The Cohomology Rings of Classical Brauer Tree Algebras

by Lee A. Chasen

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mathematics

APPROVED:

Edward Green, Committee Chairman
Mathematics Department

Daniel Farkas
Mathematics Department

Peter Fletcher
Mathematics Department

Robert McCoy
Mathematics Department

Joseph Ball
Mathematics Department

July 3, 1995
Blacksburg, Virginia

Key Words: Mathematics, Algebra, Ring Theory, Module Theory, Homological Algebra

The Cohomology Rings of Brauer Tree Algebras

by

Lee Andrew Chasen

Committee Chairman: Edward Green
Mathematics

Abstract

In this dissertation a simple algorithm is given for calculating minimal projective resolutions of nonprojective indecomposable modules over Brauer tree algebras. Those calculated resolutions lead to an algorithm for calculating a minimal set of generators for the cohomology ring of a Brauer tree algebra.

With All My Love

This Thesis is Dedicated to

My Wife

My Family

And My Friends

Acknowledgments

I would like to thank Dr. Edward Green, my advisor, for all of his help. I am especially thankful for the patience and good humor that he demonstrated while I was his student. I would also like to thank Dr. Ball, Dr. Farkas, Dr. Fletcher, and Dr. McCoy for serving on my committee.

I would also like to recognize all of the other people who helped me reach my goal. There are many that I will not forget. But here, I name only a few. Amy Chasen, my wife, ensured that this dissertation was actually completed, and completed on time. Brian Hager always had faith in me and encouraged me frequently. Dan Eno ligtened my days. James Lynch taught me a great deal about intuition and understanding. I thank them all.

Contents

List of Figures

Chapter 1

Introduction

Let G be a finite group, K a splitting field of characteristic $p \neq 0$, and B a p-block of KG having a cyclic defect group where KG is the group ring. We note that a block is an indecomposable direct factor of KG. In 1941 in [2] Richard Brauer began using trees, or planar nondirected graphs with n edges and $n+1$ vertices, as a notational tool for describing the various characters, modules, and other objects associated with B.

In [12] D. Higman showed that if K is a field of characteristic p which divides the order of a group G, then KG is of finite representation type if and only if the p-Sylow subgroups of G are cyclic. Here finite representation type means that there are only a finite number of nonisomorphic indecomposable modules over KG. Later Kasch, Kneser, and Kupisch [14] and G.Janusz [13] provide detailed information about the number of nonisomorphic indecomposable modules in this case.

In [5] E. Dade describes the character theory developed by Brauer for B when B has a cyclic defect group. This includes the case where G contains a cyclic p-Sylow subgroup. It is in that case that G. Janusz in [13] used Dade's results to construct the complete set of nonisomorphic indecomposable modules over B, a two sided ideal direct summand of KG. He also showed how to build the multiplication table for B from the Brauer tree and the character theory for G. Furthermore, he proved that once the multiplication table is known, the indecomposable modules can be determined without knowing the character theory.

A few years later, in [1] J. Alperin and G. Janusz began considering resolutions and periodicity. In particular, they proved that if G is a finite group with a cyclic p-Sylow subgroup for a prime p and F is a splitting field of characteristic p for G, then each term of the minimal projective resolution of F over FG is indecomposable. Actually they constructed the resolution and showed that it is periodic of period $2n$ where n is the number of edges in the Brauer tree corresponding to B the principal block of FG. From the construction of that resolution, they also obtained minimal projective resolutions for a few other indecomposable modules. In particular they obtained minimal projective resolutions for the socle of the projective indecomposables $v\Lambda$ where v is a looped vertex of a Brauer quiver and Λ is the associated Brauer tree algebra both of which we define in Chapter 2. For a definition of a looped vertex see section 5.1. We note that at about the same time J.A. Green [11] obtained similar results.

About a year later P. Gabriel [6] summarized many of the preceding results using, for the first time, a Brauer quiver, Γ, and the category rep(Γ, ρ) which I also define in chapter 2. He credits Dade and Janusz with proving that if G is a finite group and B is a block of the group algebra KG, where K is an algebraically closed field of characteristic $p > 0$, then if B has only finitely many indecomposable B-modules, then B is either semi-simple, or the category of B-modules is equivalent to the category rep(Γ, ρ) for some Brauer quiver Γ and some exceptional cycle. Thus in this paper we will focus on the category rep(Γ, ρ) which is Morita equivalent the category of modules over the Brauer tree algebra corresponding to Γ. These ring became known as Brauer tree algebras and are our focus in this thesis. We refer to them as Classical Brauer trees in the title since in recent years the term Brauer tree algebra has occasionally taken on different meanings.

