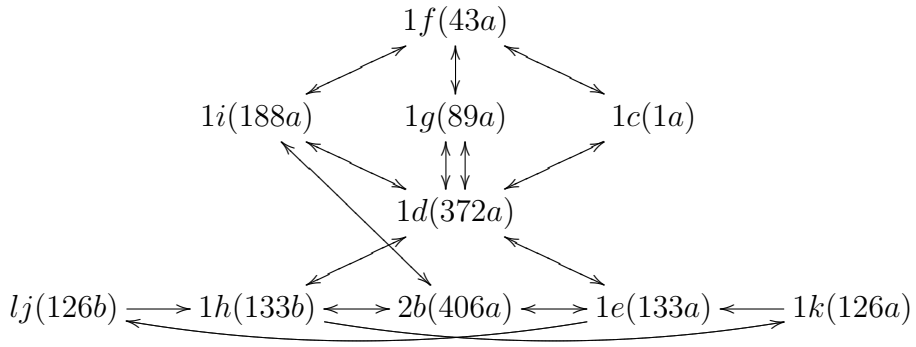


FIGURE 5.25. Ext-quiver of A_{11} in characteristic 5

Characteristic 7: We take the subgroup N with number 706 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^7bab, a^3baba^2b, a^4, c\}.$$

The condensation subalgebra has dimension 244 and is generated by

$$z7, z6, z9, z5.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.26.

$$1b(1a) \longleftrightarrow 1c(131a) \longleftrightarrow 1i(199a) \longleftrightarrow 2d(626a) \longleftrightarrow 1f(474a) \longleftrightarrow 1j(120a)$$

FIGURE 5.26. Ext-quiver of A_{11} in characteristic 7

Characteristic 11: We take the subgroup N with number 761 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^4, a^2baba^2b^2a^3, ba^3ba^2b^2a\}.$$

The condensation subalgebra has dimension 199 and is generated by

$$z7, z10, z9, z3.$$

The basic algebra has dimension 19 and its Ext-quiver is given in Figure 5.27.

$$1b(1a) \longleftrightarrow 2a(9a) \longleftrightarrow 2b(36a) \longleftrightarrow 2d(84a) \longleftrightarrow 1d(126a) \circlearrowleft$$

FIGURE 5.27. Ext-quiver of A_{11} in characteristic 11

5.3.8 A_{12}

The order of A_{12} is $2^9 \cdot 3^5 \cdot 5^2 \cdot 7 \cdot 11 = 239500800$.

Characteristic 3: We take the subgroup N with number 2003 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^2abab^3, aba^2bab^2, a^2, a^2ba^2b^3\}.$$

The condensation subalgebra has dimension 24472 and is generated by

$$z9, z2, z6, z7, z3(z1, z3).$$

The basic algebra has dimension 1781 and its Ext-quiver is given in Figure 5.28.

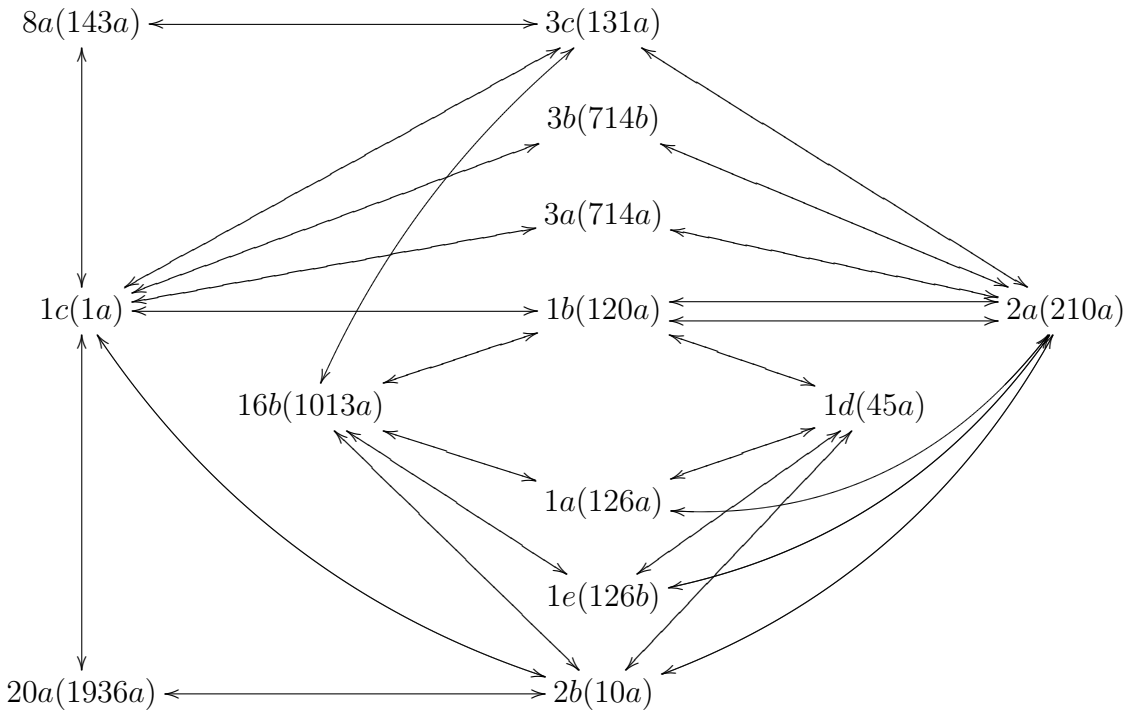


FIGURE 5.28. Ext-quiver of A_{12} in characteristic 3

Characteristic 5: We take the subgroup N with number 2195 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal

subgroup of N generated by

$$\{a^3b^2ab^2c, b, abab^3a, baba, a^3b^2ab^2, a^2b^2, a^2ba^2b^3, a^2\}.$$

The condensation subalgebra has dimension 1953 and is generated by

$$z3, z5, z1, z9, z4, z4(z1, z3).$$

The basic algebra has dimension 178 and its Ext-quiver is given in Figure 5.29.

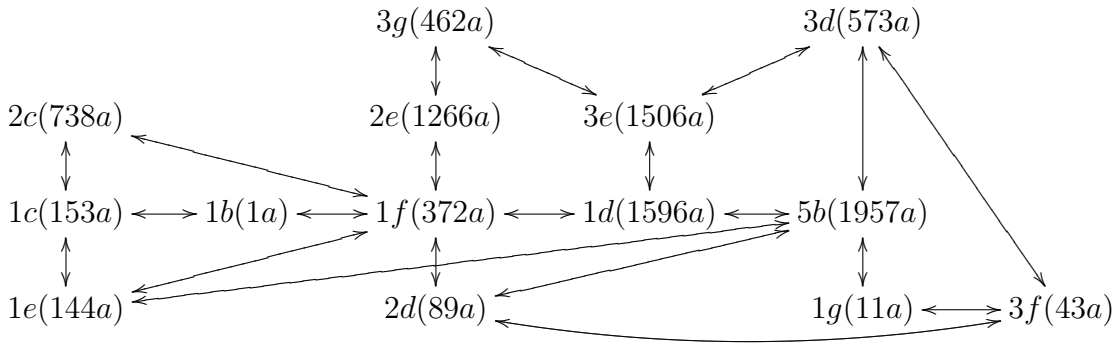


FIGURE 5.29. Ext-quiver of A_{12} in characteristic 5

Characteristic 7: Let H be the subgroup with number 2356 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words $z2$, $z3$ and $z11$, and has dimension 112 over F. The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.30.

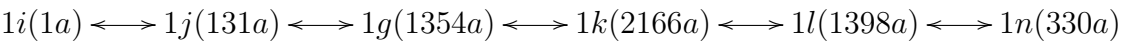


FIGURE 5.30. Ext-quiver of A_{12} in characteristic 7

Characteristic 11: We take the subgroup N with number 2482 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^4ba^2b^2a^2b, ba^2b^2a^2b, a^5ba^2b^3a, a^5ba^3b^2, ba^2baca, a^5ba^5b^2c^2, a^5ba^4c^2ad\}.$$

The condensation subalgebra has dimension 62 and is generated by

$$z2, z5(z1, z3), z5(z1, z4), z10(z1, z5).$$

The basic algebra has dimension 19 and its Ext-quiver is given in Figure 5.31.

$$1b(1a) \longleftrightarrow 1c(53a) \longleftrightarrow 1h(267a) \longleftrightarrow 1e(678a) \rightleftharpoons 1d(1050a) \circlearrowleft$$

FIGURE 5.31. Ext-quiver of A_{12} in characteristic 11

5.4 Symmetric Groups

5.4.1 S_5

The order of S_5 is $2^3 \cdot 3 \cdot 5 = 120$.

Characteristic 2: The only possible condensation subgroup for S_5 in characteristic 2 is the trivial subgroup. So we skip condensation and work with the group algebra. We use the standard generators given by the Table of Marks in *GAP*. Since there is no generation issue, we also have no need for normalizing elements. The basic algebra has dimension 19 and its Ext-quiver is given in Figure 5.32.

$$4b \longleftrightarrow 1a \circlearrowleft$$

FIGURE 5.32. Ext-quiver of S_5 in characteristic 2

Characteristic 3: We take the subgroup N with number 12 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab\}.$$

The condensation subalgebra has dimension 9 and is generated by

$$z1, z4.$$

The basic algebra has dimension 6 and its Ext-quiver is given in Figure 5.33.

$$1a(1a) \longleftrightarrow 1d(4b)$$

FIGURE 5.33. Ext-quiver of S_5 in characteristic 3

Characteristic 5: We take the subgroup N with number 15 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b\}.$$

The condensation subalgebra has dimension 16 and is generated by

$$z4, z2.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.34.

$$1a(1b) \longleftrightarrow 1b(3b) \longleftrightarrow 1d(3a) \longleftrightarrow 1c(1a)$$

FIGURE 5.34. Ext-quiver of S_5 in characteristic 5

5.4.2 S_6

The order of S_6 is $2^4 \cdot 3^2 \cdot 5 = 720$.

Characteristic 2: The only possible condensation subgroup for S_6 in characteristic 2 is the trivial subgroup. So we skip condensation and work with the group algebra. We use the standard generators given by the Table of Marks in *GAP*. Since there is no generation issue, we also have no need for normalizing elements. The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.35.

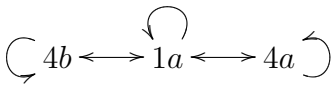


FIGURE 5.35. Ext-quiver of S_6 in characteristic 2

Characteristic 3: We take the subgroup N with number 34 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2\}.$$

The condensation subalgebra has dimension 184 and is generated by

$$z5, z6.$$

The basic algebra has dimension 51 and its Ext-quiver is given in Figure 5.36.

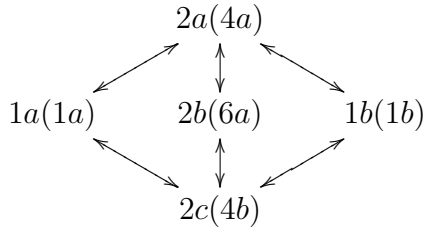


FIGURE 5.36. Ext-quiver of S_6 in characteristic 3

Characteristic 5: We take the subgroup N with number 34 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal

subgroup of N generated by

$$\{a^3c, ab, a^2\}.$$

The condensation subalgebra has dimension 18 and is generated by

$$z8, z4, z2.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.37.

$$1a(1b) \longleftrightarrow 1b(8b) \longleftrightarrow 1h(8a) \longleftrightarrow 1g(1a)$$

FIGURE 5.37. Ext-quiver of S_6 in characteristic 5

5.4.3 S_7

The order of S_7 is $2^4 \cdot 3^2 \cdot 5 \cdot 7 = 5040$.

Characteristic 2: We take the subgroup N with number 108 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abab^2\}.$$

The condensation subalgebra has dimension 108 and is generated by

$$z11, z3, z5$$

and the element corresponding to the word

$$abab^2a.$$

The basic algebra has dimension 38 and its Ext-quiver is given in Figure 5.38.

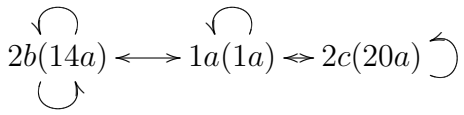


FIGURE 5.38. Ext-quiver of S_7 in characteristic 2

Characteristic 3: We take the subgroup N with number 76 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abab^2\}.$$

The condensation subalgebra has dimension 108 and is generated by

$$z3, z5$$

and the element corresponding to the word

$$b.$$

The basic algebra has dimension 51 and its Ext-quiver is given in Figure 5.39.

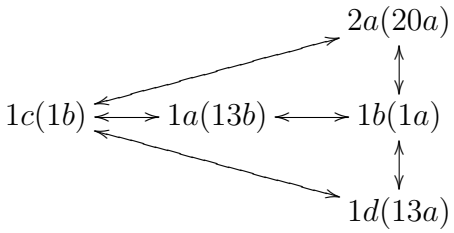


FIGURE 5.39. Ext-quiver of S_7 in characteristic 3

Characteristic 5: We take the subgroup N with number 79 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ac^2d, c^2, b^2\}.$$

The condensation subalgebra has dimension 54 and is generated by

$$z3, z6, z5.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.40.

$$1d(1a) \longleftrightarrow 1e(13a) \longleftrightarrow 1g(8b) \longleftrightarrow 2c(6b)$$

FIGURE 5.40. Ext-quiver of S_7 in characteristic 5

Characteristic 7: We take the subgroup N with number 90 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^2a, ab^2ab^2, a^2\}.$$

The condensation subalgebra has dimension 70 and is generated by

$$z5, z7, z4, z8.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.41.

$$1e(1a) \longleftrightarrow 2a(5a) \longleftrightarrow 1a(10a) \longleftrightarrow 1b(10b) \longleftrightarrow 2b(5b) \longleftrightarrow 1f(1b)$$

FIGURE 5.41. Ext-quiver of S_7 in characteristic 7

5.4.4 S_8

The order of S_8 is $2^7 \cdot 3^2 \cdot 5 \cdot 7 = 40320$.

Characteristic 2: We take the subgroup N with number 233 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^2ab^2\}.$$

The condensation subalgebra has dimension 4496 and is generated by

$$z8, z4,$$

and the elements corresponding to the words

$$d, b^2abc, a.$$

The basic algebra has dimension 289 and its Ext-quiver is given in Figure 5.42.

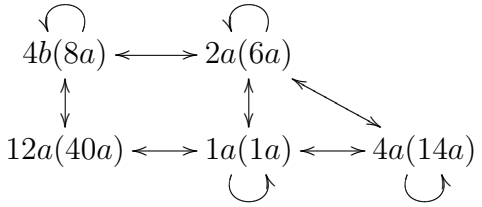


FIGURE 5.42. Ext-quiver of S_8 in characteristic 2

Characteristic 3: We take the subgroup N with number 225 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{bacd, abab, a, a^2\}.$$

The condensation subalgebra has dimension 186 and is generated by

$$z7, z3, z4, z7(z1, z4).$$

The basic algebra has dimension 46 and its Ext-quiver is given in Figure 5.43.

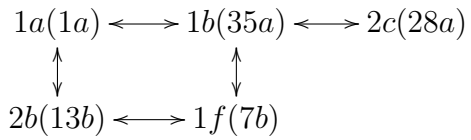


FIGURE 5.43. Ext-quiver of S_8 in characteristic 3

Characteristic 5: We take the subgroup N with number 261 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^3cbc, b, b^4cb^2c, b^2\}.$$

The condensation subalgebra has dimension 60 and is generated by

$$z2, z11, z8.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.44.

$$1f(1a) \longleftrightarrow 1j(13a) \longleftrightarrow 1i(43b) \longleftrightarrow 1g(21b)$$

FIGURE 5.44. Ext-quiver of S_8 in characteristic 5

Characteristic 7: We take the subgroup N with number 271 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^2abcad, abab, bababab, acac, ab^2ab\}.$$

The condensation subalgebra has dimension 34 and is generated by

$$z3, z7, z11, z4.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.45.

$$1g(1b) \longleftrightarrow 1h(19b) \longleftrightarrow 1m(45b) \longleftrightarrow 1l(45a) \longleftrightarrow 1j(19a) \longleftrightarrow 1i(1a)$$

FIGURE 5.45. Ext-quiver of S_8 in characteristic 7

5.4.5 S_9

The order of S_9 is $2^7 \cdot 3^4 \cdot 5 \cdot 7 = 362880$.

Characteristic 2: We take the subgroup N with number 354 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^8bab, a^3\}.$$

The condensation subalgebra has dimension 4492 and is generated by

$$z6, z3,$$

and the element corresponding to the word

$$a^4ba^2.$$

The basic algebra has dimension 370 and its Ext-quiver is given in Figure 5.46.

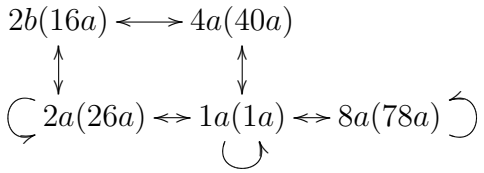


FIGURE 5.46. Ext-quiver of S_9 in characteristic 2

Characteristic 3: We take the subgroup N with number 456 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^3cac, baba, aba^2ba\}.$$

The condensation subalgebra has dimension 1024 and is generated by

$$z2, z10, z5, z8(z1, z3).$$

The basic algebra has dimension 332 and its Ext-quiver is given in Figure 5.47.

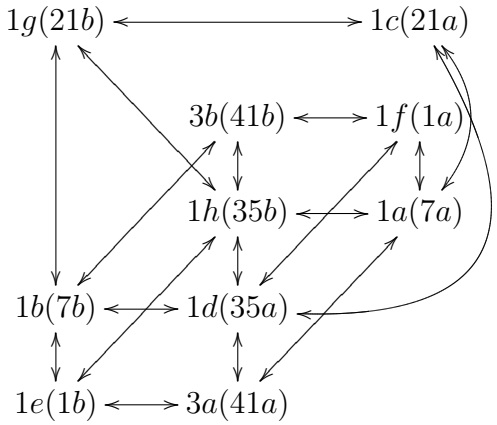


FIGURE 5.47. Ext-quiver of S_9 in characteristic 3

Characteristic 5: We take the subgroup N with number 476 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{d, aba, c^2a, ab^2ab, c^2\}.$$

The condensation subalgebra has dimension 54 and is generated by

$$z7, z4(z1, z3), z6(z1, z3), z11(z1, z6).$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.48.

$$1b(56a) \longleftrightarrow 1f(133b) \longleftrightarrow 1h(83a) \longleftrightarrow 1e(1a)$$

FIGURE 5.48. Ext-quiver of S_9 in characteristic 5

Characteristic 7: We take the subgroup N with number 454 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^2cd, b^2acac, a, aba, ab^2ab\}.$$

The condensation subalgebra has dimension 126 and is generated by

$$z9, z1.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.49.

$$2e(8b) \longleftrightarrow 1f(19b) \longleftrightarrow 1h(101b) \longleftrightarrow 1g(115a) \longleftrightarrow 2h(47a) \longleftrightarrow 1j(1a)$$

FIGURE 5.49. Ext-quiver of S_9 in characteristic 7

5.4.6 S_{10}

The order of S_{10} is $2^8 \cdot 3^4 \cdot 5^2 \cdot 7 = 3628800$.

Characteristic 2: We take the subgroup N with number 1389 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^2ab^2a, ab^2abab^2ab^3\}.$$

The condensation subalgebra has dimension 45000 and is generated by

$$z11, z4,$$

and the elements corresponding to the words

$$b^3ab^2ab^2, ab^5aba.$$

The basic algebra has dimension 1292 and its Ext-quiver is given in Figure 5.50.