Gabriel goes on to credit Janusz and Kupisch with showing that all of the indecomposable representations for B in that case correspond to the set of string modules available for that Brauer quiver. We define those modules in Chapter 3.

Continuing to use the quiver approach M.C.R. Butler and C.M. Ringel [4] generalize the quiver approach. They defines a large class of algebras called string algebras. This class of algebras contains the Brauer tree algebras. They also gives a complete description of the indecomposable modules over string algebras. As an immediate consequence of their results we have that the set of string modules for the Brauer tree algebras is a complete set of nonprojective indecomposable modules for the Brauer tree algebra. So in studying the string modules over a Brauer tree algebra we will be studying the nonprojective indecomposable modules over the Brauer tree algebra.

In this thesis we will extend the resolution results of Alperin and Janusz. In particular we will give a simple constructive algorithm for computing the minimal projective resolutions for all indecomposable modules over a Brauer tree algebra. We will show that the periodicity is either n or $2n$ where n is the number of vertices in the Brauer quiver. We will use those results to compute a minimal set of generators for the cohomology ring of the Brauer tree algebra with respect to the Jacobson radical. We define the Jacobson radical and the cohomology ring in chapters 5 and 6 respectively. We note that in [3] P. Brown, using different techniques, determined these minimal generating sets less algorithmically for a larger class of algebras, namely representation-finite biserial algebras.

The basic layout for this thesis is as follows: In Chapter 5 we give an algorithm for constructing minimal projective resolutions of string modules over a Brauer tree algebra with exceptional number $N \geq 1$. In Chapter 6 we use those resolutions to compute a minimal set of generators for the cohomology ring of a Brauer tree algebra with exceptional number N. On some occasions we compute minimal projective resolutions for an indecomposable module over a Brauer tree algebra with exceptional number $N \geq 1$ by computing a minimal projective resolution for an indecomposable module over a Brauer tree algebra with exceptional number 1. We justify this technique using covering theory which was developed in [7] and [8] which we present in section 3.2. In Chapter 2 we define both path algebras and Brauer tree algebras which are quotients of certain path algebras.

In [15] H.Meltzer and A.Skowroński gave a short time line describing some of the events above and it was instrumental in the organization of this introduction. Although not explicitly referred to, we list the related works [10], [17], and [16].

Chapter 2

Preliminaries

2.1 Path Algebras

We recall the definition of a path algebra. By a *quiver path* or *directed path* P we will mean a finite sequence of arrows $a_1 a_2 \cdots a_r$ such that $t(a_i) = o(a_{i+1})$, where r is a positive integer and $1 \leq i \leq r$. We will frequently leave off the word 'quiver' when referring to a quiver path. We also define $t(P) =$ the terminus of the path P, and $o(P) =$ the origin of the path P. The K-basis for $K\Gamma$ is the set of all vertices (i.e. paths of length zero) together with the set of all finite paths. We define the multiplication on basis elements and extend linearly. Recall that we are thinking of vertices as paths of length zero. Let P and Q be paths. Then

$$P \cdot Q = \begin{cases} PQ & \text{if } t(P) = o(Q) \\ 0 & \text{if } t(P) \neq o(Q). \end{cases}$$

2.2 Brauer Tree Algebras

We define a Brauer tree algebra, Λ, as a quotient of a path algebra, ${}^{K\Gamma}/_{I}$. In this section we describe the quiver Γ that is associated with the Brauer tree and give a set of relations (or generators) which generate the ideal I. We will illustrate the construction with an example.

A *Brauer tree* is a finite connected undirected tree that has been placed in a plane with a clockwise orientation. The tree also has one exceptional node, i.e., one node that has a positive integer (the exceptional number) associated with it. In the diagrams we will place the exceptional number close to the exceptional node. Using the (underlying) Brauer tree we construct a quiver. For our example we use the underlying Brauer tree in figure 2.1.

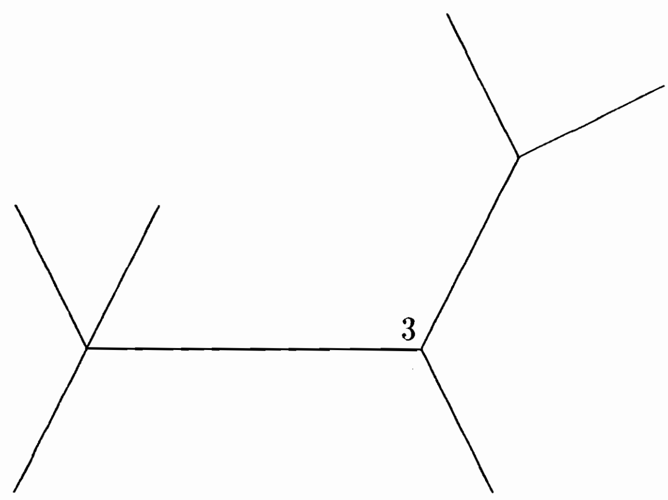


Figure 2.1:

As in figure 2.2 we label each edge (approximately at the midpoint), $v_1, v_2, \ldots, v_n$. These will be the vertices of the directed quiver Γ that we will build.