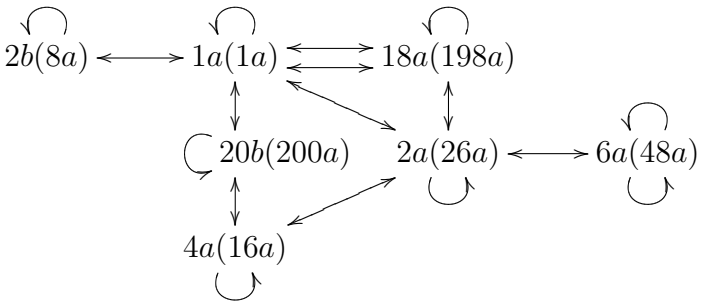


FIGURE 5.50. Ext-quiver of S_{10} in characteristic 2

Characteristic 3: We take the subgroup N with number 1351 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^2abad, b^3abac, bab^2ab^3, bab^2ab, abab^3, abab^2ab^3ab^2\}.$$

The condensation subalgebra has dimension 1052 and is generated by

$$z7, z8, z3, z2.$$

The basic algebra has dimension 332 and its Ext-quiver is given in Figure 5.51.

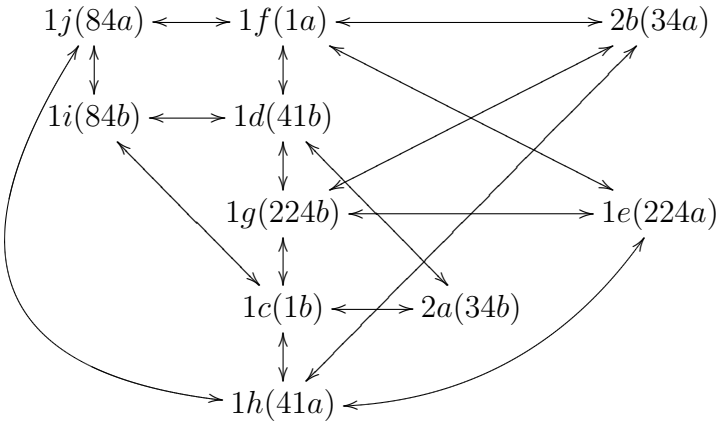


FIGURE 5.51. Ext-quiver of S_{10} in characteristic 3

Characteristic 5: We take the subgroup N with number 1379 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^5ba^4bc, a^3ba^3b, a^2ba^2b, a^2\}.$$

The condensation subalgebra has dimension 3008 and is generated by

$$z5, z4, z7.$$

The basic algebra has dimension 183 and its Ext-quiver is given in Figure 5.52.

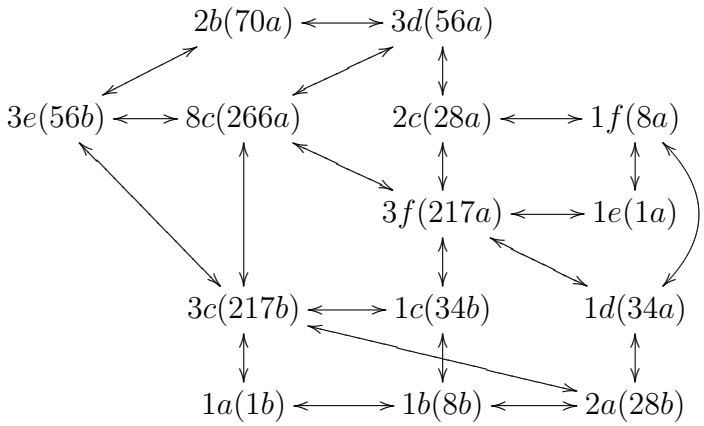


FIGURE 5.52. Ext-quiver of S_{10} in characteristic 5

Characteristic 7: We take the subgroup N with number 1510 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^5ab^2, ab^7ab, ab^3ab^5, aba^2bab^2, a^2bab^2ab, a^3ba^2b^3ab^4, a^2\}.$$

The condensation subalgebra has dimension 110 and is generated by

$$z7, z11, z1, z11(z1, z4).$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.53.

$$1q(1a) \longleftrightarrow 1s(89a) \longleftrightarrow 1r(199a) \longleftrightarrow 1k(101b) \longleftrightarrow 1e(124b) \longleftrightarrow 1j(36b)$$

FIGURE 5.53. Ext-quiver of S_{10} in characteristic 7

5.4.7 S_{11}

The order of S_{11} is $2^8 \cdot 3^4 \cdot 5^2 \cdot 7 \cdot 11 = 39916800$.

Characteristic 2: We take the subgroup N with number 2401 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^3ba^3b, a^2b^2, a^4ba^2b\}.$$

The condensation subalgebra has dimension 10352 and is generated by

$$z8, z4, z2,$$

and the elements corresponding to the words

$$ba^5bab, ab^2a^2.$$

The basic algebra has dimension 1124 and its Ext-quiver is given in Figure 5.54.

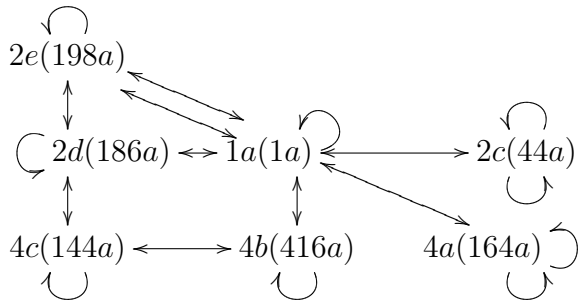


FIGURE 5.54. Ext-quiver of S_{11} in characteristic 2

Characteristic 3: We take the subgroup N with number 2509 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^3cd, a^{18}b, a^4b^2, a^{10}, a^4\}.$$

The condensation subalgebra has dimension 6756 and is generated by

$$z5, z8, z9, z11, z1, z5 + z1,$$

where the last word was added to improve the calculation of peakwords. The basic algebra has dimension 372 and its Ext-quiver is given in Figure 5.55.

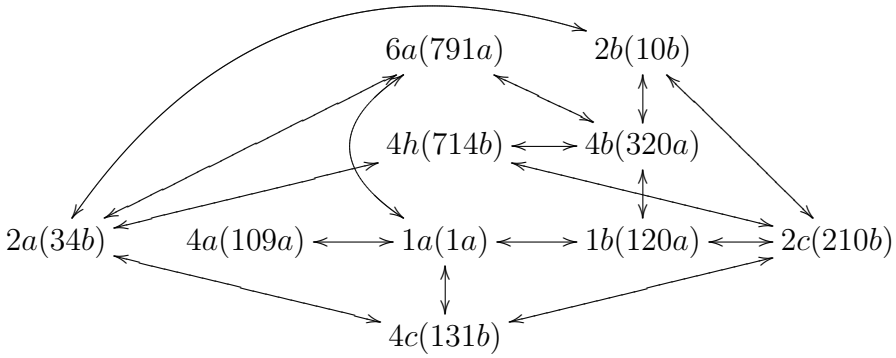


FIGURE 5.55. Ext-quiver of S_{11} in characteristic 3

Characteristic 5: We take the subgroup N with number 2526 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{c, a^2b^3ab, a^3, a^2, a^4ba^2b\}.$$

The condensation subalgebra has dimension 1472 and is generated by

$$z4, z9, z8, z1.$$

The basic algebra has dimension 178 and its Ext-quiver is given in Figure 5.56.

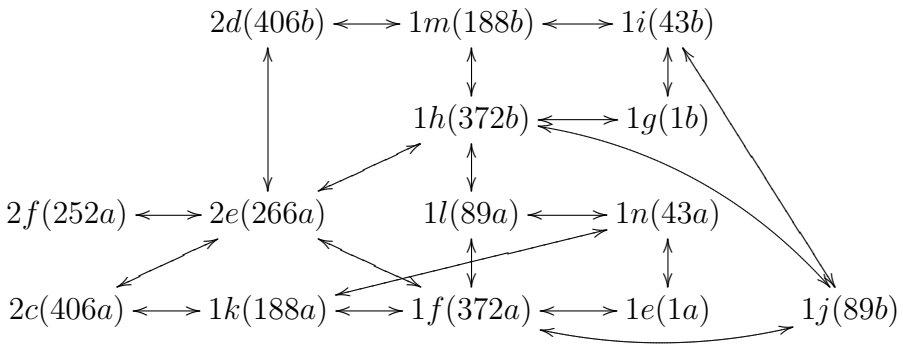


FIGURE 5.56. Ext-quiver of S_{11} in characteristic 5

Characteristic 7: We take the subgroup N with number 2910 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2ba^2bab, a^5ba^2b^2abacbc\}.$$

The condensation subalgebra has dimension 488 and is generated by

$$z6, z1, z2, z5(z1, z4).$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.57.

$$1b(120b) \longleftrightarrow 1c(474b) \longleftrightarrow 2i(626b) \longleftrightarrow 1n(199a) \longleftrightarrow 1u(131a) \longleftrightarrow 1t(1a)$$

FIGURE 5.57. Ext-quiver of S_{11} in characteristic 7

Characteristic 11: We take the subgroup N with number 3036 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^8, a^2ba^2b^2a^2b^3, a^6ba^2baba^2ba\}.$$

The condensation subalgebra has dimension 398 and is generated by

$$z1, z2, z5(z1, z3), z5(z1, z4).$$

The basic algebra has dimension 38 and its Ext-quiver is given in Figure 5.58.

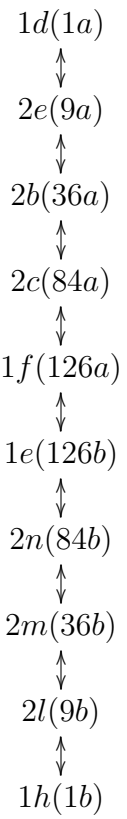


FIGURE 5.58. Ext-quiver of S_{11} in characteristic 11

5.5 Sporadic Simple Groups

5.5.1 M_{11}

The order of the Mathieu group M_{11} is $2^4 \cdot 3^2 \cdot 5 \cdot 11 = 7920$.

Characteristic 2: We take the subgroup N with number 35 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2b^2ab^2a, a^2\}.$$

The condensation subalgebra has dimension 112 and is generated by

$$z2, z6,$$

and the elements corresponding to the words

$$a^2ba^4, a^3b^2ab^2a, a^5bab, a^4b^2.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.59.

$$\circlearrowleft\ 4a(44a) \longleftrightarrow 1a(1a) \longleftrightarrow 2a(10a)\ \circlearrowright$$

FIGURE 5.59. Ext-quiver of M_{11} in characteristic 2

Characteristic 3: We take the subgroup N with number 21 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababa\}.$$

The condensation subalgebra has dimension 320 and is generated by

$$z3, z8, z1.$$

The basic algebra has dimension 83 and its Ext-quiver is given in Figure 5.60.

Characteristic 5: Let H be the subgroup with number 18 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words $z7$ and $z9$, and has dimension 37. The basic algebra has dimension 20 and its Ext-quiver is given in Figure 5.61.

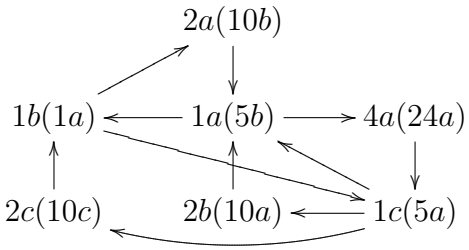


FIGURE 5.60. Ext-quiver of M_{11} in characteristic 3

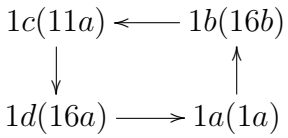


FIGURE 5.61. Ext-quiver of M_{11} in characteristic 5

Characteristic 11: We take the subgroup N with number 18 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ba, abab, b^2\}.$$

The condensation subalgebra has dimension 126 and is generated by

$$z2, z7.$$

The basic algebra has dimension 25 and its Ext-quiver is given in Figure 5.62.

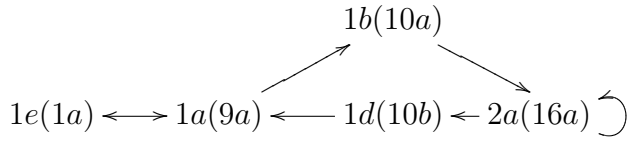


FIGURE 5.62. Ext-quiver of M_{11} in characteristic 11

5.5.2 M_{12}

The order of the Mathieu group M_{12} is $2^6 \cdot 3^3 \cdot 5 \cdot 11 = 95040$.

Characteristic 2: We take the subgroup N with number 133 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababab^2ababab^2, bababab^2ababab\}.$$

The condensation subalgebra has dimension 1216 and is generated by

$$z1$$

and the elements corresponding to the words

$$ababab^2ab^2a, baba, abababababab^2abab^2a, babababa.$$

The basic algebra has dimension 134 and its Ext-quiver is given in Figure 5.63.

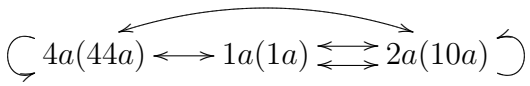


FIGURE 5.63. Ext-quiver of M_{12} in characteristic 2

Characteristic 3: We take the subgroup N with number 106 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2c, abab, a^2\}.$$

The condensation subalgebra has dimension 424 and is generated by

$$z2, z1, z7, z4.$$

The basic algebra has dimension 163 and its Ext-quiver is given in Figure 5.64.

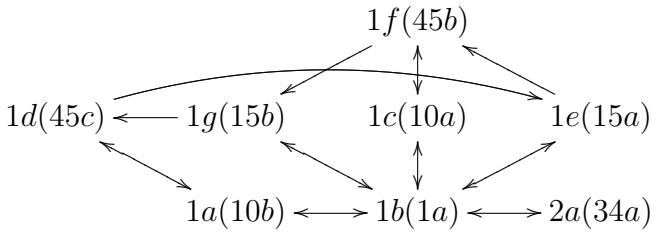


FIGURE 5.64. Ext-quiver of M_{12} in characteristic 3

Characteristic 5: We take the subgroup N with number 125 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab^3aba, ab^2ab^2a, b^2, bab^2ab, abab^2aba\}.$$

The condensation subalgebra has dimension 38 and is generated by

$$z5, z6.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.65.

$$1d(1a) \longleftrightarrow 1c(98a) \longleftrightarrow 1a(78a) \longleftrightarrow 1b(66a)$$

FIGURE 5.65. Ext-quiver of M_{12} in characteristic 5

Characteristic 11: Let H be the subgroup with number 113 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words $z11$, $z2$ and $z6$, and has dimension 29. The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.66.

$$1d(1a) \longleftrightarrow 1b(53a) \longleftrightarrow 1a(91a) \longleftrightarrow 1c(29a) \leftarrow 1f(16a) \circlearrowleft$$

FIGURE 5.66. Ext-quiver of M_{12} in characteristic 11

5.5.3 J_1

The order of the Janko group J_1 is $2^3 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 19 = 175560$.

Characteristic 2: We take the subgroup N with number 28 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{aba, abab^2\}.$$

The condensation subalgebra has dimension 412 and is generated by

$$z10, z2$$

and the element corresponding to the word

$$bab^2.$$

The basic algebra has dimension 82 and its Ext-quiver is given in Figure 5.67.

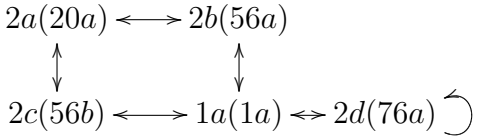


FIGURE 5.67. Ext-quiver of J_1 in characteristic 2

Characteristic 3: Let H be the subgroup with number 35 in the Table of Marks. The algebra $e_H \mathbb{F}G e_H$ is generated by the words

$$z10, z2,$$

and has dimension 19. The basic algebra has dimension 6 and its Ext-quiver is given in Figure 5.68.

$$1a(1a) \longleftrightarrow 1b(76a)$$

FIGURE 5.68. Ext-quiver of J_1 in characteristic 3

Characteristic 5: We take the subgroup N with number 36 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b, b^2abab^2aba\}.$$

The condensation subalgebra has dimension 68 and is generated by

$$z8, z6.$$

The basic algebra has dimension 10 and its Ext-quiver is given in Figure 5.69.

$$1b(1a) \longleftrightarrow 2b(76b)$$

FIGURE 5.69. Ext-quiver of J_1 in characteristic 5

Characteristic 7: We take the subgroup N with number 36 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b, b^2abab^2aba\}.$$

The condensation subalgebra has dimension 68 and is generated by

$$z1, z3.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.70.

$$1f(1a) \longleftrightarrow 1a(75a) \longleftrightarrow 1b(45a) \longleftrightarrow 1d(31a) \longleftrightarrow 1e(89a) \longleftrightarrow 2a(120a)$$

FIGURE 5.70. Ext-quiver of J_1 in characteristic 7

Characteristic 11: We take the subgroup N with number 34 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b, babacaca, abab\}.$$

The condensation subalgebra has dimension 198 and is generated by

$$z5, z3.$$

The basic algebra has dimension 38 and its Ext-quiver is given in Figure 5.71.

Characteristic 19: Let H be the subgroup with number 28 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z1, z2,$$

and has dimension 107. The basic algebra has dimension 30 and its Ext-quiver is given in Figure 5.72.

5.5.4 M_{22}

The order of the Mathieu group M_{22} is $2^7 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 = 443520$.

Characteristic 2: We take the subgroup N with number 109 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^2a^2, ab^2a\}.$$

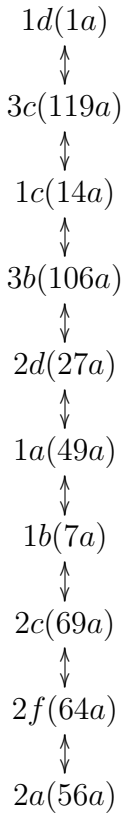


FIGURE 5.71. Ext-quiver of J_1 in characteristic 11

The condensation subalgebra has dimension 5504 and is generated by

$$z10, z3,$$

and the elements corresponding to the words

$$ab^2ab, ba^2ba, a^3b^2a.$$

The basic algebra has dimension 799 and its Ext-quiver is given in Figure 5.73.

Characteristic 3: We take the subgroup N with number 122 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^2ca, a^2c^2, a, a^7cac\}.$$

$$1d(1a) \longleftrightarrow 1e(55a) \longleftrightarrow 1c(22a) \longleftrightarrow 1b(34a) \longleftrightarrow 1a(43a) \longleftrightarrow 2c(77a)$$

FIGURE 5.72. Ext-quiver of J_1 in characteristic 19

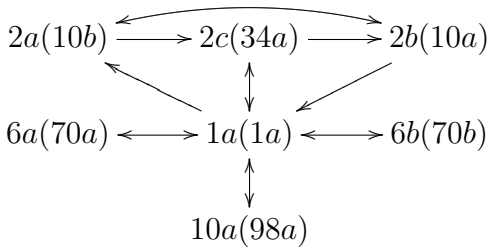


FIGURE 5.73. Ext-quiver of M_{22} in characteristic 2

The condensation subalgebra has dimension 145 and is generated by

$$z7, z5, z2.$$

The basic algebra has dimension 51 and its Ext-quiver is given in Figure 5.74.

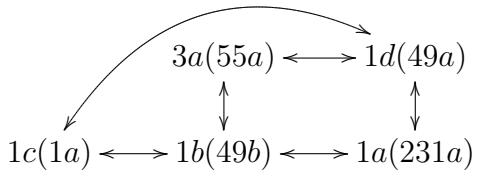


FIGURE 5.74. Ext-quiver of M_{22} in characteristic 3

Characteristic 5: Let H be the subgroup with number 135 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z3, z6, z1,$$

and has dimension 36. The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.75.

$$1d(1a) \longleftrightarrow 1b(98a) \longleftrightarrow 1c(133a) \longleftrightarrow 2b(21a)$$

FIGURE 5.75. Ext-quiver of M_{22} in characteristic 5

Characteristic 7: Let H be the subgroup with number 109 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z6, z4,$$

and has dimension 98 over F. The basic algebra has dimension 11 and its Ext-quiver is given in Figure 5.76.

$$1c(1a) \longleftrightarrow 1a(54a) \leftrightarrow 1b(45a) \circlearrowleft$$

FIGURE 5.76. Ext-quiver of M_{22} in characteristic 7

Characteristic 11: Let H be the subgroup with number 109 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z6, z4,$$

and has dimension 98. The basic algebra has dimension 11 and its Ext-quiver is given in Figure 5.77.

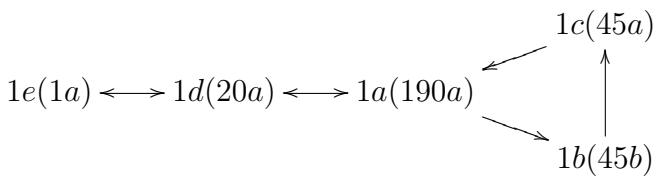


FIGURE 5.77. Ext-quiver of M_{22} in characteristic 11

5.5.5 J_2

The order of the Janko group J_2 is $2^7 \cdot 3^3 \cdot 5^2 \cdot 7 = 604800$.

Characteristic 2: We take the subgroup N with number 139 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababcbabca\}.$$

The condensation subalgebra has dimension 24288 and is generated by

$$z2,$$

and the elements corresponding to the words

$$ababacacabacb, cbabc, abacbacacb.$$

The basic algebra has dimension 1592 and its Ext-quiver is given in Figure 5.78.

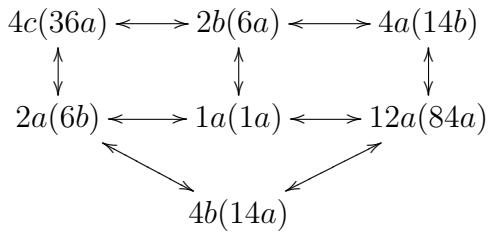


FIGURE 5.78. Ext-quiver of J_2 in characteristic 2

Characteristic 3: We take the subgroup N with number 118 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababac, abababcbac, abababacbc, baba, abababab\}.$$

The condensation subalgebra has dimension 660 and is generated by

$$z7, z8, z5, z9(z1, z3).$$

The basic algebra has dimension 204 and its Ext-quiver is given in Figure 5.79.

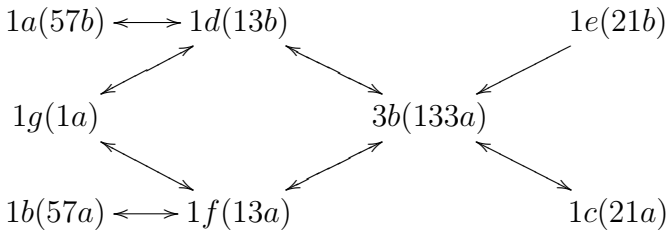


FIGURE 5.79. Ext-quiver of J_2 in characteristic 3

Characteristic 5: We take the subgroup N with number 129 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ab, abab, a^2, b^6, a^2bab^2ab, b^4\}.$$

The condensation subalgebra has dimension 95 and is generated by

$$z9, z8, z2.$$

The basic algebra has dimension 72 and its Ext-quiver is given in Figure 5.80.

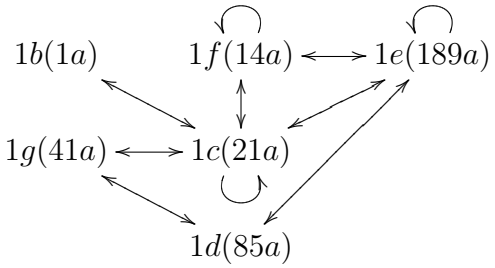


FIGURE 5.80. Ext-quiver of J_2 in characteristic 5

Characteristic 7: We take the subgroup N with number 128 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^4babac, a^2, a^2baba, a^4babab, a^3bab^2a^2, a^4b^2\}.$$

The condensation subalgebra has dimension 81 and is generated by

$$z2, z8, z1.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.81.

$$1d(1a) \longleftrightarrow 1c(89a) \longleftrightarrow 2c(199a) \longleftrightarrow 1b(101a) \longleftrightarrow 2b(124a) \longleftrightarrow 1a(36a)$$

FIGURE 5.81. Ext-quiver of J_2 in characteristic 7

5.5.6 M_{23}

The order of the Mathieu group M_{23} is $2^7 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 23 = 10200960$.

Characteristic 2: We take the subgroup N with number 78 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{baba^2, ab^2a\}.$$

The condensation subalgebra has dimension 23212 and is generated by

$$z11, z9, z6$$

and the element corresponding to the word

$$b^3.$$

The basic algebra has dimension 1513 and its Ext-quiver is given in Figure 5.82.

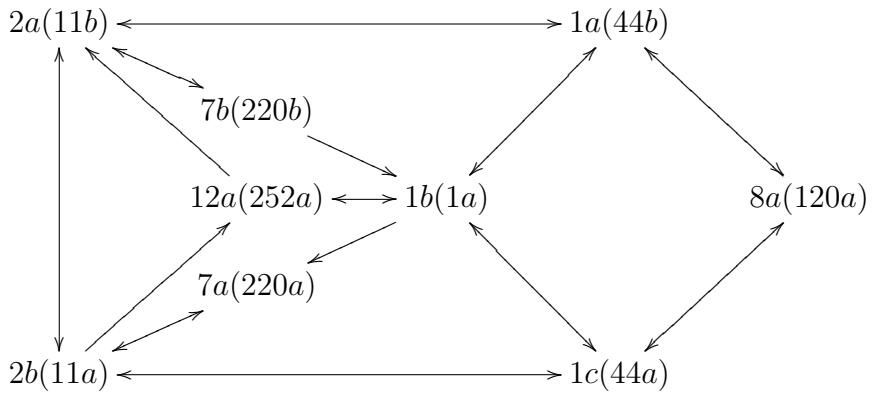


FIGURE 5.82. Ext-quiver of M_{23} in characteristic 2

Characteristic 3: We take the subgroup N with number 162 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abab, a^3bab^3a^2, a^3b^4a, a^2ba^2b^2a^2b, a^2ba^2ba^2b^2, a^2\}.$$

The condensation subalgebra has dimension 534 and is generated by

$$z8, z4, z3, z1.$$

The basic algebra has dimension 81 and its Ext-quiver is given in Figure 5.83.

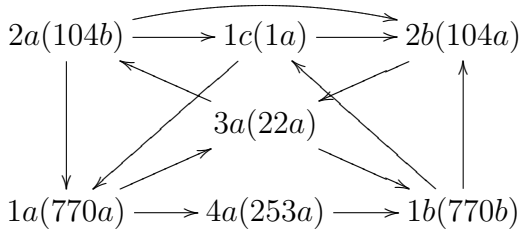


FIGURE 5.83. Ext-quiver of M_{23} in characteristic 3

Characteristic 5: Let H be the subgroup with number 178 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z9, z4,$$

and has dimension 62 over F. The basic algebra has dimension 81 and its Ext-quiver is given in Figure 5.84.

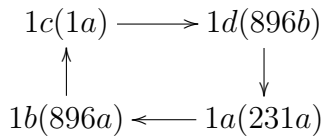


FIGURE 5.84. Ext-quiver of M_{23} in characteristic 5

Characteristic 7: Let H be the subgroup with number 185 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z4, z9(z1, z9),$$

and has dimension 19 over F. The basic algebra has dimension 11 and its Ext-quiver is given in Figure 5.85.

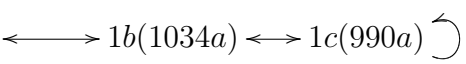


FIGURE 5.85. Ext-quiver of M_{23} in characteristic 7

Characteristic 11: We take the subgroup N with number 134 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^6ba^3, a^2, a^4ba^2ba^2, a^7ba^3b, aba^2ba\}.$$

The condensation subalgebra has dimension 2030 and is generated by

$$z5, z11.$$

The basic algebra has dimension 29 and its Ext-quiver is given in Figure 5.86.

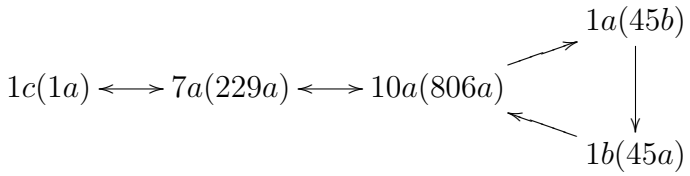


FIGURE 5.86. Ext-quiver of M_{23} in characteristic 11

Characteristic 23: We take the subgroup N with number 134 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^6ba^3, a^2, a^4ba^2ba^2, a^7ba^3b, aba^2ba\}.$$

The condensation subalgebra has dimension 2030 and is generated by

$$z8, z3.$$

The basic algebra has dimension 57 and its Ext-quiver is given in Figure 5.87.

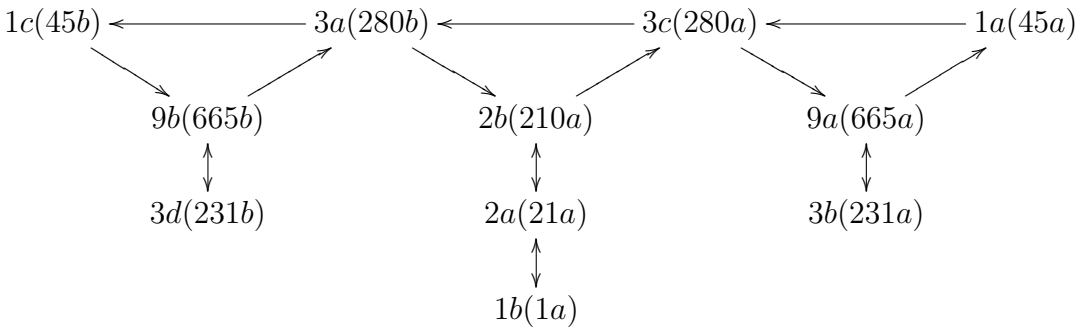


FIGURE 5.87. Ext-quiver of M_{23} in characteristic 23

5.5.7 *HS*

The order of the Higman-Sims group is $2^9 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 11 = 44352000$.

Characteristic 2: We take the subgroup N with number 253 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^2abab, babab\}.$$

The condensation subalgebra has dimension 100732 and is generated by

$$z5, z8, z4,$$

and the element corresponding to the word

$$bab^2.$$

The basic algebra has dimension 2462 and its Ext-quiver is given in Figure 5.88.

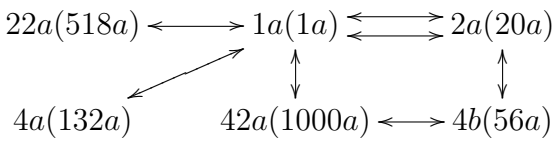


FIGURE 5.88. Ext-quiver of HS in characteristic 2

Characteristic 3: We take the subgroup N with number 520 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abab^4cba, b^2cabc, b^2, abcbca, ab^4cac, bacabc, ab^5cabc\}.$$

The condensation subalgebra has dimension 746 and is generated by

$$z3, z11, z8, z5.$$

The basic algebra has dimension 75 and its Ext-quiver is given in Figure 5.89.

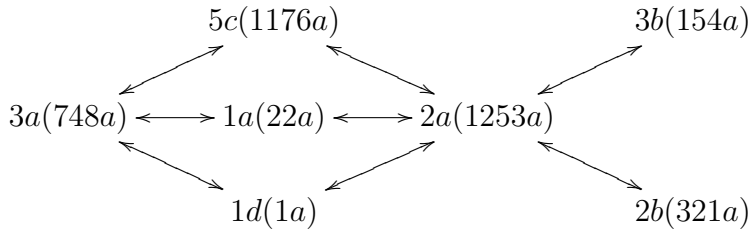


FIGURE 5.89. Ext-quiver of HS in characteristic 3

Characteristic 5: We take the subgroup N with number 513 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abacabc, acabcb, bac^2ac^2b, bc^2b, bacbac^3, bcabc^3a, abacabac\}.$$

The condensation subalgebra has dimension 1430 and is generated by

$$z5, z6, z8.$$

The basic algebra has dimension 444 and its Ext-quiver is given in Figure 5.90.

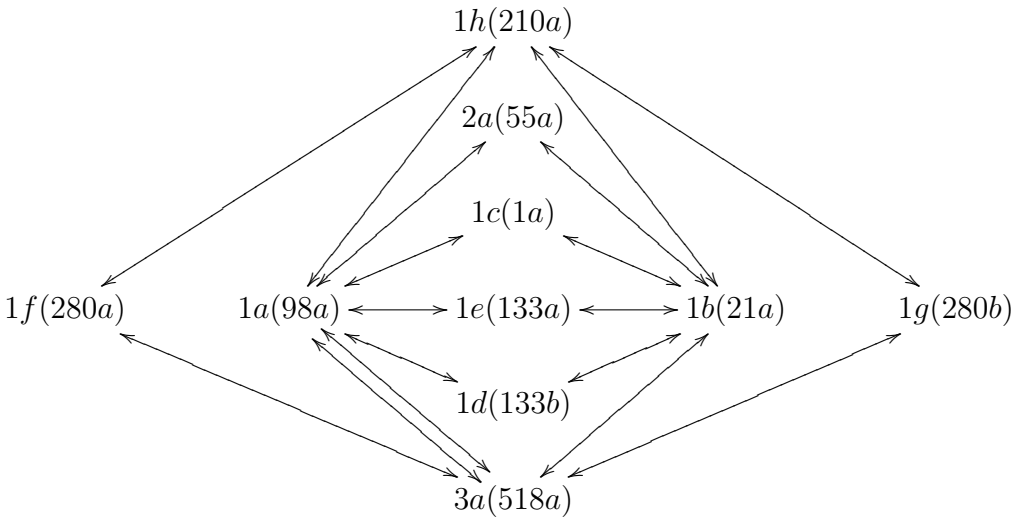


FIGURE 5.90. Ext-quiver of HS in characteristic 5

Characteristic 7: We take the subgroup N with number 563 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abab^4cb^2, bab^4cac, ab^2abacb^2a, ab^3ab^2cb, b^6c, b^4acab^2, abababcab, ab^6ca, b^6\}.$$

The condensation subalgebra has dimension 106 and is generated by

$$z5, z11, z1, z2.$$

The basic algebra has dimension 22 and its Ext-quiver is given in Figure 5.91.

$$1b(1a) \longleftrightarrow 1a(1055a) \longleftrightarrow 2a(2145a) \longleftrightarrow 1f(605a) \longleftrightarrow 2e(803a) \longleftrightarrow 1h(22a)$$

FIGURE 5.91. Ext-quiver of HS in characteristic 7

Characteristic 11: Let H be the subgroup with number 550 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z1, z2,$$

and has dimension 91 over F. The basic algebra has dimension 19 and its Ext-quiver is given in Figure 5.92.

$$1f(1a) \longleftrightarrow 1g(174a) \longleftrightarrow 1e(2346a) \longleftrightarrow 1b(854a) \leftarrow 1c(896a) \circlearrowleft$$

FIGURE 5.92. Ext-quiver of HS in characteristic 11

5.5.8 J_3

The order of the Janko group J_3 is $2^7 \cdot 3^5 \cdot 5 \cdot 17 \cdot 19 = 50232960$.

Characteristic 2: We take the subgroup N with number 105 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{ababab^3, bababab^2, bababa\}.$$

The condensation subalgebra has dimension 68932 and is generated by

$$z8, z2,$$

and the element corresponding to the word

$$a.$$

The basic algebra has dimension 1169 and its Ext-quiver is given in Figure 5.93.

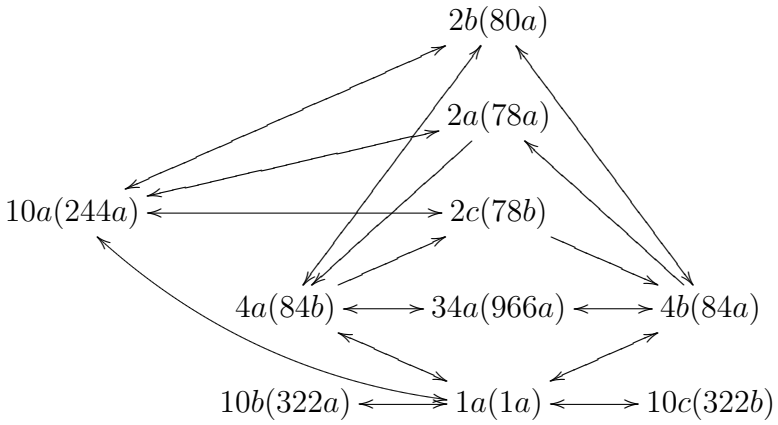


FIGURE 5.93. Ext-quiver of J_3 in characteristic 2

Characteristic 3: We take the subgroup N with number 120 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a, abab^5, a^4baba^2b^4, a^2bab^2ab^3, a^3b^2ab^2ab^2\}.$$

The condensation subalgebra has dimension 7926 and is generated by

$$z2, z10, z7,$$

and the element corresponding to the word

$$aba^2b^5ab^2.$$

The basic algebra has dimension 1754 and its Ext-quiver is given in Figure 5.94.

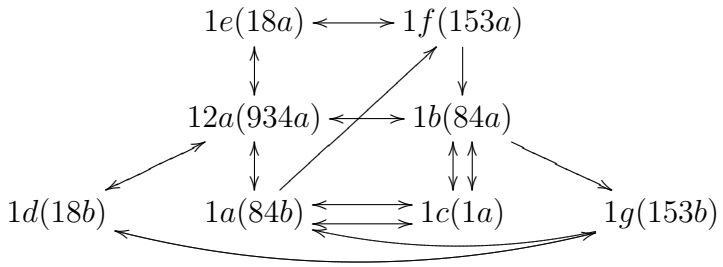


FIGURE 5.94. Ext-quiver of J_3 in characteristic 3

Characteristic 5: Let H be the subgroup with number 127 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z4, z1,$$

and has dimension 57. The basic algebra has dimension 7 and its Ext-quiver is given in Figure 5.95.

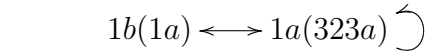


FIGURE 5.95. Ext-quiver of J_3 in characteristic 5

Characteristic 17: Let H be the subgroup with number 123 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z4, z5,$$

and has dimension 127. The basic algebra has dimension 34 and its Ext-quiver is given in Figure 5.96.

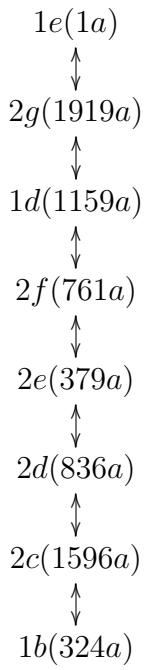


FIGURE 5.96. Ext-quiver of J_3 in characteristic 17

Characteristic 19: We take the subgroup N with number 127 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a, ab^5ab, ab^4abab^2ab^2, abab^2ab^5ab, bababab^5ab, ab^2abababb^5,$$
$$babababababa, ab^3ab^3ab^3ab^3\}.$$

The condensation subalgebra has dimension 405 and is generated by

$$z10, z9.$$

The basic algebra has dimension 35 and its Ext-quiver is given in Figure 5.97.

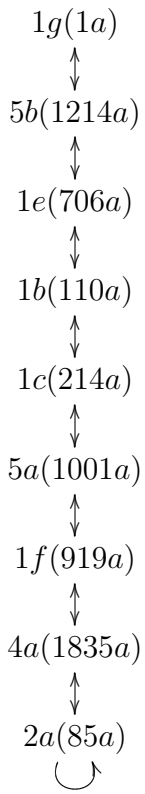


FIGURE 5.97. Ext-quiver of J_3 in characteristic 19

5.5.9 M_{24}

The order of the Mathieu group M_{24} is $2^{10} \cdot 3^3 \cdot 5 \cdot 7 \cdot 11 \cdot 23 = 244823040$.