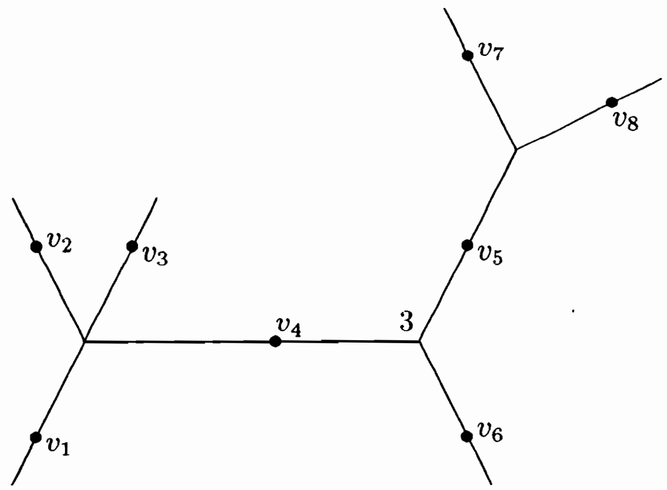


Figure 2.2:

We enclose each node within a cycle of directed segments (arrows) which connect the midpoints (vertices) of the edges that have that node as an endpoint. These directed segments (which are the arrows of Γ) must follow the orientation of the plane. Label the arrows that have just been created. See figure 2.3.

From now on when we use the word *cycle* we will mean one of these cycles of arrows surrounding a node. We will call the cycle surrounding the exceptional node the *exceptional cycle*. And in diagrams we will place the exceptional number in the center of the exceptional cycle.

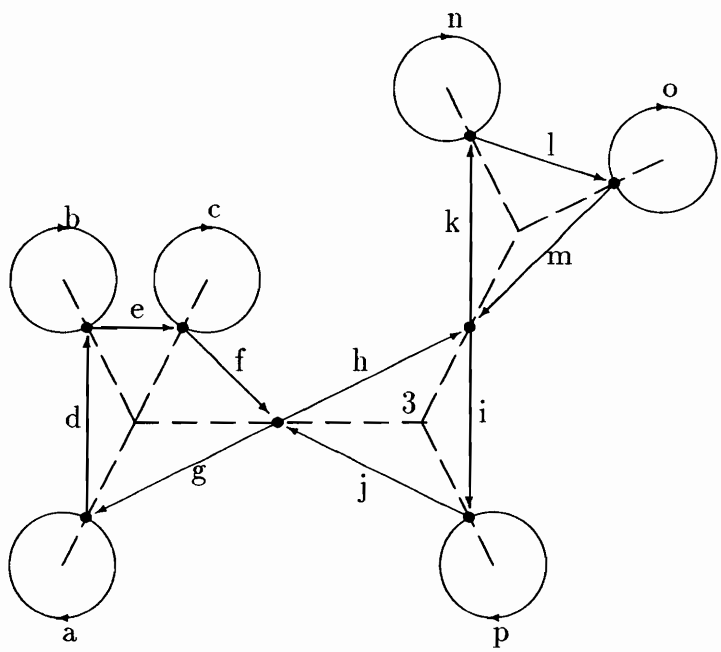


Figure 2.3:

Thus, given a Brauer tree, we have constructed the quiver Γ that we will associate to that tree. To present the relations for the ideal I, the set of which we will call ρ, we need some terminology. A path, P, will be called a *maximal path* if a) P lies entirely within a cycle and b) if that cycle is not the exceptional cycle, the length of P is the number of arrows in the cycle and c) if that cycle is the exceptional cycle, length of P is $N \cdot$ (the number of arrows that are in the exceptional cycle) where N is the exceptional number.

There are three types of relations that generate I.

Type A: Any path of length two that doesn't lie entirely within one cycle.

Type B Any path lying entirely within a cycle that contains a maximal path and is one arrow longer than that maximal path.

Type C The difference of any two maximal paths that have the same origin (and hence the same terminus).