Characteristic 2: We take the subgroup N with number 1162 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{aba^2ba, a^2ba^2b, a^3baba^3bab\}.$$

The condensation subalgebra has dimension 336224 and is generated by

$$z3, z7, z9$$

and the elements corresponding to the words

$$abababa^2bab, a^2bababab, aba^2ba^3.$$

The basic algebra has dimension 16151. We do not have enough information to construct the Ext-quiver or the basic algebra. The problem is that the programs discussed in this dissertation are impractical to use on an algebra of dimension 336224. We can find the projective indecomposable $e_H\mathbb{F}Ge_H$-modules in $e_H\mathbb{F}Ge_H$-modules smaller than the regular representation. The $e_H\mathbb{F}Ge_H$-module corresponding to the subgroup of M_{24} with number 1172 has dimension 35840 and contains 7 of the projective indecomposable $e_H\mathbb{F}Ge_H$-modules. This $e_H\mathbb{F}Ge_H$-module is within the range of our programs. The $e_H\mathbb{F}Ge_H$-modules corresponding to the subgroups of M_{24} with numbers 606 and 617 have dimensions 164864 and 144160, contain the other 6 projective indecomposable $e_H\mathbb{F}Ge_H$-modules. These $e_H\mathbb{F}Ge_H$-modules are not within the range of our programs.

A modification of the *C-MeatAxe* program `zmw` will bring these $e_H\mathbb{F}Ge_H$-modules within reach of our programs. We use these programs to construct a vector in the kernel of a peakword. The problem we are running into is that `zmw` requires that the elements the word is being evaluated on all fit into memory. Modifying `zmw` to work for sparse format matrices is not enough. Even though the matrices for the generators for $e_H\mathbb{F}Ge_H$ acting on these modules fit into about 5 megabytes, the result would have to be stored in matrix format and requires about 2.5 gigabytes. The approach we can take is to write a program for the *C-MeatAxe* package which applies a word to a vector without evaluating the word.

Table 5.6 summarizes what we know about the structure of projective indecomposable $e_H\mathbb{F}_2M_{24}e_H$-modules.

Number	Condensed Simple	Simple	Page
1172	$1c$	$1a$	171
	$6a$	$220a$	172
	$6b$	$220b$	172
	$8a$	$320a$	173
	$8c$	$320b$	173
	$46a$	$1242a$	174
	$64a$	$1792a$	174
617	$2a$	$44a$	
	$2b$	$44b$	
	$6c$	$120a$	
	$8b$	$252a$	
606	$1a$	$11a$	
	$1b$	$11b$	

TABLE 5.6. Projective indecomposable $e_H\mathbb{F}_2M_{24}e_H$-modules

Characteristic 3: We take the subgroup N with number 1410 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{babcbab, bcb^2cb, bab^2cb^2acbc, bab^3cb^3abc, cb^4c, b^4, abab^3\}.$$

The condensation subalgebra has dimension 2884 and is generated by

$$z3, z10, z1, z7(z1, z3), z5(z1, z8).$$

The basic algebra has dimension 213 and its Ext-quiver is given in Figure 5.102.

Characteristic 5: Let H be the subgroup with number 1493 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z4, z1,$$

$1c$
$1b \oplus 2b \oplus 2a \oplus 6a \oplus 8b \oplus 46a$
$4*1c \oplus 4*1a \oplus 2b \oplus 2a \oplus 6b \oplus 6c \oplus 8c \oplus 8a$
$4*1c \oplus 5*1b \oplus 1a \oplus 2*2b \oplus 2a \oplus 3*6c \oplus 6a \oplus 8b \oplus 8c \oplus 2*46a \oplus 64a$
$4*1c \oplus 3*1b \oplus 3*1a \oplus 2*2b \oplus 5*2a \oplus 3*6b \oplus 6c \oplus 4*8b \oplus 2*8c \oplus 8a$
$9*1c \oplus 3*1b \oplus 6*1a \oplus 5*2b \oplus 2a \oplus 4*6c \oplus 3*6a \oplus 8c \oplus 2*64a$
$5*1c \oplus 7*1b \oplus 6*1a \oplus 5*2b \oplus 5*2a \oplus 2*6b \oplus 6c \oplus 2*6a \oplus 4*8b \oplus 8c \oplus 8a$
$9*1c \oplus 5*1b \oplus 10*1a \oplus 3*2b \oplus 5*2a \oplus 6b \oplus 5*6c \oplus 6a \oplus 2*8a \oplus 2*46a$
$5*1c \oplus 9*1b \oplus 3*1a \oplus 4*2b \oplus 5*2a \oplus 3*6b \oplus 3*6c \oplus 2*6a \oplus 3*8b \oplus 8c$
$9*1c \oplus 6*1b \oplus 9*1a \oplus 6*2b \oplus 5*2a \oplus 4*6b \oplus 4*6c \oplus 2*6a \oplus 2*8b \oplus 8a \oplus 46a$
$7*1c \oplus 7*1b \oplus 6*1a \oplus 3*2b \oplus 4*2a \oplus 5*6c \oplus 4*6a \oplus 2*8b \oplus 2*8c \oplus 2*46a$
$4*1c \oplus 8*1b \oplus 11*1a \oplus 6*2b \oplus 8*2a \oplus 5*6b \oplus 4*6c \oplus 6a \oplus 7*8b \oplus 8c \oplus 8a \oplus 2*46a \oplus 64a$
$14*1c \oplus 12*1b \oplus 7*1a \oplus 4*2b \oplus 4*2a \oplus 7*6c \oplus 4*6a \oplus 8b \oplus 2*8c \oplus 8a \oplus 3*46a \oplus 64a$
$4*1c \oplus 9*1b \oplus 10*1a \oplus 10*2b \oplus 10*2a \oplus 4*6b \oplus 4*6c \oplus 3*6a \oplus 6*8b \oplus 4*8c \oplus 8a \oplus 64a$
$16*1c \oplus 9*1b \oplus 16*1a \oplus 7*2b \oplus 7*2a \oplus 3*6b \oplus 8*6c \oplus 4*6a \oplus 8b \oplus 2*8a \oplus 2*46a \oplus 2*64a$
$9*1c \oplus 15*1b \oplus 8*1a \oplus 9*2b \oplus 9*2a \oplus 3*6b \oplus 2*6c \oplus 6a \oplus 6*8b \oplus 3*8c \oplus 2*8a \oplus 46a$
$13*1c \oplus 8*1b \oplus 13*1a \oplus 8*2b \oplus 6*2a \oplus 6*6b \oplus 7*6c \oplus 3*6a \oplus 4*8b \oplus 2*8a \oplus 46a \oplus 64a$
$13*1c \oplus 12*1b \oplus 9*1a \oplus 8*2b \oplus 6*2a \oplus 3*6b \oplus 7*6c \oplus 7*6a \oplus 6*8b \oplus 2*8c \oplus 8a \oplus 46a$
$10*1c \oplus 8*1b \oplus 15*1a \oplus 4*2b \oplus 9*2a \oplus 5*6b \oplus 9*6c \oplus 3*6a \oplus 6*8b \oplus 2*8a \oplus 2*46a \oplus 64a$
$10*1c \oplus 15*1b \oplus 7*1a \oplus 8*2b \oplus 8*2a \oplus 6b \oplus 6*6c \oplus 6*6a \oplus 5*8b \oplus 3*8c \oplus 3*46a$
$10*1c \oplus 9*1b \oplus 16*1a \oplus 11*2b \oplus 8*2a \oplus 5*6b \oplus 6*6c \oplus 6a \oplus 3*8b \oplus 3*8c \oplus 2*8a \oplus 46a \oplus 64a$
$9*1c \oplus 13*1b \oplus 11*1a \oplus 6*2b \oplus 8*2a \oplus 4*6b \oplus 11*6c \oplus 5*6a \oplus 3*8b \oplus 8c \oplus 2*46a \oplus 64a$
$11*1c \oplus 7*1b \oplus 12*1a \oplus 8*2b \oplus 11*2a \oplus 4*6b \oplus 6c \oplus 3*6a \oplus 10*8b \oplus 8c \oplus 3*8a$
$15*1c \oplus 13*1b \oplus 15*1a \oplus 7*2b \oplus 3*2a \oplus 2*6b \oplus 10*6c \oplus 3*6a \oplus 3*8b \oplus 2*46a$
$6*1c \oplus 20*1b \oplus 5*1a \oplus 12*2b \oplus 9*2a \oplus 4*6b \oplus 2*6c \oplus 5*6a \oplus 5*8b \oplus 2*8c \oplus 2*8a \oplus 46a$
$12*1c \oplus 5*1b \oplus 16*1a \oplus 4*2b \oplus 11*2a \oplus 9*6b \oplus 10*6c \oplus 4*6a \oplus 8c \oplus 3*8a \oplus 2*46a \oplus 64a$
$9*1c \oplus 14*1b \oplus 8*1a \oplus 5*2b \oplus 7*2a \oplus 6b \oplus 2*6c \oplus 7*6a \oplus 7*8b \oplus 3*8c \oplus 8a \oplus 2*46a$
$11*1c \oplus 8*1b \oplus 17*1a \oplus 7*2b \oplus 5*2a \oplus 5*6b \oplus 8*6c \oplus 6a \oplus 6*8b \oplus 2*8a \oplus 2*46a \oplus 64a$
$9*1c \oplus 14*1b \oplus 4*1a \oplus 7*2b \oplus 5*2a \oplus 7*6c \oplus 2*6a \oplus 4*8b \oplus 8c \oplus 2*46a$
$7*1c \oplus 5*1b \oplus 5*1a \oplus 9*2b \oplus 8*2a \oplus 4*6b \oplus 5*6c \oplus 6a \oplus 4*8b \oplus 3*8c \oplus 2*8a \oplus 64a$
$7*1c \oplus 7*1b \oplus 10*1a \oplus 6*2b \oplus 4*2a \oplus 6b \oplus 10*6c \oplus 6*6a \oplus 3*8b \oplus 8a \oplus 46a \oplus 64a$
$7*1c \oplus 8*1b \oplus 9*1a \oplus 2*2b \oplus 10*2a \oplus 4*6b \oplus 6c \oplus 8*8b \oplus 2*8a$
$9*1c \oplus 11*1b \oplus 6*1a \oplus 5*2b \oplus 3*2a \oplus 3*6b \oplus 4*6c \oplus 4*6a \oplus 2*8b \oplus 46a$
$3*1c \oplus 6*1b \oplus 7*1a \oplus 10*2b \oplus 7*2a \oplus 4*6b \oplus 2*6c \oplus 3*6a \oplus 2*8b \oplus 2*8c \oplus 8a$
$3*1c \oplus 6*1b \oplus 10*1a \oplus 2b \oplus 5*2a \oplus 4*6b \oplus 7*6c \oplus 3*6a \oplus 2*8b \oplus 2*8a \oplus 2*46a$
$8*1c \oplus 7*1b \oplus 5*1a \oplus 3*2b \oplus 3*2a \oplus 6c \oplus 3*6a \oplus 4*8b \oplus 8c \oplus 8a$
$6*1c \oplus 5*1b \oplus 5*1a \oplus 5*2b \oplus 2*2a \oplus 2*6b \oplus 4*6c \oplus 3*8b \oplus 8c \oplus 46a \oplus 64a$
$1c \oplus 7*1b \oplus 3*1a \oplus 6*2b \oplus 2*2a \oplus 6b \oplus 4*6c \oplus 3*6a \oplus 8b \oplus 8c \oplus 46a$
$5*1c \oplus 2*1b \oplus 5*1a \oplus 3*2b \oplus 5*2a \oplus 3*6b \oplus 2*6c \oplus 6a \oplus 2*8b \oplus 8c \oplus 2*8a \oplus 64a$
$6*1c \oplus 5*1b \oplus 5*1a \oplus 2b \oplus 2a \oplus 3*6c \oplus 2*6a \oplus 8b \oplus 46a$
$2*1c \oplus 5*1b \oplus 3*1a \oplus 4*2b \oplus 2a \oplus 6b \oplus 6c \oplus 6a \oplus 5*8b \oplus 8a$
$4*1c \oplus 3*1b \oplus 2*1a \oplus 2b \oplus 3*2a \oplus 2*6b \oplus 3*6c \oplus 6a \oplus 8c \oplus 8a$
$2*1c \oplus 4*1b \oplus 4*1a \oplus 4*2b \oplus 3*2a \oplus 6b \oplus 6c \oplus 2*6a \oplus 8b \oplus 8c$
$3*1c \oplus 3*1b \oplus 5*1a \oplus 2b \oplus 2a \oplus 6b \oplus 2*6c \oplus 6a \oplus 3*8b \oplus 2*8a \oplus 46a$
$4*1c \oplus 4*1b \oplus 2*1a \oplus 2*2b \oplus 3*6c \oplus 8b$
$1c \oplus 2*1b \oplus 3*2b \oplus 3*2a \oplus 6b \oplus 6a \oplus 2*8b \oplus 8c \oplus 46a$
$2*1c \oplus 1b \oplus 6*1a \oplus 2b \oplus 2a \oplus 6b \oplus 3*6c \oplus 2*6a \oplus 8a$
$2*1c \oplus 5*1b \oplus 2*1a \oplus 2b \oplus 2*2a \oplus 6b \oplus 3*8b \oplus 8a$
$3*1c \oplus 2*1b \oplus 1a \oplus 2b \oplus 2a \oplus 6b \oplus 6c \oplus 8b$
$1c \oplus 2*1b \oplus 1a \oplus 3*2b \oplus 2a \oplus 2*6c \oplus 2*6a \oplus 8b \oplus 8c$
$1b \oplus 5*1a \oplus 3*2a \oplus 2*6b \oplus 3*6c \oplus 2*8b \oplus 8a$
$5*1c \oplus 3*1b \oplus 1a \oplus 2a \oplus 6c \oplus 2*8b$
$2*1c \oplus 1b \oplus 1a \oplus 2*2b \oplus 2a \oplus 8b$
$2*1b \oplus 2*1a \oplus 2*2b \oplus 3*6c \oplus 6a$
$2*1b \oplus 1a \oplus 3*2a \oplus 2*6b \oplus 2*8b$
$4*1c \oplus 2*1a \oplus 6a$
$3*1b \oplus 1a \oplus 2*2b \oplus 6a \oplus 2*8b$
$1b \oplus 1a \oplus 2*2a \oplus 2*6b \oplus 2*6c \oplus 8a$
$1c \oplus 1b \oplus 2b \oplus 2a \oplus 6a \oplus 8c$
$2*1a \oplus 2b \oplus 6c$
$1b \oplus 8b$
$1c \oplus 2b$
$1b \oplus 2b \oplus 6c \oplus 6a$
$1a \oplus 2a \oplus 6b \oplus 8b$
$1c$

FIGURE 5.98. Radical series for projective cover of $1c$

$6a$
$1a \oplus 2a$
$1b \oplus 1c \oplus 46a$
$2b \oplus 6b \oplus 8b \oplus 8c$
$2*1a \oplus 3*1c \oplus 6a \oplus 2*6c \oplus 64a$
$2*1b \oplus 1a \oplus 2*2a \oplus 2b \oplus 8b \oplus 8a$
$1b \oplus 2*1a \oplus 2*1c \oplus 2a \oplus 2*2b \oplus 6c \oplus 46a$
$4*1b \oplus 1c \oplus 2*2b \oplus 6a \oplus 6b \oplus 8c$
$3*1a \oplus 3*1c \oplus 3*2a \oplus 6a \oplus 3*6b \oplus 6c \oplus 8a \oplus 64a$
$2*1b \oplus 3*1a \oplus 4*1c \oplus 2a \oplus 2*6a \oplus 6c \oplus 8a \oplus 46a$
$3*1b \oplus 3*1a \oplus 2*2b \oplus 2*6c \oplus 4*8b \oplus 8c$
$5*1b \oplus 2*1a \oplus 5*1c \oplus 2a \oplus 4*6c \oplus 2*46a \oplus 64a$
$2*1b \oplus 5*2a \oplus 5*2b \oplus 6a \oplus 6b \oplus 8b \oplus 3*8c$
$8*1a \oplus 4*1c \oplus 2a \oplus 3*2b \oplus 6a \oplus 2*6c \oplus 64a$
$7*1b \oplus 1a \oplus 3*1c \oplus 2*2a \oplus 2b \oplus 6a \oplus 2*8b \oplus 8a$
$1b \oplus 4*1a \oplus 3*1c \oplus 5*2a \oplus 2*2b \oplus 5*6b \oplus 8b \oplus 8a \oplus 46a$
$4*1b \oplus 3*1a \oplus 6*1c \oplus 3*2b \oplus 5*6a \oplus 3*6c$
$3*1b \oplus 6*1a \oplus 3*1c \oplus 3*2a \oplus 2b \oplus 2*6b \oplus 2*6c \oplus 5*8b \oplus 2*8a$
$5*1b \oplus 2*1a \oplus 5*1c \oplus 2*2a \oplus 2b \oplus 6a \oplus 7*6c \oplus 3*46a$
$4*1b \oplus 2*1a \oplus 1c \oplus 4*2a \oplus 7*2b \oplus 6a \oplus 3*8b \oplus 4*8c$
$2*1b \oplus 11*1a \oplus 5*1c \oplus 2*2a \oplus 2b \oplus 2*6b \oplus 7*6c \oplus 64a$
$7*1b \oplus 4*1c \oplus 5*2a \oplus 2*2b \oplus 6a \oplus 6b \oplus 4*8b$
$6*1a \oplus 4*1c \oplus 3*2a \oplus 4*2b \oplus 6a \oplus 3*6b \oplus 6c \oplus 8b$
$10*1b \oplus 3*1a \oplus 2*1c \oplus 2a \oplus 4*2b \oplus 5*6a \oplus 8b \oplus 46a$
$2*1b \oplus 6*1a \oplus 4*1c \oplus 7*2a \oplus 2b \oplus 7*6b \oplus 8b \oplus 3*8a$
$6*1b \oplus 2*1a \oplus 9*1c \oplus 2*2b \oplus 5*6a \oplus 3*6c \oplus 2*46a$
$3*1b \oplus 5*1a \oplus 1c \oplus 2a \oplus 5*2b \oplus 6c \oplus 6*8b \oplus 8c \oplus 8a$
$5*1b \oplus 3*1a \oplus 5*1c \oplus 2a \oplus 11*6c \oplus 2*46a \oplus 64a$
$1b \oplus 6*2a \oplus 5*2b \oplus 2*8b \oplus 3*8c$
$10*1a \oplus 4*1c \oplus 2*2b \oplus 5*6c \oplus 64a$
$8*1b \oplus 1c \oplus 3*2a \oplus 2*2b \oplus 6a \oplus 5*8b$
$4*1a \oplus 4*1c \oplus 7*2a \oplus 2b \oplus 7*6b$
$7*1b \oplus 3*1a \oplus 3*1c \oplus 4*2b \oplus 7*6a$
$2*1b \oplus 6*1a \oplus 4*1c \oplus 3*2a \oplus 2b \oplus 3*6b \oplus 2*8b \oplus 4*8a$
$6*1b \oplus 1a \oplus 4*1c \oplus 2*6a \oplus 2*6c \oplus 2*46a$
$2*1a \oplus 2a \oplus 6*2b \oplus 2*8b \oplus 2*8c$
$2*1b \oplus 3*1a \oplus 2*1c \oplus 7*6c \oplus 46a \oplus 64a$
$2*1b \oplus 1c \oplus 3*2a \oplus 2*2b \oplus 3*8b$
$2*1a \oplus 3*1c \oplus 2*6c$
$3*1b \oplus 2a \oplus 2*2b \oplus 6a \oplus 8b$
$2*1a \oplus 1c \oplus 3*2a \oplus 3*6b \oplus 8a$
$3*1b \oplus 1a \oplus 2*1c \oplus 2b \oplus 2*6a$
$1b \oplus 2*1a \oplus 1c \oplus 2b \oplus 6b \oplus 2*8b \oplus 8a$
$2*1b \oplus 1a \oplus 1c \oplus 6a \oplus 3*6c \oplus 46a$
$1b \oplus 2*2a \oplus 2*2b \oplus 8b \oplus 8c$
$3*1a \oplus 1c \oplus 6c$
$2*1b \oplus 1c \oplus 2b \oplus 8b$
$1c \oplus 2a \oplus 6b$
$1b \oplus 1a \oplus 1c \oplus 2b \oplus 2*6a$
$1b \oplus 2*1a \oplus 2a \oplus 6b \oplus 8b \oplus 8a$
$2*1b \oplus 2*1c \oplus 6c$
$2a \oplus 2*2b \oplus 8b \oplus 8c$
$3*1a \oplus 3*6c$
$2*1b \oplus 2a \oplus 2*8b$
$1a \oplus 2*1c$
$2*1b \oplus 2*2b \oplus 6a$
$1a \oplus 2*2a \oplus 2*6b \oplus 8a$
$1b \oplus 2*1c \oplus 6a$
$1a \oplus 2b \oplus 8b$
$1b \oplus 1a \oplus 6c$
$2a \oplus 2b$
$1a \oplus 6c$
$1b \oplus 8b$
$1c \oplus 6b$
$6a$