For the example that we have been working with

$I = < ad, db, be, ec, cf, fh, hk, kn, nl, lo, om, mi, ip, pj, jg, aa, bb, cc, gdefg, defgd,$
$efgde, fgdef, hijhijhijh, ijhijhijhi, jhijhijhij, nn, oo, pp, klmk, lmkl,\ \ mklm, a -$
$defg, b - efgd,\ c - fgde, hijhijhij - gdef, p - jhijhijhi,$
$ijhijhijh - klm,\ n - lmk, o - mkl >$

2.3 rep$(QUIV, \rho)$

For the definitions we follow [7]. We let $QUIV$ be a directed graph and we let ρ be a set of generators of an ideal I in $K\,QUIV$. Also let $QUIV_0$ be the set of vertices in $QUIV$, and $QUIV^1$ be the set of arrows in $QUIV$. A K-representation of $QUIV$ is a tuple $X = (X_v, \alpha(a))_{v \in QUIV_0, a \in QUIV_1}$, where each X_v is a finite-dimensional K vector space, all but a finite number zero, and if a is an arrow from vertex v to vertex w, then $\alpha(a)$ is a K-linear map from X_v to X_w. We will write the maps $\alpha(a)$ on the right. The *category of K-representations of $QUIV$*, denoted rep$(QUIV)$, has as objects the K-representations of $QUIV$ and as morphisms $B : (X_v, \alpha(a)) \to (Y_v, \beta(a))$ tuples $B = (B_v)_{v \in QUIV_0}$ such that each B_v is a K-linear map from X_v to Y_v such that if a is an arrow from vertex v to vertex w then $B_w \circ \alpha(a) = \beta(a) \circ B_v$.

We will say that the K-representation X satisfies the relations ρ if whenever $\coprod_{i=1}^{k} a_{i1}a_{i2}\cdots a_{in_i} \in \rho$, we have $\coprod_{i=1}^{k} \alpha(a_{i1})\alpha(a_{i2})\cdots\alpha(a_{in_i}) = 0$. We let rep$(QUIV, \rho)$ denote the full subcategory of K-representations of $QUIV$ satisfying the relations ρ.

Chapter 3

String Modules and Covering Theory

3.1 Strings and String Modules

Given a quiver, $QUIV$, which is either Γ or Γ^* and a set of relations rel, we define the *formal inverse* of an arrow a to be a^{-1} and define $o(a^{-1}) = t(a)$ and $t(a^{-1}) = o(a)$. We also define $(a^{-1})^{-1} = a$. A *formal path*, $a_1 \cdots a_n$, is a sequence where for $1 \leq i < n$, a_i is of the form a or a^{-1} and $t(a_i) = o(a_{i+1})$. We define $(a_1 \cdots a_n)^{-1} = a_n^{-1} \cdots a_1^{-1}$, $t(a_1 \cdots a_n) = t(a_n)$, and $o(a_1 \cdots a_n) = o(a_1)$.

A *string* is a formal path $a_1 \cdots a_n$ where $a_{i+1} \neq a_i^{-1}$ for all $1 \leq i < n$, and no subpath nor its inverse is a maximal path, no subpath nor its inverse is in rel. We also define strings of length zero. For any vertex, v, of the quiver define 1_v to be a string of length zero. We define and always reduce

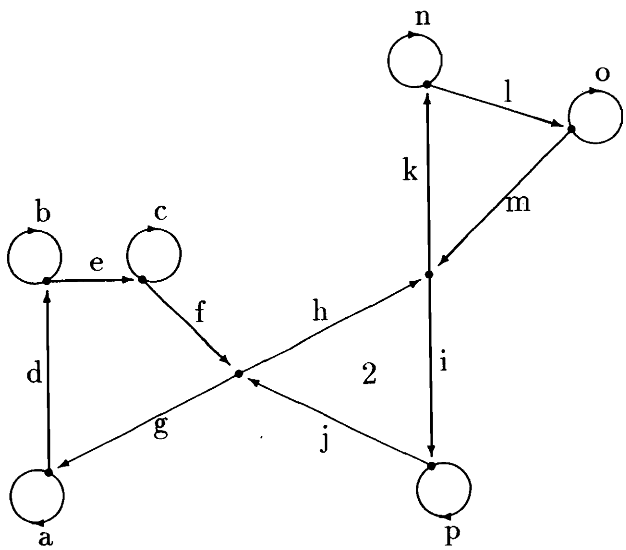
$$1_v a = \begin{cases} a & \text{if } o(a) = v \\ \text{undefined} & \text{otherwise} \end{cases} \quad \text{and} \quad a 1_v = \begin{cases} a & \text{if } t(a) = v \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Following [4], given a string of length greater than zero $S = a_1 \cdots a_n$ we define a *string module* $M(S)$. We define it by giving its representation in $\text{rep}(QUIV, \rho)$. Let $u(i) = o(a_{i+1})$ for $0 \leq i < n$ and $u(n) = t(a_n)$. Then if v is any vertex in $QUIV$, we let $I_v = \{i | u(i) = v\}$. $M(S)_v$ will be a vector space of dimension the cardinality of I_v with basis vectors $z_i, i \in I_v$. If $a_i = a$, define $a(z_{i-1}) = z_i$, and if $a_i = a^{-1}$, then define $a(z_i) = z_{i-1}$. If $\gamma : w \rightarrow w'$ is an arrow and z_j is one of the basis vectors of $M(S)$ and $\gamma(z_j)$ has not yet been defined , let $\gamma(z_j) = 0$. Note that $M(S)$ and $M(S^{-1})$ are isomorphic.