$6b$
$1b \oplus 1c \oplus 6a$
$1a \oplus 2a \oplus 8b$
$1b \oplus 1c \oplus 6c \oplus 46a$
$2a \oplus 2*2b \oplus 6b \oplus 2*8c$
$4*1a \oplus 3*1c \oplus 2b \oplus 6c \oplus 6a \oplus 64a$
$3*1b \oplus 1a \oplus 2*1c \oplus 2a \oplus 8a \oplus 8b$
$1b \oplus 1a \oplus 1c \oplus 2*2a \oplus 2b \oplus 6b \oplus 6c \oplus 8b \oplus 46a$
$3*1b \oplus 1a \oplus 2*1c \oplus 2*2b \oplus 6b \oplus 6c \oplus 2*6a \oplus 8c$
$1b \oplus 4*1a \oplus 2*1c \oplus 3*2a \oplus 2*6b \oplus 2*6c \oplus 8a \oplus 8b \oplus 64a$
$4*1b \oplus 2*1a \oplus 4*1c \oplus 2*2a \oplus 2b \oplus 2*6c \oplus 6a \oplus 8a \oplus 46a$
$2*1b \oplus 2*1a \oplus 1c \oplus 2a \oplus 4*2b \oplus 6c \oplus 3*8b \oplus 2*8c$
$3*1b \oplus 5*1a \oplus 4*1c \oplus 2a \oplus 2b \oplus 6b \oplus 4*6c \oplus 46a \oplus 64a$
$4*1b \oplus 3*1c \oplus 4*2a \oplus 2*2b \oplus 6b \oplus 2*6a \oplus 2*8b \oplus 8c$
$6*1a \oplus 4*1c \oplus 2*2a \oplus 2*2b \oplus 6b \oplus 6c \oplus 6a \oplus 8b$
$7*1b \oplus 2*1a \oplus 1c \oplus 2a \oplus 2*2b \oplus 6c \oplus 3*6a \oplus 8b \oplus 46a$
$1b \oplus 6*1a \oplus 3*1c \oplus 6*2a \oplus 3*2b \oplus 5*6b \oplus 2*8a \oplus 46a$
$5*1b \oplus 4*1a \oplus 7*1c \oplus 3*2b \oplus 3*6c \oplus 3*6a \oplus 46a$
$5*1b \oplus 3*1a \oplus 3*1c \oplus 2a \oplus 3*2b \oplus 6b \oplus 2*6c \oplus 8a \oplus 6*8b \oplus 8c$
$3*1b \oplus 2*1a \oplus 6*1c \oplus 3*2a \oplus 6b \oplus 9*6c \oplus 6a \oplus 2*46a \oplus 64a$
$3*1b \oplus 2*1a \oplus 1c \oplus 6*2a \oplus 5*2b \oplus 2*6a \oplus 4*8b \oplus 4*8c$
$2*1b \oplus 13*1a \oplus 5*1c \oplus 2a \oplus 2*2b \oplus 6b \oplus 4*6c \oplus 64a$
$10*1b \oplus 3*1c \oplus 3*2a \oplus 4*2b \oplus 6b \oplus 3*8b \oplus 46a$
$3*1a \oplus 3*1c \oplus 5*2a \oplus 4*2b \oplus 5*6b \oplus 6c \oplus 6a \oplus 8a$
$6*1b \oplus 4*1a \oplus 5*1c \oplus 2a \oplus 3*2b \oplus 7*6a \oplus 8b$
$3*1b \oplus 8*1a \oplus 4*1c \oplus 5*2a \oplus 5*6b \oplus 6c \oplus 3*8a \oplus 2*8b$
$8*1b \oplus 1a \oplus 7*1c \oplus 2a \oplus 2*2b \oplus 4*6c \oplus 3*6a \oplus 2*46a$
$2*1b \oplus 3*1a \oplus 1c \oplus 2*2a \oplus 7*2b \oplus 8a \oplus 5*8b \oplus 2*8c$
$3*1b \oplus 6*1a \oplus 2*1c \oplus 2b \oplus 12*6c \oplus 2*46a \oplus 64a$
$3*1b \oplus 1c \oplus 6*2a \oplus 3*2b \oplus 4*8b \oplus 2*8c$
$8*1a \oplus 6*1c \oplus 2a \oplus 6b \oplus 3*6c \oplus 64a$
$7*1b \oplus 2*2a \oplus 3*2b \oplus 3*6a \oplus 3*8b$
$5*1a \oplus 4*1c \oplus 6*2a \oplus 2b \oplus 7*6b \oplus 8a$
$8*1b \oplus 3*1a \oplus 3*1c \oplus 3*2b \oplus 4*6a$
$2*1b \oplus 3*1a \oplus 3*1c \oplus 2*2a \oplus 3*2b \oplus 2*6b \oplus 3*8a \oplus 3*8b$
$3*1b \oplus 2*1a \oplus 3*1c \oplus 5*6c \oplus 2*6a \oplus 2*46a$
$1b \oplus 2*1a \oplus 3*2a \oplus 3*2b \oplus 2*8b \oplus 2*8c$
$2*1b \oplus 3*1a \oplus 3*1c \oplus 3*6c \oplus 46a \oplus 64a$
$2*1b \oplus 1c \oplus 2a \oplus 3*2b \oplus 8b$
$1a \oplus 2*1c \oplus 2a \oplus 6b \oplus 2*6c$
$2*1b \oplus 1a \oplus 1c \oplus 2a \oplus 2b \oplus 2*6a \oplus 8b$
$1b \oplus 3*1a \oplus 1c \oplus 2*2a \oplus 2*6b \oplus 8a \oplus 8b$
$4*1b \oplus 2*1c \oplus 2b \oplus 6c \oplus 6a$
$1a \oplus 1c \oplus 2a \oplus 2*2b \oplus 6b \oplus 8a \oplus 8b \oplus 8c$
$1b \oplus 3*1a \oplus 3*6c \oplus 6a \oplus 46a$
$2*1b \oplus 1c \oplus 2a \oplus 2b \oplus 2*8b$
$1a \oplus 2*1c$
$2*1b \oplus 2*2b \oplus 6a$
$1a \oplus 2*2a \oplus 2*6b \oplus 8a$
$1b \oplus 1a \oplus 2*1c \oplus 6a$
$1b \oplus 1a \oplus 2b \oplus 2*8b$
$1b \oplus 1a \oplus 3*6c$
$1b \oplus 2*2a \oplus 2b \oplus 8b \oplus 8c$
$2*1a \oplus 1c \oplus 6c$
$1b \oplus 2b \oplus 8b$
$1c \oplus 2a \oplus 6b$
$1b \oplus 1a \oplus 1c \oplus 2b \oplus 2*6a$
$2*1a \oplus 2a \oplus 6b$
$2*1b \oplus 1c$
$2b \oplus 8b$
$2*6c$
$2a \oplus 8b$
$1a \oplus 1c$
$1b \oplus 2b$
$6b$

FIGURE 5.99. Radical series for projective covers of $6a$ and $6b$

$$
\begin{array}{c}
8a \\
6c \oplus 46a \\
2{*}1b \oplus 1c \oplus 8b \oplus 8c \\
1c \oplus 1a \oplus 2b \oplus 2a \oplus 6b \oplus 64a \\
1b \oplus 2{*}1c \oplus 1a \oplus 2b \oplus 2{*}6a \\
1b \oplus 1c \oplus 3{*}1a \oplus 2a \oplus 8a \oplus 8b \\
2{*}1b \oplus 1c \oplus 2{*}6c \oplus 46a \\
1b \oplus 2{*}2b \oplus 2a \oplus 6b \oplus 8c \\
1c \oplus 3{*}1a \oplus 2b \oplus 2{*}6c \oplus 6a \\
2{*}1b \oplus 1a \oplus 2{*}2a \oplus 2{*}8b \\
1b \oplus 2{*}1c \oplus 1a \oplus 2b \oplus 2a \oplus 6b \\
3{*}1b \oplus 1c \oplus 1a \oplus 3{*}2b \oplus 2{*}6a \\
1b \oplus 2{*}1c \oplus 3{*}1a \oplus 2{*}2a \oplus 3{*}6b \oplus 2{*}8a \oplus 8b \\
2{*}1b \oplus 4{*}1c \oplus 3{*}6c \oplus 6a \oplus 2{*}46a \\
1b \oplus 1a \oplus 2{*}2b \oplus 2a \oplus 2{*}8b \oplus 2{*}8c \\
1b \oplus 3{*}1c \oplus 4{*}1a \oplus 4{*}6c \oplus 64a \\
2{*}1b \oplus 2b \oplus 4{*}2a \oplus 2{*}8b \\
2{*}1c \oplus 4{*}1a \oplus 2{*}2b \oplus 6c \\
5{*}1b \oplus 2{*}2b \oplus 6a \oplus 8b \\
3{*}1c \oplus 1a \oplus 5{*}2a \oplus 4{*}6b \oplus 8a \\
1b \oplus 3{*}1c \oplus 2{*}1a \oplus 2b \oplus 6c \oplus 4{*}6a \\
3{*}1b \oplus 1c \oplus 4{*}1a \oplus 8a \oplus 3{*}8b \\
4{*}1b \oplus 1c \oplus 3{*}6c \oplus 2{*}46a \\
4{*}2b \oplus 2{*}2a \oplus 3{*}8c \\
2{*}1c \oplus 4{*}1a \oplus 4{*}6c \oplus 64a \\
2{*}1b \oplus 1c \oplus 3{*}2a \oplus 3{*}8b \\
3{*}1c \oplus 3{*}1a \oplus 2b \oplus 6b \\
4{*}1b \oplus 3{*}2b \oplus 2{*}6a \\
1c \oplus 2{*}1a \oplus 3{*}2a \oplus 3{*}6b \oplus 2{*}8a \\
2{*}1b \oplus 3{*}1c \oplus 2{*}6a \\
2{*}1a \oplus 2b \oplus 2{*}8b \\
2{*}1b \oplus 1a \oplus 3{*}6c \oplus 46a \\
1b \oplus 2{*}2b \oplus 2{*}2a \oplus 8b \\
2{*}1c \oplus 2{*}1a \oplus 2{*}6c \\
2{*}1b \oplus 2b \oplus 8b \\
1c \oplus 2{*}2a \oplus 2{*}6b \\
1b \oplus 1c \oplus 2{*}1a \oplus 2b \oplus 2{*}6a \\
1b \oplus 1c \oplus 2{*}1a \oplus 8a \oplus 8b \\
1b \oplus 1c \oplus 2{*}6c \oplus 46a \\
2b \oplus 2a \oplus 2{*}8c \\
2{*}1a \oplus 6c \oplus 64a \\
1b \oplus 1c \oplus 8b \\
1c \\
1b \oplus 2b \oplus 6a \\
1a \oplus 2a \oplus 6b \oplus 8a \\
1b \oplus 1c \\
2b \oplus 8b \\
1a \oplus 2{*}6c \\
1b \oplus 2a \oplus 8b \\
1c \oplus 1a \\
1b \oplus 2b \\
2a \oplus 6b \\
6a \\
1a \\
1b \\
2b \\
6c \\
8b \\
1c \\
1b \\
8a
\end{array}
\qquad
\begin{array}{c}
8c \\
1a \oplus 64a \\
1c \oplus 8a \\
6c \oplus 8b \oplus 46a \\
1c \oplus 2{*}1b \oplus 6c \\
2{*}2a \oplus 2b \oplus 6b \oplus 8c \\
3{*}1a \oplus 1c \oplus 2b \oplus 6a \\
1a \oplus 2{*}1c \oplus 2{*}1b \\
1b \oplus 2a \oplus 6b \oplus 8b \\
1a \oplus 1c \oplus 2b \oplus 6c \oplus 6a \oplus 6b \\
1a \oplus 1b \oplus 2{*}6c \oplus 6a \oplus 8b \\
1a \oplus 1c \oplus 2{*}1b \oplus 2{*}2a \oplus 6c \oplus 8b \oplus 46a \\
1c \oplus 1b \oplus 2a \oplus 2{*}2b \oplus 2{*}8c \\
4{*}1a \oplus 1c \oplus 1b \oplus 3{*}2b \oplus 6c \oplus 64a \\
1a \oplus 2{*}1c \oplus 2{*}1b \oplus 2a \oplus 6b \oplus 8b \oplus 2{*}8a \\
1a \oplus 2{*}1c \oplus 2a \oplus 2b \oplus 6b \\
1a \oplus 2{*}1c \oplus 2{*}1b \oplus 2b \oplus 2{*}6a \\
3{*}1a \oplus 1c \oplus 1b \oplus 2{*}2a \oplus 6c \oplus 2{*}6b \oplus 8b \oplus 8a \\
2{*}1c \oplus 3{*}1b \oplus 2{*}6c \oplus 6a \oplus 46a \\
1a \oplus 1b \oplus 2a \oplus 2{*}2b \oplus 8c \oplus 2{*}8b \\
3{*}1a \oplus 2{*}1c \oplus 1b \oplus 4{*}6c \oplus 46a \oplus 64a \\
1b \oplus 3{*}2a \oplus 2b \oplus 8c \oplus 2{*}8b \\
4{*}1a \oplus 3{*}1c \oplus 2b \oplus 6c \\
5{*}1b \oplus 2a \oplus 2{*}2b \oplus 6a \oplus 8b \\
2{*}1a \oplus 2{*}1c \oplus 4{*}2a \oplus 2b \oplus 3{*}6b \oplus 8a \\
1a \oplus 3{*}1c \oplus 2{*}1b \oplus 2b \oplus 6c \oplus 3{*}6a \\
3{*}1a \oplus 1c \oplus 2{*}1b \oplus 2a \oplus 6b \oplus 2{*}8b \oplus 8a \\
2{*}1c \oplus 3{*}1b \oplus 2{*}6c \oplus 6a \oplus 2{*}46a \\
1a \oplus 2a \oplus 3{*}2b \oplus 2{*}8c \oplus 8b \\
3{*}1a \oplus 2{*}1c \oplus 1b \oplus 4{*}6c \\
1c \oplus 1b \oplus 3{*}2a \oplus 2b \oplus 2{*}8b \\
3{*}1a \oplus 2{*}1c \oplus 2b \oplus 6c \\
4{*}1b \oplus 2{*}2b \oplus 6a \oplus 8b \\
1a \oplus 1c \oplus 3{*}2a \oplus 4{*}6b \oplus 8a \\
1a \oplus 2{*}1c \oplus 2{*}1b \oplus 2b \oplus 3{*}6a \\
3{*}1a \oplus 1c \oplus 2b \oplus 2{*}8b \oplus 8a \\
1a \oplus 3{*}1b \oplus 3{*}6c \oplus 46a \\
1b \oplus 2a \oplus 2{*}2b \oplus 8c \oplus 8b \\
1a \oplus 2{*}1c \oplus 2{*}6c \oplus 64a \\
1b \oplus 2a \oplus 2b \oplus 8b \\
1a \oplus 1c \oplus 2a \oplus 6b \\
1a \oplus 1c \oplus 2{*}1b \oplus 2b \oplus 6a \\
1a \oplus 1b \oplus 2a \oplus 6b \oplus 8b \oplus 8a \\
2{*}1c \oplus 1b \oplus 6c \oplus 6a \\
1a \oplus 2a \oplus 2b \oplus 8c \\
2{*}1a \oplus 6c \oplus 46a \\
1c \oplus 1b \oplus 8b \\
1c \\
1b \oplus 2b \oplus 6a \\
1a \oplus 2a \oplus 6b \\
1c \oplus 1b \\
2b \oplus 8b \\
1a \oplus 6c \\
2a \\
1a \\
1b \\
2a \oplus 6b \\
1a \oplus 1c \\
8b \\
6c \\
8c
\end{array}
$$

FIGURE 5.100. Radical series for projective covers of $8a$ and $8c$

$46a$
$1c \oplus 1b \oplus 8c$
$1a \oplus 2a \oplus 2b \oplus 6b \oplus 64a$
$2*1c \oplus 1a \oplus 6a \oplus 8a$
$1a \oplus 1b \oplus 6c \oplus 8b$
$2*1b \oplus 2a \oplus 6c \oplus 46a$
$2a \oplus 2b \oplus 6b \oplus 8c$
$2*1c \oplus 2*1a \oplus 2b \oplus 6a \oplus 6c$
$1a \oplus 1b \oplus 2a \oplus 8b$
$1c \oplus 1a \oplus 1b \oplus 2a \oplus 2b \oplus 6b \oplus 46a$
$2*1c \oplus 1a \oplus 2*1b \oplus 2*2b \oplus 6a$
$2*1c \oplus 2*1a \oplus 1b \oplus 2a \oplus 2*6b \oplus 2*6c \oplus 8a \oplus 8b$
$3*1c \oplus 2*1b \oplus 2a \oplus 6a \oplus 2*6c \oplus 46a$
$2*1a \oplus 1b \oplus 2a \oplus 3*2b \oplus 2*8b \oplus 2*8c$
$2*1c \oplus 3*1a \oplus 2*1b \oplus 2*6c \oplus 64a$
$1c \oplus 2*1b \oplus 2*2a \oplus 2*2b \oplus 6a \oplus 6b \oplus 8b$
$1c \oplus 3*1a \oplus 2a \oplus 2b \oplus 6a \oplus 6c$
$1c \oplus 1a \oplus 4*1b \oplus 2a \oplus 2b \oplus 6a \oplus 8b$
$2*1c \oplus 2*1a \oplus 1b \oplus 3*2a \oplus 2b \oplus 3*6b \oplus 8a$
$3*1c \oplus 2*1a \oplus 2*1b \oplus 2*2b \oplus 2*6a \oplus 2*6c$
$1c \oplus 2*1a \oplus 2*1b \oplus 2b \oplus 8a \oplus 4*8b$
$2*1c \oplus 1a \oplus 2*1b \oplus 4*6c \oplus 2*46a$
$1b \oplus 3*2a \oplus 3*2b \oplus 6a \oplus 2*8c$
$2*1c \oplus 6*1a \oplus 2a \oplus 6b \oplus 2*6c \oplus 64a$
$1c \oplus 3*1b \oplus 2*2a \oplus 2*8b$
$2*1c \oplus 2*1a \oplus 2*2a \oplus 2*2b \oplus 2*6b$
$2*1c \oplus 1a \oplus 4*1b \oplus 2*2b \oplus 2*6a$
$2*1c \oplus 2*1a \oplus 1b \oplus 2*2a \oplus 2*6b \oplus 2*8a \oplus 8b$
$2*1c \oplus 1b \oplus 2*6a \oplus 2*6c \oplus 46a$
$2*1a \oplus 1b \oplus 2a \oplus 2b \oplus 2*8b$
$1c \oplus 2*1b \oplus 3*6c \oplus 46a$
$2*2a \oplus 2*2b \oplus 8c$
$1c \oplus 3*1a \oplus 2*6c$
$2*1b \oplus 2a \oplus 2*8b$
$2*1c \oplus 1a \oplus 2a \oplus 2b \oplus 2*6b$
$1a \oplus 2*1b \oplus 2*2b \oplus 2*6a$
$1c \oplus 2*1a \oplus 2a \oplus 6b \oplus 8a$
$1c \oplus 2*1b \oplus 6a \oplus 6c \oplus 46a$
$1a \oplus 2b \oplus 8b \oplus 8c$
$1c \oplus 1b \oplus 6c \oplus 64a$
$2a \oplus 2b$
$1a \oplus 6c$
$1b \oplus 8b$
$1c \oplus 2a \oplus 6b$
$1b \oplus 2b \oplus 6a$
$1c \oplus 1a \oplus 8a$
$46a$

$64a$
$8a$
$6c \oplus 46a$
$1c \oplus 2*1b \oplus 8c$
$1a \oplus 2b \oplus 2a \oplus 6b$
$1a \oplus 2*1c \oplus 6a$
$1a \oplus 1b \oplus 8b$
$1b \oplus 6c$
$2b \oplus 2a \oplus 6b$
$1a \oplus 2b \oplus 6c \oplus 6a$
$1a \oplus 1b \oplus 2a \oplus 8b$
$1c \oplus 1b \oplus 2a \oplus 6b$
$1a \oplus 1c \oplus 1b \oplus 2*2b \oplus 6a$
$2*1a \oplus 1c \oplus 1b \oplus 6b \oplus 8a \oplus 8b$
$2*1c \oplus 1b \oplus 2*6c \oplus 46a$
$1b \oplus 2b \oplus 2a \oplus 8c$
$2*1a \oplus 1c \oplus 6c \oplus 64a$
$1b \oplus 2a \oplus 8b$
$1a \oplus 1c \oplus 2b$
$2*1b \oplus 2b \oplus 6a$
$1a \oplus 1c \oplus 2*2a \oplus 6b \oplus 8a$
$1c \oplus 6c \oplus 6a$
$1a \oplus 1b \oplus 8b$
$1b \oplus 6c \oplus 46a$
$2b \oplus 2a \oplus 8c$
$2*1a \oplus 1c \oplus 6c$
$1b \oplus 2a \oplus 8b$
$1a \oplus 1c \oplus 2b \oplus 6b$
$1b \oplus 2b \oplus 6a$
$1a \oplus 1c \oplus 2a \oplus 6b$
$1c \oplus 1b \oplus 6a$
$1a \oplus 8b$
$1b \oplus 6c$
$2b \oplus 2a$
$1a \oplus 6c$
$1b \oplus 8b$
$1c \oplus 2a \oplus 6b$
$1a \oplus 1b \oplus 2b \oplus 6a$
$1a \oplus 1c \oplus 8a$
$6c \oplus 46a$
$8c$
$64a$

FIGURE 5.101. Radical series for projective covers of $46a$ and $64a$

and has dimension 27. The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.103.

Characteristic 7: We take the subgroup N with number 1471 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{abababa^2babcb, baba^3bacbc, a^3ba^3bababa^3babc^2, ba^3bc^2a, a^2ba^2bababa^2c^2, a^2ba^2b,$$
$$aba^3ba^2bababa^2b, a^2baba^3ba^2babab, a^3baba^2bababa^2b, a^2\}.$$

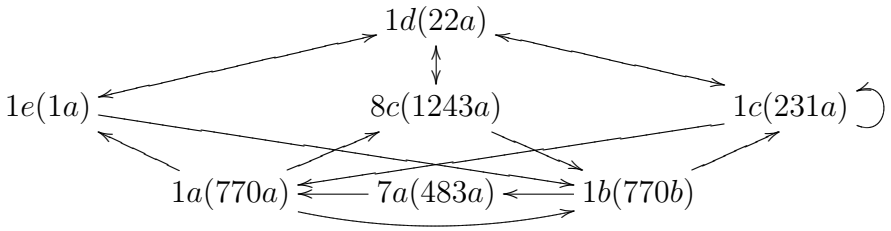


FIGURE 5.102. Ext-quiver of M_{24} in characteristic 3

FIGURE 5.103. Ext-quiver of M_{24} in characteristic 5

The condensation subalgebra has dimension 173 and is generated by

$$z8, z6(z1, z3), z7(z1, z3).$$

The basic algebra has dimension 11 and its Ext-quiver is given in Figure 5.104.

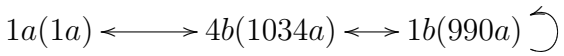


FIGURE 5.104. Ext-quiver of M_{24} in characteristic 7

Characteristic 11: We take the subgroup N with number 1396 in the Table of Marks which is generated by $\{a, b, c, d\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{cac^3d, ac^2acba, bcbc^3, b, babc^3ac, bcabac, cabacb, bacacabcac, ac^3ac\}.$$

The condensation subalgebra has dimension 1232 and is generated by

$$z10, z5.$$

The basic algebra has dimension 58 and its Ext-quiver is given in Figure 5.105.

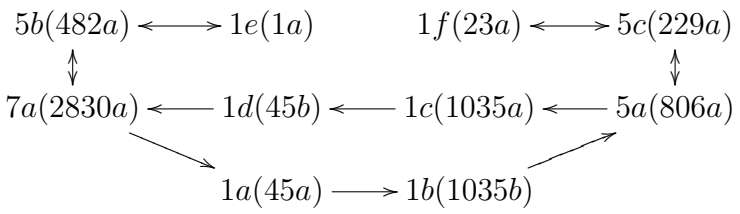


FIGURE 5.105. Ext-quiver of M_{24} in characteristic 11

Characteristic 23 We take the subgroup N with number 1347 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{b^2, abab^2ac^2a, b^2abc^2a, b^3c^2, b^3c^2a, b^3ac^2, ab^3ac^2a\}.$$

The condensation subalgebra has dimension 3131 and is generated by

$$z2, z8.$$

The basic algebra has dimension 69 and its Ext-quiver is given in Figure 5.106.

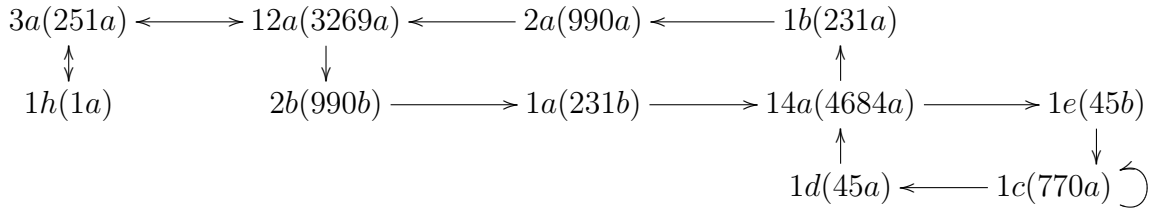


FIGURE 5.106. Ext-quiver of M_{24} in characteristic 23

5.5.10 *McL*

The order of the McLaughlin group is $2^7 \cdot 3^6 \cdot 5^3 \cdot 7 \cdot 11 = 898128000$.

Characteristic 2: We take the subgroup N with number 304 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^8babca, a^4b^2cbc^2, a^3b^2ab, a\}.$$

The condensation subalgebra has dimension 15318 and is generated by

$$z8, z6, z5, z11.$$

The basic algebra has dimension 1004 and its Ext-quiver is given in Figure 5.107.

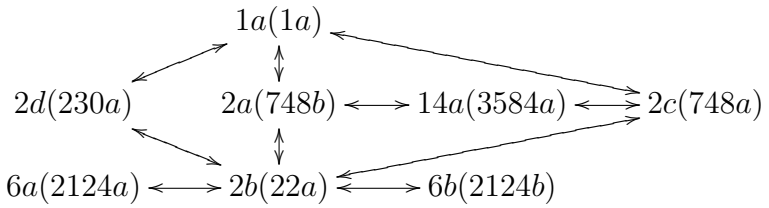


FIGURE 5.107. Ext-quiver of McL in characteristic 2

Characteristic 5: We take the subgroup N with number 304 in the Table of Marks which is generated by $\{a, b, c\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^8babca, a^4b^2cbc^2, a^3b^2ab, a\}.$$

The condensation subalgebra has dimension 15318 and is generated by

$$z7, z2.$$

The basic algebra has dimension 788 and its Ext-quiver is given in Figure 5.108.

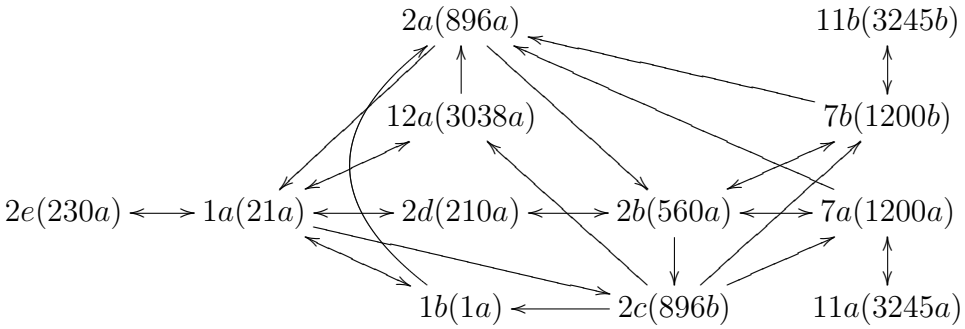


FIGURE 5.108. Ext-quiver of McL in characteristic 5

Characteristic 7: We take the subgroup N with number 355 in the Table of Marks which is generated by $\{a, b\}$. The condensation subgroup H we used is the normal subgroup of N generated by

$$\{a^5b^3a^3, a^3bab^2ab^3aba^2, a^3bab^2abababab, a^3baba^2b^2aba, a^4ba^2bab^2ab^4, a^7b^5a\}.$$

The condensation subalgebra has dimension 704 and is generated by

$$z7, z6.$$

The basic algebra has dimension 14 and its Ext-quiver is given in Figure 5.109.

$$1b(1a) \longleftrightarrow 1a(4499a) \longleftrightarrow 4b(3520a)$$

FIGURE 5.109. Ext-quiver of McL in characteristic 7

Characteristic 11: Let H be the subgroup with number 324 in the Table of Marks. The algebra $e_H\mathbb{F}Ge_H$ is generated by the words

$$z4, z7,$$

and has dimension 389. The basic algebra has dimension 19 and its Ext-quiver is given in Figure 5.110.

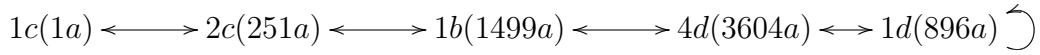


FIGURE 5.110. Ext-quiver of McL in characteristic 11

5.5.11 He

The order of the Held group is $2^{10} \cdot 3^3 \cdot 5^2 \cdot 7^3 \cdot 17 = 4030387200$.

Characteristic 7: Let H be the subgroup with number 1587 in the Table of Marks. The algebra $e_H \mathbb{F}G e_H$ is generated by the words

$$z2, z8, z3$$

and has dimension 1779. The basic algebra has dimension 487 and its Ext-quiver is given in Figure 5.111.

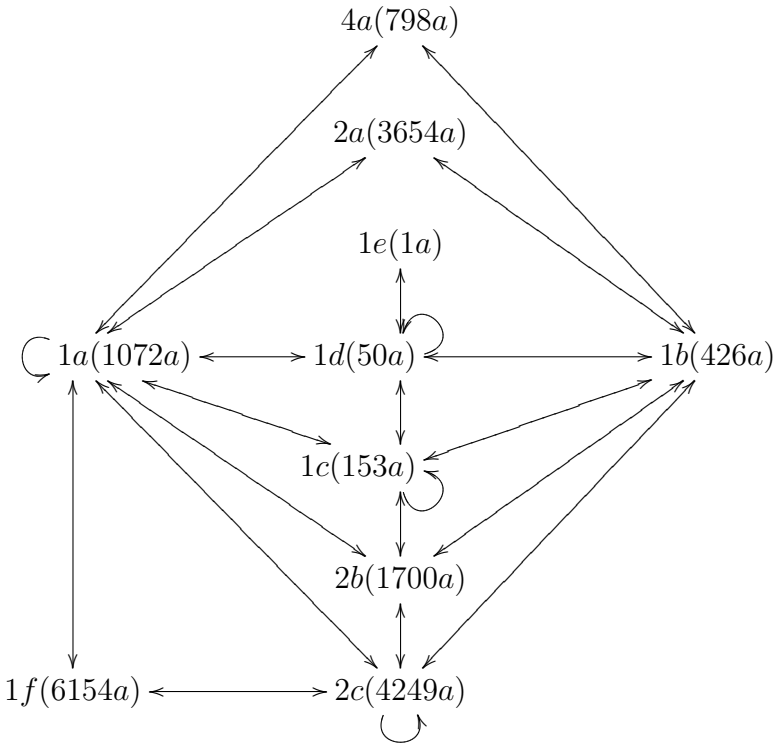


FIGURE 5.111. Ext-quiver of He in characteristic 7

Appendix A
Programs

A.1 The *GAP*-procedure FindCondSubgroup

```
#
# Find the largest condensation subgroup inside its normalizer
#
# Return straight line programs for this subgroup and normalizing
# p-elements in terms of the generators of its normalizer
#
InstallGlobalFunction( FindCondSubgroup, function( grp, prime )
    local gp, N, Nrep, Hrep, ellist, r, i, j, Krep, syl, sylgens,
          f, sylext, hom, kgens, kdim, numberintom, permchars,
          ct, fus, posconj, isconj, overcond, innorm, endpos,
          begpos, mattomtr, mattom, pos, normlist, result, grpinfo,
          tom, names, modtafel, dec, fil, strlinekgens, gpgens,
          strlinekgensstring, strlinesylgens, strlinesylgensstring,
          modchars, info, script;
# get group info
    grpinfo:=InfoKond(grp,prime,1);
    tom:=TableOfMarks(grp);
    gp:=UnderlyingGroup(tom);
    info:=StandardGeneratorsInfo(tom);
    script:=fail;
    if info[1].script <> fail and info[1].ATLAS = true then
       if IsStandardGeneratorsOfGroup(info[1],gp,
            GeneratorsOfGroup(gp)) then
          script:=true;
       fi;
    fi;
# need this line because M24 gens are in agreement but script
# info is flawed
    if grp = "M24" then script:=true; fi;
# construct the normalizer of the condensation subgroup and
# the conjugate of H in it
    N:=NormalizerTom(tom,grpinfo.numberinburn);
    Nrep:=RepresentativeTom(tom,N);
    Hrep:=RepresentativeTom(tom,grpinfo.numberinburn);
    normlist:=NormalSubgroups(Nrep);
    posconj:=Filtered(normlist,x-> Size(x)=Size(Hrep));
    isconj:=List([1..Length(posconj)],x->IsConjugate(gp,
       posconj[x],Hrep));
    pos:=Position(isconj,true);
    Krep:=posconj[pos];
# get the prime-Sylow subgroup of the normalizer and extend H by it
    syl:=SylowSubgroup(Nrep,prime);
```

```
    sylgens:=[];
    names:=["a","b","c","d","e"];
    f:=FreeGroup(names{[1..Length(GeneratorsOfGroup(Nrep))]});
    sylext:=Group(Concatenation(GeneratorsOfGroup(Krep),
      GeneratorsOfGroup(syl)));
# find the possible entries in the table of marks for H extended
# by p-elements
    mattom:=MatTom(tom);;
    mattomtr:=TransposedMat(mattom);;
    begpos:=Position(mattomtr[1],Size(gp)/Size(sylext));
    endpos:=Length(mattomtr[1])-Position(Reversed(mattomtr[1]),
        Size(gp)/Size(sylext))+1;
# find which of the above possible classes are contained in the
# normalizer of H and contain H
    innorm:=List([begpos..endpos],x->[x,ContainedTom(tom,x,N)]);
    overcond:=List(innorm,x->[x[1],x[2],ContainingTom(tom,
      grpinfo.numberinburn, x[1])]);
    posconj:=Filtered(overcond,x-> x[2]>0 and x[3]>0);
# sort through all of the posconj to see which is conj to H extended
    if Length(posconj)>1 then
        isconj:=List(posconj,x->[x[1],IsConjugate(gp,sylext,
                RepresentativeTom(tom,x[1]))]);
        for i in [1..Length(isconj)] do
                if isconj[i][2]=true then
                        numberintom:=isconj[i][1];
                fi;
        od;
    else
        numberintom:=posconj[1][1];
    fi;
# get the permutation characters from table of marks
    ct:=CharacterTable(grp);
    fus:=FusionCharTableTom(ct,tom);
    if IsList ( fus[1] ) then
       fus :=fus[1];
       Print("warning there is more than one possible fusion\n");
    fi;
    permchars:=PermCharsTom(fus,tom);;
# use permchars to calculate the dimension of the intermediate
# algebra kFGe
    kdim:=MatScalarProducts(ct,[permchars[grpinfo.numberinburn]],
        [permchars[numberintom]]);
# get modular char information and find perm rep containing all
# simple reps
    modtafel:=CharacterTable(Concatenation(grp,"mod",String(prime)));
    modchars:=Irr(modtafel);
    dec:=Decomposition(modchars,Restricted(ct,modtafel,permchars),5);
    fil:=Filtered(dec,x->ForAll(x,y->y>0));
# get straight line programs for gens of H and H extended
    hom:=GroupHomomorphismByImagesNC(f,Nrep,GeneratorsOfGroup(f),
        GeneratorsOfGroup(Nrep));
```

```
    kgens:=[];
    strlinekgens:=[];
    strlinekgensstring:=[];
    for i in [1..Length(GeneratorsOfGroup(Krep))] do
        kgens[i]:=PreImagesRepresentative(hom,
           GeneratorsOfGroup(Krep)[i]);
        strlinekgens[i]:=SplitString(String(kgens[i]),"*");
        strlinekgens[i]:=JoinStringsWithSeparator(strlinekgens[i],"");
        strlinekgens[i]:=StraightLineProgram(strlinekgens[i],
          names{[1..Length(GeneratorsOfGroup(Nrep))]});
        strlinekgensstring[i]:=StringOfResultOfStraightLineProgram(
              strlinekgens[i],names{[1..Length(
                 GeneratorsOfGroup(Nrep))]}, "LATEX");
    od;
    strlinesylgens:=[];
    strlinesylgensstring:=[];
    for i in [1..Length(GeneratorsOfGroup(syl))] do
        sylgens[i]:=PreImagesRepresentative(hom,
           GeneratorsOfGroup(syl)[i]);
        strlinesylgens[i]:=SplitString(String(sylgens[i]),"*");
        strlinesylgens[i]:=JoinStringsWithSeparator(
           strlinesylgens[i],"");
        strlinesylgens[i]:=StraightLineProgram(strlinesylgens[i],
          names{[1..Length(GeneratorsOfGroup(Nrep))]});
        strlinesylgensstring[i]:=StringOfResultOfStraightLineProgram(
          strlinesylgens[i],names{[1..Length(
             GeneratorsOfGroup(Nrep))]}, "LATEX");
    od;
# construct record of results to return
    result:=rec();
    result.SimpleRep:=Position(dec,fil[Length(fil)]);
    result.Normalizer:=N;
    result.kdim:=kdim[1][1];
    result.knumber:=numberintom;
    result.gens:=[strlinekgens,strlinesylgens];
    result.strline:=[strlinekgensstring,strlinesylgensstring];
    result.horder:=Size(Hrep);
    result.korder:=Size(sylext);
    result.script:=script;
    return(result);
end );
```