If $S = 1_v$ for some vertex in v then we define $M(1_v)$ to be the simple representation corresponding to the vertex v in $QUIV$. In other words $M(S)_v$ is a one dimensional vector space, $M(S)_u$ is a zero dimensional vector space for $u \neq v$, and all maps are the zero map.

In figure 3.1 we give a quiver Γ. Letting the relations be the Brauer tree relations, we give an example of a string S and the representation for $M(S)$. In our illustration of the representation we will write v when we are refering to a one dimensional subspace of $M(S)_v$. To distinguish two different one dimensional subspaces of $M(S)_v$ we would write v once and $v\prime$ another time. We have drawn the nonzero maps which we note always map one dimensional subspaces to one dimensional subspaces.

Γ:



For $S = k^{-1}(ij)(efgdef)^{-1}$
$\quad = k^{-1}ijf^{-1}e^{-1}d^{-1}g^{-1}f^{-1}e^{-1}$,
$M(S) =$

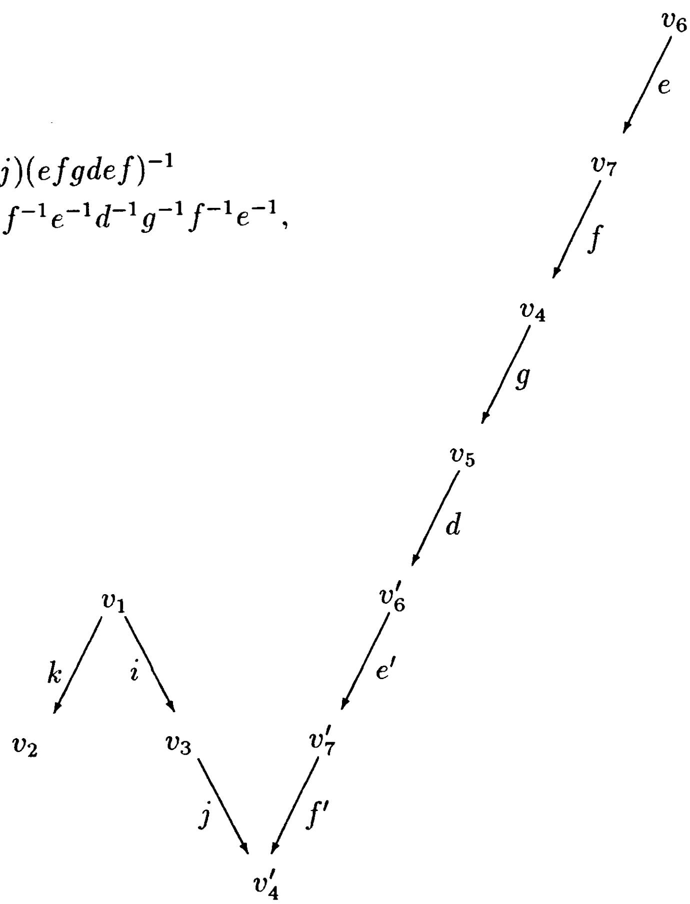


Figure 3.1:

These modules are nice to work with since up to isomorphism they are determined by only a few pieces of information. We will show that this is the case where the exceptional number is 1 by showing that string functions can be determined using only a few pieces of information. We introduce string functions in Chapter 4 and show that every string module is isomorphic to the image of some string function. But more than just another pretty module, Butler and Ringel [4] prove

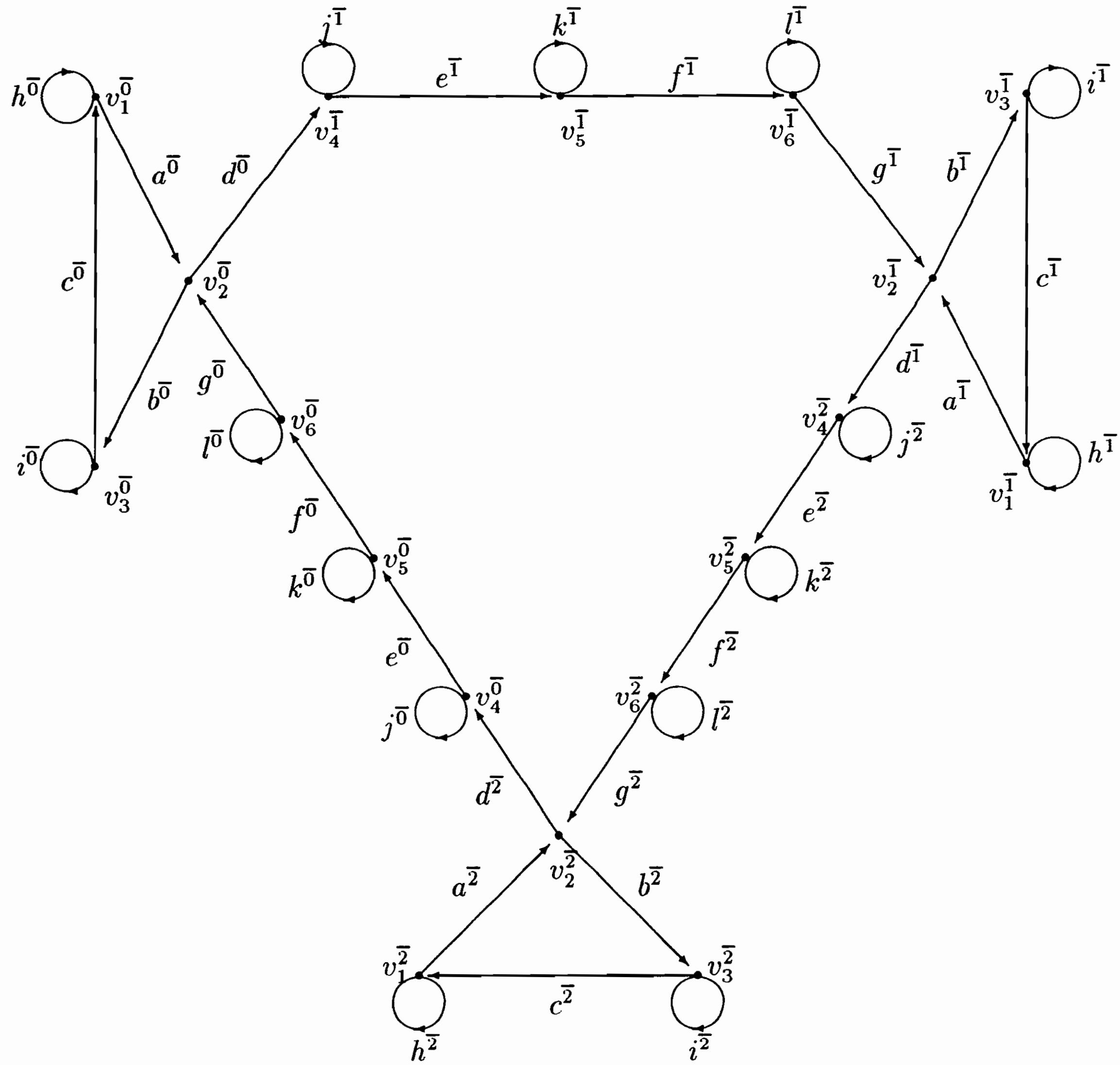
Theorem 3.1 *$\{M(S)\ :\ S$ is a string$\}$ is a complete set of the nonisomorphic nonprojective indecomposable modules over Λ.*

Therefore if we find the minimal projective resolutions for all string modules over Λ we will have found all of the minimal projective resolutions for the nonprojective indecomposable modules over Λ.

3.2 Coverings

The covering theory that follows was developed by E.L.Green and can be found in [7] and [8]. Let Γ be the quiver for a Brauer tree algebra with exceptional number N. We will construct a second Brauer tree, Γ, and its associated Brauer tree algebra, Λ^*, with exceptional number 1. The theory works more generally but we only concentrate on Brauer tree algebras. We define Γ_0 to be the set of vertices of Γ, and Γ_1 be the set of arrows of Γ. Let W be an arbitrary set function from Γ_1 to a group G. For us G will be the group of integers modulo N which we denote $\boldsymbol{Z}_N$.