A.2 The *GAP*-procedure GetPerms

```
#
# construct permutations for G on G/H
#
InstallGlobalFunction( GetPerms, function( grp, grpinfo, subinfo)
local rep, j, outfile, name, infile, kperms, hperms, perms;
hperms:=[];kperms:=[];perms:=[];
#check to see if the standard generators in the table of marks
#agree with the modular atlas
if subinfo.script <> fail then
#check if there are normalizing p-elements
if subinfo.korder > subinfo.horder then
#get vectors fixed by K but not by G and a representation of G
#constructs the perms in BasicWorkingSpace
   rep:=FindVecs( grp, grpinfo.numberinburn, grpinfo.numberinburn,
                  subinfo.gens[1],
                  subinfo.horder, subinfo.gens[2]);
#read in the permutations if they exist
   if rep <> 0 then
      for j in [1..2+Maximum( 1, Length(subinfo.gens[1]))
                    +Length(subinfo.gens[2])] do
          outfile:=Filename( BasicWorkingSpace, "perm");
          name:=OutputTextFile( outfile, false);
          Process( BasicWorkingSpace, zpr, NoneIn, name,
                   [ "-G", Concatenation("perm.",String(j))]);
          CloseStream( name);
          infile:= Filename( BasicWorkingSpace, "perm" );;
          name:= InputTextFile(infile);;
          Read(name);
          hperms[j]:=MeatAxe.Perms[1];
      od;
#get vectors fixed by H but not by G and a representation of G
#constructs the perms in BasicWorkingSpace
      rep:=FindVecs( grp, grpinfo.numberinburn, grpinfo.numberinburn,
                     Flat(subinfo.gens), subinfo.korder, []);
#read in the permutations if they exist
      if rep <> 0 then
         for j in [1..2+Maximum( 1, Length(subinfo.gens[1]))
                       +Length(subinfo.gens[2])] do
             outfile:=Filename( BasicWorkingSpace, "perm");
             name:=OutputTextFile( outfile, false);
             Process( BasicWorkingSpace, zpr, NoneIn, name,
                      [ "-G", Concatenation("perm.",String(j))]);
             CloseStream( name);
             infile:= Filename( BasicWorkingSpace, "perm" );;
             name:= InputTextFile(infile);;
             Read(name);
             kperms[j]:=MeatAxe.Perms[1];
         od;
      fi;
```

```
   fi;
else
#do the same as above but just H (K=H)
   rep:=FindVecs( grp, grpinfo.numberinburn, grpinfo.numberinburn,
                  subinfo.gens[1],
                  subinfo.horder, []);
   if rep <> 0 then
      for j in [1..2+Maximum( 1, Length(subinfo.gens[1]))] do
          outfile:=Filename( BasicWorkingSpace, "perm");
          name:=OutputTextFile( outfile, false);
          Process( BasicWorkingSpace, zpr, NoneIn, name,
                   [ "-G", Concatenation("perm.",String(j))]);
          CloseStream( name);
          infile:= Filename( BasicWorkingSpace, "perm" );;
          name:= InputTextFile(infile);;
          Read(name);
          hperms[j]:=MeatAxe.Perms[1];
      od;
   fi;
fi;
else
  rep := 0;
fi;
#if the above method fails, use MakePermOnCond(Norm) which calls
#ImprimPerm
if rep = 0 then
   if subinfo.korder > subinfo.horder then
      perms:=MakePermOnCondNorm(grp,subinfo.Normalizer,
              subinfo.gens[1], subinfo.gens[2]);
      return([perms[3],perms[2]]);
   else
      perms:=MakePermOnCond(grp,subinfo.Normalizer,subinfo.gens[1]);
      return([perms[1],[2,[]]]);
   fi;
fi;
return([[2,hperms], [2,kperms]]);
end );
```

A.3 The *GAP*-procedure FindAlgGens

```
#
# Find generators for the algebra eFGe
#
InstallGlobalFunction( FindAlgGens,
      function(grp,prime,condperms,normperms,dim,numsylgens)
local listgens, numgens, t, n, counter, genset, orbit, brows, gens,
      perms, id, secperm, firstperm, s, chartab, i, basedim, maxpos,
      max, permlist, grpinfo, outfile, options, infile, name;
grpinfo:=InfoKond(grp,prime,1);
chartab:=CharacterTable(grp);
orbit:=String(Size(chartab)/(grpinfo.permchar[1]));
Unbind(chartab);
#make a vector to seed the spinning
id:=NullMat(1,dim,GF(grpinfo.splitf));
id[1][1]:=One(GF(prime));
outfile:=Filename(BasicWorkingSpace, "tvec");
PrintTo(outfile,MeatAxeString([id[1]],grpinfo.splitf));
Process(BasicWorkingSpace, zcv, NoneIn, NoneOut, ["tvec", "vec"]);
Unbind(id);
s:=grpinfo.norm;t:=0;
n:=12;
listgens:=[];
genset:=0;brows:=0;
MeatAxe.MatrixRows:=1;
numgens:=numsylgens;firstperm:=1;secperm:=2;
#make the condensed matrices for the normalizing p-elements if
#they exist
if (numsylgens > 0) then
  gens:=normperms[2]{[Length(normperms[2])-numsylgens+1..
        Length(normperms[2])]};
  permlist:=[normperms[1],normperms[2]{[1..Length(normperms[2])-
        numsylgens]}];
  Condense( permlist, gens, grpinfo.splitf);
  for i in [1..numsylgens] do
    options:=[ Concatenation( "Kdg.", String(i)),
               Concatenation( "z.", String(i))];
    Process(BasicWorkingSpace, zcv, NoneIn, NoneOut, options);
  od;
  options:=[ "-g", String(numsylgens), "-b", "bas", "z", "vec"];
  Process(BasicWorkingSpace, newzsp, NoneIn, NoneOut, options);
  outfile:=Filename( BasicWorkingSpace, "res" );
  name:=OutputTextFile( outfile, false );
  Process(BasicWorkingSpace, zpr, NoneIn, name, [ "-Gs", "bas" ]);
  CloseStream(name);
  infile:=Filename(BasicWorkingSpace, "res");
  Read(infile);
  basedim:=MeatAxe.MatrixRows;
else
  basedim:=1;
```

```
  Process(BasicWorkingSpace, Cp, NoneIn, NoneOut, ["vec", "bas"]);
fi;
#make elements of eFGe and check generation
while basedim <  dim do
  genset:=genset+1;
  listgens[genset]:=rec();
  listgens[genset].genlist:=[];
  listgens[genset].basedim:=[];
  if genset=1 then
    if Length(normperms)>0 then
      listgens[genset].resstrline:=ResultOfStraightLineProgram(fr,
          normperms[2]{[1..normperms[1]]});
    fi;
    listgens[genset].bigresstrline:=ResultOfStraightLineProgram(fr,
        condperms[2]{[1..condperms[1]]});
  else
    secperm:=secperm+1;
    if secperm=firstperm then
      secperm:=secperm+1;
    fi;
    if secperm > 11 then
      firstperm:=firstperm+1;
      secperm:=1;
    fi;
    listgens[genset].name:=Concatenation(String(firstperm),":",
        String(secperm));
    if Length(normperms)>0 then
      listgens[genset].resstrline:=ResultOfStraightLineProgram(fr,
          [listgens[1].resstrline[firstperm],
              listgens[1].resstrline[secperm]]);
    fi;
    listgens[genset].bigresstrline:=ResultOfStraightLineProgram(fr,
        [listgens[1].bigresstrline[firstperm],
            listgens[1].bigresstrline[secperm]]);
  fi;
  if Length(normperms)>0 then
    gens:=listgens[genset].resstrline{[1,2,3,4,5,6,7,8,9,10,11]};
    perms:=[normperms[1],normperms[2]{[1..Length(normperms[2])-
                numsylgens]}];
  else
    gens:=listgens[genset].bigresstrline{[1,2,3,4,5,6,7,8,9,10,11]};
    perms:=[condperms[1],condperms[2]{[1..Length(condperms[2])]}];
  fi;
  Condense( perms, gens, grpinfo.splitf);
  for i in [1..11] do
    options:=[ Concatenation( "Kdg.", String(i)),
               Concatenation("kdg.", String(i))];
    Process(BasicWorkingSpace, zcv, NoneIn, NoneOut, options);
    options:=[ Concatenation("kdg.",String(i)), "try.1"];
    Process(BasicWorkingSpace, Cp, NoneIn, NoneOut, options);
    options:=[ "-g", "1", "-b", Concatenation("bas.",String(i)),
```

```
            "try", "vec"];
  Process(BasicWorkingSpace, newzsp, NoneIn, NoneOut, options);
  if (basedim > 1) then
    options:=[ "bas", Concatenation( "bas.", String(i)),
               Concatenation( "clean.", String(i)), "junk"];
    Process(BasicWorkingSpace, zcl, NoneIn, NoneOut, options);
    options:=[ Concatenation( "clean.", String(i)),
               Concatenation( "bas.", String(i))];
    Process(BasicWorkingSpace, zef, NoneIn, NoneOut, options);
  fi;
    outfile:=Filename(BasicWorkingSpace, "res");
    name:=OutputTextFile( outfile, false);
    options:=[ "-Gs", Concatenation("bas.",String(i))];
    Process(BasicWorkingSpace, zpr, NoneIn, name, options);
    CloseStream(name);
    infile:=Filename(BasicWorkingSpace,"res");
    Read(infile);
    listgens[genset].basedim[i]:=MeatAxe.MatrixRows;
od;
while ((basedim < dim) and (ForAny(listgens[genset].basedim,
                x->x>0))) do
  max:=Maximum(listgens[genset].basedim);
  maxpos:=Position(listgens[genset].basedim,max);
  Append(listgens[genset].genlist,[maxpos]);
  options:=[ Concatenation("kdg.",String(maxpos)),
             Concatenation("z.",String(numgens+1))];
  Process(BasicWorkingSpace, Mv, NoneIn, NoneOut, options);
  options:=[ "newbas", Concatenation("bas.",String(maxpos)), "bas"];
  Process( BasicWorkingSpace, zpt, NoneIn, NoneOut, options);
  Process( BasicWorkingSpace, zef, NoneIn, NoneOut,
           ["newbas", "bas"]);
  numgens:=numgens+1;
  options:=[ "-g", String(numgens), "-c", "-b", "tempbas", "-x",
             String(999), "z", "bas"];
  Process( BasicWorkingSpace, newzsp, NoneIn, NoneOut, options);
  Process( BasicWorkingSpace, Mv, NoneIn, NoneOut,
           ["tempbas", "bas"]);
  outfile:=Filename( BasicWorkingSpace, "res");
  name:=OutputTextFile( outfile, false);
  Process(BasicWorkingSpace, zpr, NoneIn, name, ["-Gs", "bas"]);
  CloseStream(name);
  infile:=Filename(BasicWorkingSpace, "res");
  Read(infile);
  basedim:=MeatAxe.MatrixRows;
  for i in [1..11] do
    options:=[ "bas", Concatenation("bas.",String(i)),
               Concatenation( "clean.", String(i)), "junk"];
    Process(BasicWorkingSpace, zcl, NoneIn, NoneOut, options);
    options:=[ Concatenation("clean.",String(i)),
               Concatenation("bas.",String(i))];
    Process(BasicWorkingSpace, zef, NoneIn, NoneOut, options);
```

```
      options:=[ "-Gs", Concatenation("bas.", String(i))];
      outfile:=Filename( BasicWorkingSpace, "res");
      name:=OutputTextFile( outfile, false);
      Process(BasicWorkingSpace, zpr, NoneIn, name, options);
      CloseStream(name);
      infile:=Filename(BasicWorkingSpace,"res");
      Read(infile);
      listgens[genset].basedim[i]:=MeatAxe.MatrixRows;
    od;
  od;
od;
#check to see if any generators are redundant
BigSieve("z",numgens,dim,listgens,numsylgens);
return(listgens);
end);
```

A.4 The *GAP*-procedure Condense

```
InstallGlobalFunction( Condense, function( perms, elms, fieldsize)
local orb, orblens, gp, subgp, bound, conv, i, j, k, outfile,
      name, cj, lst, zeropos, c, ints, tmp;
gp:=Group(perms[2]{[1..perms[1]]});
subgp:=Group(perms[2]{[perms[1]+1..Length(perms[2])]});
#get the list of field elements in MeatAxe order
conv:=FFList(GF(fieldsize));
orb:=Orbits(subgp,[1..LargestMovedPoint(gp)]);
orblens:=List(orb,Length);
#find the bound on row-spareness
bound:=Minimum(Maximum(orblens),Length(orblens));
for i in [1..Length(elms)] do
#construct a sparse MeatAxe matrix
    c:=" ";
    name:=Filename( BasicWorkingSpace,
                    Concatenation( "Kdg.", String(i)));
    outfile:=OutputTextFile( name, false);
    SetPrintFormattingStatus( outfile, false);
#write the header
    PrintTo( outfile, "sparse matrix field=", String(fieldsize),
             " rows=", String(Length(orb)),
             " cols=", String(Length(orb)),
             " nonzero=", String(bound), "\n");
    for j in [1..Length(orb)] do
        cj:=[]; k:=1;
        lst:=List(orb[j], x->OnPoints(x, elms[i]));
#find a row
        while lst <> [] do
          ints:=Intersection(lst, orb[k]);
          if Length(ints) > 0 then
             cj[Length(cj)+1]:=
               [Position(conv,
               ((One(GF(fieldsize))*orblens[k])^-1)
                 *Length(ints))-1, k];
             lst:=Difference(lst, ints);
          fi;
          k:=k+1;
        od;
#fill the row to full sparse length
        zeropos:=Filtered( [1..bound+1], x->
                           ForAll(cj, y->y[2]<>x))[1];
        while Length(cj) < bound do
              Append(cj,[[ 0, zeropos]]);
        od;
        tmp:=String(Flat(cj));
        tmp:=tmp{[3..Length(tmp)-2]};
        tmp:=JoinStringsWithSeparator(SplitString(tmp,',')," ");
        Append(tmp,"\n");
#write row to string c
```

```
        Append(c, tmp);
    od;
#write out data for matrix
    AppendTo(outfile, c);
    CloseStream( outfile);
od;
end );
```

A.5 The *GAP*-procedure CondenseModuleTom

```
#
# Condense a permutation module coming from a subgroup given in
# the table of marks
#
InstallGlobalFunction(CondenseModuleTom,
function(grp,prime,makenum,condnum,alggens,subinfo)
local tom, gp, mattom, maxlist, maxl, i, lsub, vec, sub, g, kdrep,
      modrep, o, o1, infile, name, outfile, m, makestr,
      p, newg, newkdrep, newmodrep, gens, perms, permlist, l, j,
      resstrlinprg, imprim, newgenperms, options, Nrep, newpgens,
      kdrepgens, firstperm, secperm, simpsub, falsenum;
tom:=TableOfMarks(grp);
gp:=UnderlyingGroup(tom);
makestr:=StraightLineProgramsTom( tom )[makenum];
falsenum:=Length(StraightLineProgramsTom(tom));
#make the group whose perm mod is being condensed
simpsub:=Group( List( makestr, x->ResultOfStraightLineProgram( x,
               GeneratorsOfGroup(gp))));
#construct perms using atlasrep
if FindVecs( grp, condnum, falsenum, makestr, Size(simpsub),
             Flat(subinfo.gens)) > 0 then
   g:=[];
#read constructed perms
   for i in [1..Length(GeneratorsOfGroup(gp))] do
       outfile:=Filename( BasicWorkingSpace, "gapperm");
       name:=OutputTextFile(outfile, false);
       Process( BasicWorkingSpace, zpr, NoneIn, name,
                ["-G", Concatenation( "perm.", String(i))]);
       CloseStream( name);
       infile:=Filename( BasicWorkingSpace, "gapperm");
       Read(infile);
       g[i]:=MeatAxe.Perms[1];
   od;
   for i in [1..Length( subinfo.gens[2])] do
       options:=[ Concatenation("perm.", String( i +
                  Length( makestr) +
                  Length( subinfo.gens[1]) +
                  Length( GeneratorsOfGroup( gp)))),
                  Concatenation( "g.", String(i))];
       Process( BasicWorkingSpace, Cp, NoneIn, NoneOut, options);
   od;
#make the generators of eFGe in this representation
   resstrlinprg:=[];
   resstrlinprg[1]:=ResultOfStraightLineProgram(fr,g);
   gens:=[];
   firstperm:=1;
   secperm:=3;
   for i in [1..Length(alggens)] do
     if (i > 1) then
```

```
        if Length(alggens[i].genlist)>0 then
            resstrlinprg[i]:=ResultOfStraightLineProgram(fr,
                   [resstrlinprg[1][firstperm],
                   resstrlinprg[1][secperm]]);
        fi;
       secperm:=secperm+1;
       if secperm = firstperm then
          secperm:=secperm+1;
       fi;
       if secperm > 11 then
          secperm:=1;
          firstperm:=firstperm+1;
       fi;
     fi;
     for j in alggens[i].genlist do
       Append(gens,[resstrlinprg[i][j]]);
     od;
   od;
#write out gens of eFGe in this rep
   p:=LargestMovedPoint(Group(g));
   for i in [1..Length(gens)] do
       m:=MeatAxeString( [ gens[i] ], p );
       outfile:=Filename(BasicWorkingSpace, "g");
       name:=OutputTextFile( outfile, false);
       SetPrintFormattingStatus( name, false);
       PrintTo( name, m );
       CloseStream( name);
       options:=[ "g", Concatenation( "g.", String( i +
                  Length(subinfo.gens[2])) )];
       Process( BasicWorkingSpace, zcv, NoneIn, NoneOut, options);
   od;
   for i in [1..Length( subinfo.gens[1])] do
       options:=[ Concatenation("perm.", String( i +
                  Length( makestr) +
                  Length( GeneratorsOfGroup( gp)))),
                  Concatenation( "kd.", String(i))];
       Process( BasicWorkingSpace, Cp, NoneIn, NoneOut, options);
   od;
#condense the generators
   Kond( prime, "g", "kd",
         Length(subinfo.gens[1]),
         Length(gens)+Length(subinfo.gens[2]));
   for i in [1..Length(gens)+Length(subinfo.gens[2])] do
       options:=[ Concatenation( "kdg.", String(i)),
                  Concatenation( "mod.", String(i))];
       Process( BasicWorkingSpace, Mv, NoneIn, NoneOut, options);
   od;
#if atlasrep method failed, use ImprimPerm to construct rep
else
  mattom:=MatTom(tom);;
  maxlist:=FindMaximalOverGroupsTom(tom,makenum);
```

```
  maxl:=Length(maxlist[1]);
  for i in [1..maxl] do
    if maxlist[2][i] <> 1 then
      lsub:=Filtered([1..Length(mattom[maxlist[1][i]])],
          x-> mattom[maxlist[1][i]][x] <> 0);
      vec:=Filtered(lsub,x->mattom[x][1] = mattom[x][x]);
      if Length(vec) = 1 then
        break;
      fi;
    fi;
  od;
  sub:=RepresentativeTom(tom,maxlist[1][i]);
  g:=Group(List(
     ImprimPerm(gp,sub,sub,GeneratorsOfGroup(gp)),x->PermList(x)));
  Nrep:=RepresentativeTomByGeneratorsNC(tom,subinfo.Normalizer,
          GeneratorsOfGroup(g));
  newpgens:=[];
  for i in [1..Length(subinfo.gens[2])] do
    newpgens[i]:=ResultOfStraightLineProgram(subinfo.gens[2][i],
          GeneratorsOfGroup(Nrep));
  od;
  kdrepgens:=[];
  for i in [1..Length(subinfo.gens[1])] do
    kdrepgens[i]:=ResultOfStraightLineProgram(subinfo.gens[1][i],
          GeneratorsOfGroup(Nrep));
  od;
  kdrep:=Group(kdrepgens);
  modrep:=RepresentativeTomByGeneratorsNC(tom,
                makenum,GeneratorsOfGroup(g));
  o:=Orbits(modrep,[1..LargestMovedPoint(g)]);
  o1:=ShallowCopy(o);
  Sort(o1, function(a,b) return Length(a) < Length(b); end);
  p:=PermList(Flat(o1))^-1;
  newg:=Group(List(GeneratorsOfGroup(g),x->x^p));
  newkdrep:=Group(List(GeneratorsOfGroup(kdrep),x->x^p));
  newmodrep:=Group(Concatenation(List(
             GeneratorsOfGroup(modrep),x->x^p),[()]));
  newgenperms:=List(newpgens,x->x^p);
  imprim:=ImprimPerm(newg,Stabilizer(newg,1),newmodrep,
          Concatenation(GeneratorsOfGroup(newg),
          GeneratorsOfGroup(newkdrep),newgenperms));
  perms:=List(imprim,x->PermList(x));;
  resstrlinprg:=[];
  resstrlinprg[1]:=ResultOfStraightLineProgram(fr,perms{[1..
          Length(GeneratorsOfGroup(newg))]});
#write out the perms created by ImprimPerm
  gens:=[];
  firstperm:=1;
  secperm:=3;
  for i in [1..Length(alggens)] do
    if (i > 1) then
```

```
       if Length(alggens[i].genlist)>0 then
           resstrlinprg[i]:=ResultOfStraightLineProgram(fr,
                  [resstrlinprg[1][firstperm],
                  resstrlinprg[1][secperm]]);
       fi;
      secperm:=secperm+1;
      if secperm = firstperm then
         secperm:=secperm+1;
      fi;
      if secperm > 11 then
         secperm:=1;
         firstperm:=firstperm+1;
      fi;
    fi;
    for j in alggens[i].genlist do
      Append(gens,[resstrlinprg[i][j]]);
    od;
  od;
  permlist:=[Length(GeneratorsOfGroup(newg)),
          perms{[1..Length(perms)-
          Length(subinfo.gens[2])]}];
#condense the gens of eFGe
  MakeGensKondSubgroup(permlist,gens,prime);
  l:=Length(gens);
  for i in [1..l] do
    options:=[ Concatenation("kdg.",String(i)),
               Concatenation("mod.",
               String(i+Length(subinfo.gens[2])))];
    Process(BasicWorkingSpace, Mv, NoneIn, NoneOut, options);
  od;
  gens:=perms{[ Length(perms)-Length(subinfo.gens[2])+
                1..Length(perms)]};
  MakeGensKondSubgroup(permlist,gens,prime);
  for i in [1..Length(subinfo.gens[2])] do
    options:=[ Concatenation("kdg.",String(i)),
               Concatenation("mod.", String(i))];
    Process(BasicWorkingSpace, Mv, NoneIn, NoneOut, options);
  od;
fi;
end );
```