A regular covering of Γ, Γ^*, is constructed in [7]. We give review that construction below. The quiver is given as follows: $\Gamma_0^* = \{v^g | v \in \Gamma_0$, and $g \in G\}$ $\Gamma_1^* = \{a^g | a \in \Gamma_1$, and $g \in G\}$ where if $v \xrightarrow{a} w$ in Γ then $v^g \xrightarrow{a^g} w^{gW(a)}$. Since Γ and Γ^* are locally homeomorphic it is clear that Γ^* is a quiver associated to some Brauer



Γ^*

Figure 3.3:

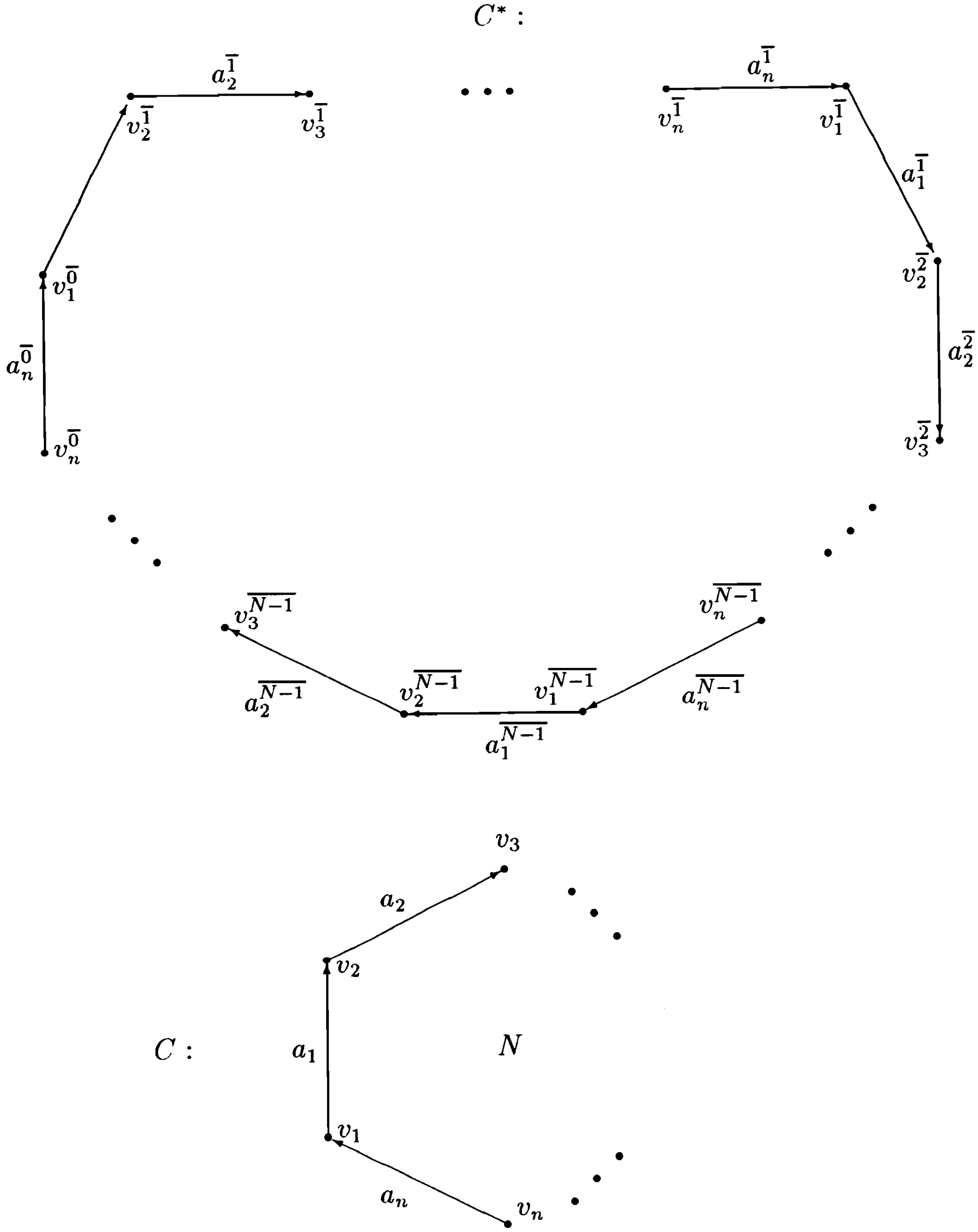
Theorem 3.2 *Λ^* is the Brauer tree algebra with exceptional number 1 associated to the quiver Γ^*.*

Proof. Since Γ and Γ^* are locally homeomorphic it is clear that Γ^* is a quiver associated to some Brauer tree. Therefore we only need to show that ρ^* is precisely the set of relations of types A,B, and C that define the relations for the Brauer tree algebra.