A.6 The *GAP*-procedure FindBasicHoms

```
#
# Find the homomorphisms which for a gen set for the basic alg
#   return quiver information
#
InstallGlobalFunction( FindBasicHoms,
      function(grp,gnamelist,repr,wordrepr,grpinfo)
local lnamelist,pimlist,newnamelist,cartan,i,ii,admat,m,k,a,HomNum,
      j,sn,ll,jj,l,stpimlist,ngen,jvalue,jpos,headpos,jlocal,input,
      name,head,pos,outfile,options,infile,Tempi,Tempk,HOMS,basicalg;
HOMS:=rec();
basicalg:=rec();
lnamelist:=[];
for i in [1..Length(gnamelist)] do
    if IsReadOnlyGlobal("CFInfo") = true then
       MakeReadWriteGlobal("CFInfo");
    fi;
    UnbindGlobal("CFInfo");
  infile:=Filename(BasicWorkingSpace, Concatenation("pim",
                                      gnamelist[i],".cfinfo"));
  Read(infile);
  lnamelist:=Union(lnamelist,CFInfo.ConstituentNames);
od;
#lnamelist is the local name list, names from the second chop
#(may not match gnamelist)
#create list of all pims (actual names)
pimlist:=[];
pimlist:=List(gnamelist, x->Concatenation("pim",x));
#find global peakwords
options:=Union(["-t"],pimlist);
Process(BasicWorkingSpace, pwkond, NoneIn, NoneOut, options);
#we need to reorder gnamelist so that it matches the cartan
#matrix we have from infokond
newnamelist:=ShallowCopy(gnamelist);
Sort(newnamelist);
cartan:=IdentityMat(Length(newnamelist));
for i in gnamelist do
  options:=Concatenation("pim",i);
  Process(BasicWorkingSpace, rad, NoneIn, NoneOut, [options]);
  if IsReadOnlyGlobal("CFInfo") = true then
     MakeReadWriteGlobal("CFInfo");
  fi;
  UnbindGlobal("CFInfo");
  infile:=Filename( BasicWorkingSpace, Concatenation("pim",i,
                    ".cfinfo"));
  Read(infile);
#pos is the position of the head of this pim in
#CFInfo.ConstituentNames
  pos:=Position(CFInfo.Heads[1],1);
  outfile:=Filename( BasicWorkingSpace, "res");
```

```
  name:=OutputTextFile( outfile, false);
  options:=[ wordrepr, Concatenation("pim",i,
             CFInfo.ConstituentNames[pos])];
  Process(BasicWorkingSpace, cfcomp, NoneIn, name, options);
  CloseStream(name);
  name := Filename( BasicWorkingSpace, "res" );;
  input := InputTextFile(name);;
  a:=ReadAll(input);
  CloseStream(input);
#head is the name of the head of this pim in terms of the global names
  head:=a{[(Position(a,'=')+2+Length(repr))..(Length(a)-1)]};
#headpos is the position of head in newnamelist
  headpos:=Position(newnamelist,head);
  for j in newnamelist do
    outfile:=Filename( BasicWorkingSpace, "res");
    name:=OutputTextFile( outfile, false);
    options:=[ Concatenation("pim",i), Concatenation(wordrepr,j)];
    Process(BasicWorkingSpace, cfcomp, NoneIn, name, options);
    CloseStream(name);
    name := Filename( BasicWorkingSpace, "res" );;
    input := InputTextFile(name);;
    a:=ReadAll(input);
    CloseStream(input);
    jpos:=Position(newnamelist,j);
    if Position(a,'=')=fail then
      cartan[headpos][jpos]:=0;
    else
      jlocal:=a{[(Position(a,'=')+5+Length(i))..(Length(a)-1)]};
      jvalue:=CFInfo.Multiplicity[Position(
                           CFInfo.ConstituentNames,jlocal)];
      cartan[headpos][jpos]:=jvalue;
    fi;
  od;
od;
gnamelist:=newnamelist;
#stpim's are the pims above transformed into a useful basis
stpimlist:=[];
sn:=String(CFInfo.NGen);
for i in gnamelist do
  options:=[ "-H", "1", Concatenation("pim",i)];
  Process(BasicWorkingSpace, rad, NoneIn, NoneOut, options);
  for j in lnamelist do
#identify the local pim which is the head of the global pim and use
#that info to find a basis and script for stpim
    if not (Filename( [BasicWorkingSpace],
                     Concatenation("pim",i,j,".h1")) = fail) then
       outfile:=Filename( BasicWorkingSpace,
                         Concatenation("GAPpim",i,j,".h1"));
       name:=OutputTextFile( outfile, false);
       options:=[ "-G", Concatenation("pim",i,j,".h1")];
       Process(BasicWorkingSpace, zpr, NoneIn, name, options);
```

```
      CloseStream(name);
      infile:=Filename( [BasicWorkingSpace],
                        Concatenation("GAPpim",i,j,".h1"));
      Read(infile);
      if not (MeatAxe.Matrix = []) then
         options:=[ "-g", sn, "-t", "-b", Concatenation("bas",i),
                    "-o", Concatenation("scr",i),
                    Concatenation("pim",i),
                    Concatenation("pim",i,j,".h1")];
         Process(BasicWorkingSpace, zsp, NoneIn, NoneOut, options);
      fi;
    fi;
  od;
  options:=[ Concatenation("bas",i), Concatenation("basin",i)];
  Process(BasicWorkingSpace, ziv, NoneIn, NoneOut, options);
#transform pims into new basis
  for j in [1..CFInfo.NGen] do
    options:=[ Concatenation("bas",i),
               Concatenation("pim",i,".",String(j)),
               Concatenation("left.", String(j))];
    Process(BasicWorkingSpace, zmu, NoneIn, NoneOut, options);
    options:=[ Concatenation("left.",String(j)),
               Concatenation("basin",i),
               Concatenation("stpim",i,".",String(j))];
    Process(BasicWorkingSpace, zmu, NoneIn, NoneOut, options);
  od;
  stpimlist:=Union(stpimlist, [Concatenation("stpim",i)]);
#chop the new pims so that we get a .cfinfo file to run pwkond
  options:=[ "-g", sn, Concatenation("stpim",i)];
  Process(BasicWorkingSpace, chop, NoneIn, NoneOut, options);
od;
#more global peakwords
options:=Union(["-t"],stpimlist);
Process(BasicWorkingSpace, pwkond, NoneIn, NoneOut, options);
admat:=NullMat(Length(gnamelist),Length(gnamelist));
#now we can identify the pims in the second radical layer
#to spin up the homs we want to calculate
#this is a big mess because we have to work around any
#naming inconsistencies introduced in the different chops
for ii in [1..Length(gnamelist)] do
  i:=gnamelist[ii];
  options:=[ "-H", "2", Concatenation("stpim",i)];
  Process(BasicWorkingSpace, rad, NoneIn, NoneOut, options);
  if IsReadOnlyGlobal("CFInfo") = true then
     MakeReadWriteGlobal("CFInfo");
  fi;
  UnbindGlobal("CFInfo");
  infile:=Filename( BasicWorkingSpace,
                    Concatenation("stpim",i,".cfinfo"));
  Read(infile);
  Tempi:=CFInfo;
```

```
for jj in [1..Length(Tempi.ConstituentNames)] do
  j:=Tempi.ConstituentNames[jj];
  outfile:=Filename( BasicWorkingSpace, "GAPmat");
  name:=OutputTextFile( outfile, false);
  options:=[ "-Gs", Concatenation("stpim",i,j,".h2")];
  Process(BasicWorkingSpace, zpr, NoneIn, name, options);
  CloseStream(name);
  Read(outfile);
  HomNum:=MeatAxe.MatrixRows;
  if not MeatAxe.MatrixRows = 0 then
    for k in gnamelist do
        if IsReadOnlyGlobal("CFInfo") = true then
           MakeReadWriteGlobal("CFInfo");
        fi;
        UnbindGlobal("CFInfo");
      infile:=Filename( BasicWorkingSpace,
                        Concatenation("stpim",k,".cfinfo"));
      Read(infile);
      Tempk:=CFInfo;
      for ll in [1..Length(Tempk.ConstituentNames)] do
        l:=Tempk.ConstituentNames[ll];
        infile:=Filename( BasicWorkingSpace,
                          Concatenation("GAPpim",k,l,".h1"));
        Read(infile);
        if not (MeatAxe.Matrix = [])
          and Tempi.PeakWord[jj]=Tempk.PeakWord[ll] then
          options:=[ "-g", sn, Concatenation("stpim",i),
                     Concatenation("stpim",i,j,".h2"),
                     Concatenation("scr",k),
                     Concatenation("hom",i,k)];
          Process(BasicWorkingSpace, zsc, NoneIn, NoneOut, options);
          for m in [1..HomNum] do
              outfile:=Filename( BasicWorkingSpace, "GAPHm");
              name:=OutputTextFile( outfile, false);
            options:=[ "-G", Concatenation("hom",i,k,".",
                       String(m))];
            Process(BasicWorkingSpace, zpr, NoneIn, name, options);
            CloseStream(name);
            infile:=Filename(BasicWorkingSpace, "GAPHm");
            Read(infile);
            admat[Position(gnamelist,k)][Position(gnamelist,i)]:=
              admat[Position(gnamelist,k)][Position(gnamelist,i)]+1;
            a:=Concatenation(i,k,String(m));
            HOMS.(a):=rec(
              start:=Position(gnamelist,k),
              ende:=ii,
              name:=Concatenation(i,k,String(m)),
              mat:=MeatAxe.Matrix);
          od;
        fi;
      od;
```

```
      od;
    fi;
  od;
od;
basicalg.group:=grp;
basicalg.generators:=[];
basicalg.npims:=Length(gnamelist);
basicalg.pimnames:=gnamelist;
basicalg.cartan:=cartan;
basicalg.field:=GF(grpinfo.splitf);
basicalg.dim:=[];
for i in gnamelist do
  outfile:=Filename( BasicWorkingSpace, "GAPstpim");
  name:=OutputTextFile( outfile, false);
  options:=[ "-G", Concatenation("stpim",i,".1")];
  Process(BasicWorkingSpace, zpr, NoneIn, name, options);
  CloseStream(name);
  infile:=Filename(BasicWorkingSpace, "GAPstpim");
  Read(infile);
  Append(basicalg.dim,[Length(MeatAxe.Matrix)]);
od;
basicalg.adjmat:=admat;
for ii in [1..Length(gnamelist)] do
  i:=gnamelist[ii];
  basicalg.(i):=rec();
  basicalg.(i).start:=ii;
  basicalg.(i).ende:=ii;
  basicalg.(i).name:=Concatenation("id",i);
  basicalg.(i).mat:=IdentityMat(basicalg.dim[ii],basicalg.field);
  Append(basicalg.generators,[i]);
od;
for i in RecNames(HOMS) do
  basicalg.(i):=HOMS.(i);
od;
Append(basicalg.generators,RecNames(HOMS));
outfile:=Filename(BasicWorkingSpace, "Input.gap");
PrintTo(outfile,"basicalg:=");
AppendTo(outfile,basicalg);
AppendTo(outfile,";");
return(basicalg);
end);
```

References

[AF92] Frank W. Anderson and Kent R. Fuller, *Rings and categories of modules*, second ed., Graduate Texts in Mathematics, vol. 13, Springer-Verlag, New York, 1992. MR 94i:16001

[Ben83a] D. Benson, *The Loewy structure of the projective indecomposable modules for A_8 in characteristic* 2, Comm. Algebra **11** (1983), no. 13, 1395–1432. MR 84k:20006a

[Ben83b] D. J. Benson, *The Loewy structure of the projective indecomposable modules for A_9 in characteristic* 2, Comm. Algebra **11** (1983), no. 13, 1433–1453. MR 84k:20006b

[Ben98] ———, *Representations and cohomology. I*, Cambridge Studies in Advanced Mathematics, vol. 30, Cambridge University Press, Cambridge, 1998, Basic representation theory of finite groups and associative algebras. MR 92m:20005

[Car96] Jon F. Carlson, *Modules and group algebras*, Lectures in Mathematics ETH Zürich, Birkhäuser Verlag, Basel, 1996, Notes by Ruedi Suter. MR 97c:20013

[CR62] Charles W. Curtis and Irving Reiner, *Representation theory of finite groups and associative algebras*, Pure and Applied Mathematics, Vol. XI, Interscience Publishers, a division of John Wiley & Sons, New York-London, 1962. MR 26 #2519

[CR81] ———, *Methods of representation theory. Vol. I*, Wiley Classics Library, John Wiley & Sons Inc., New York, 1981, With applications to finite groups and orders, A Wiley-Interscience Publication. MR 90k:20001

[Erd90] Karin Erdmann, *Blocks of tame representation type and related algebras*, Lecture Notes in Mathematics, vol. 1428, Springer-Verlag, Berlin, 1990. MR 91c:20016

[Fei82] Walter Feit, *The representation theory of finite groups*, North-Holland Mathematical Library, vol. 25, North-Holland Publishing Co., Amsterdam, 1982. MR 83g:20001

[GAP02] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.3*, 2002, (`http://www.gap-system.org`).

[Gro83] Larry C. Grove, *Algebra*, Pure and Applied Mathematics, vol. 110, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1983. MR 86j:00002

[HL89] G. Hiss and K. Lux, *Brauer trees of sporadic groups*, Oxford Science Publications, The Clarendon Press Oxford University Press, New York, 1989. MR 91k:20018

[HS97] P. J. Hilton and U. Stammbach, *A course in homological algebra*, second ed., Graduate Texts in Mathematics, vol. 4, Springer-Verlag, New York, 1997. MR 97k:18001

[Hun87] Thomas W. Hungerford, *Algebra*, Graduate Texts in Mathematics, vol. 73, Springer-Verlag, New York, 1987, Reprint of the 1974 original. MR 82a:00006

[Isa76] I. Martin Isaacs, *Character theory of finite groups*, Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976, Pure and Applied Mathematics, No. 69. MR 57 #417

[Jan66] Zvonimir Janko, *A new finite simple group with abelian Sylow* 2*-subgroups and its characterization*, J. Algebra **3** (1966), 147–186. MR 33 #1359

[JLPW95] Christoph Jansen, Klaus Lux, Richard Parker, and Robert Wilson, *An atlas of Brauer characters*, London Mathematical Society Monographs. New Series, vol. 11, The Clarendon Press Oxford University Press, New York, 1995, Appendix 2 by T. Breuer and S. Norton, Oxford Science Publications. MR 96k:20016

[Lux97] Klaus Lux, *Algorithmic methods in modular representation theory*, Habilitation thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, 1997.

[LW01] Klaus Lux and Markus Wiegelmann, *Determination of socle series using the condensation method*, J. Symbolic Comput. **31** (2001), no. 1-2, 163–178, Computational algebra and number theory (Milwaukee, WI, 1996). MR 2001m:20022

[Mar89] Stuart Martin, *On the ordinary quiver of the principal block of certain symmetric groups*, Quart. J. Math. Oxford Ser. (2) **40** (1989), no. 158, 209–223. MR 90d:20032

[Mar90] _____, *Ordinary quivers for symmetric groups. II*, Quart. J. Math. Oxford Ser. (2) **41** (1990), no. 161, 79–92. MR 91d:20018

[MR96] Stuart Martin and Lee Russell, *On the Ext-quiver of blocks of defect* 3 *of symmetric group algebras*, J. Algebra **185** (1996), no. 2, 440–480. MR 97j:20014

[Pah03] H. Pahlings, *Representations of groups – theory and practice*, Draft, 215 pages, July 2003.

[Pfe97] Götz Pfeiffer, *The subgroups of* M_{24}*, or how to compute the table of marks of a finite group*, Experiment. Math. **6** (1997), no. 3, 247–270. MR 98h:20032

[Rin00] Micheal Ringe, *The C-MeatAxe, release 2.4.0*, (`http://www.math.rwth-aachen.de/homes/MTX/`), 2000.

[Ryb90] A. J. E. Ryba, *Computer condensation of modular representations*, J. Symbolic Comput. **9** (1990), no. 5-6, 591–600, Computational group theory, Part 1. MR 91j:20002

[Ser96] Jean-Pierre Serre, *Linear representations of finite groups*, Springer-Verlag, New York, 1996, Translated from the second French edition by Leonard L. Scott, Graduate Texts in Mathematics, Vol. 42. MR 56 #8675

[Sin96] Peter Sin, *Modular representations of the Hall-Janko group*, Comm. Algebra **24** (1996), no. 14, 4513–4547. MR 97j:20016

[Sző98] Magdolna Szőke, *Examining Green correspondents of weight modules*, Ph.D. thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, 1998.

[Tha81] J. G. Thackray, *Modular representations of some finite groups*, Ph.D. thesis, University of Cambridge, 1981.

[W+04] Robert Wilson et al., *Atlas of finite group representations*, (`http://web.mat.bham.ac.uk/atlas/v2.0/`), 2004.