We first show that ρ^* contains all of the relations of type A,B, and C. Since the quivers are locally homeomorphic it is clear that ρ^* contains the relations of type A, products of arrows not lying in the same cycle.

To see that all of the relations of types B and C are contained in ρ^* we first show that each maximal path in Γ^* lifts from a maximal path in Γ. Note that each nonexceptional cycle in Γ^* lifts from a homeomorphic nonexceptional cycle in Γ. Thus it is clear that each maximal path that lies in a nonexceptional cycle in Γ^* lifts from a maximal path in Γ.

The exceptional cycle C with exceptional number N in Γ lifts to an exceptional cycle C^* with exceptional number 1 in Γ^*. These cycles are not homeomorphic. In fact C^* has N times the number of arrows that appear in C. We illustrate this in figure 3.4. Where we are letting C be the cycle with arrows $a_1, .., a_n$, and $W(a_1) = \overline{1}$. We construct C^*.



C^* :
$a_2^{\overline{1}}$
$v_2^{\overline{1}}$
$v_3^{\overline{1}}$
$a_n^{\overline{1}}$
$v_n^{\overline{1}}$
$v_1^{\overline{1}}$
$a_1^{\overline{1}}$
$v_1^{\overline{0}}$
$a_n^{\overline{0}}$
$v_n^{\overline{0}}$
$v_2^{\overline{2}}$
$a_2^{\overline{2}}$
$v_3^{\overline{2}}$
$v_3^{\overline{N-1}}$
$v_n^{\overline{N-1}}$
$a_2^{\overline{N-1}}$
$v_2^{\overline{N-1}}$
$v_1^{\overline{N-1}}$
$a_n^{\overline{N-1}}$
$a_1^{\overline{N-1}}$
v_3
a_2
v_2
C :
a_1
N
v_1
a_n
v_n

Figure 3.4:

Note that the maximal path in C^*,

$$(a_1^{\overline{0}}a_2^{\overline{1}}\dots a_n^{\overline{1}})(a_1^{\overline{1}}a_2^{\overline{2}}\dots a_n^{\overline{2}})\cdots(a_1^{\overline{N-2}}a_2^{\overline{N-1}}\dots a_n^{\overline{N-1}})(a_1^{\overline{N-1}}a_2^{\overline{0}}\dots a_n^{\overline{0}}),$$

projects to the maximal path in C

$$\underbrace{(a_1a_2\dots a_n)(a_1a_2\dots a_n)\cdots(a_1a_2\dots a_n)(a_1a_2\dots a_n).}_{N\text{times}}$$

Similarly we see that each maximal path in C^* lifts from a maximal path in C.

Now that we have every maximal path in C^* lifts from a maximal path in C, we can easily see that all of the relations of types B and C are contained in ρ^*. This is because they are all either an arrow times a maximal path, or the difference of two maximal paths. It is also now clear that ρ^* only contains relations of type A, B, and C. ■

We will let Mod(Λ) and Mod(Λ^*) be the categories of modules over Λ and Λ^* respectively. We let rep(Γ,ρ) be the category of K-representations of Γ satisfying the relations ρ that generate I, which we defined in Section 2.3. In a similar fashion we define rep(Γ^*,ρ) where ρ^* are the relations that generate I^*. We let F and J be the standard functors: F : rep$(\Gamma,\rho)\to$ Mod(Λ) and J : Mod(Λ^*) $\to$ rep(Γ^*,ρ^*). See [7]

Let X be a tuple in rep(Γ^*,ρ^*) given by $(X_{v^t},\alpha(a^r))_{v\in\Gamma_0^*,t,r\in G,a\in\Gamma_1^*}$ where $\alpha(a)$ is a linear map from the vector space $X_{o(a)}$ to $X_{t(a)}$. Then we define H : rep$(\Gamma^*,\rho^*)\to$ rep(Γ,ρ) to be the functor given by $H((X))=(Y_v,\beta(a))$ where $Y_v=\coprod_{t\in G}X_{v^t}$ and $\beta(a)=\sum_{r\in G}\alpha(a^r)$. It was shown in [9] that since Λ and Λ^* are finite dimensional over K, H is exact and additive. We define the functor E to be the composition of functors, $J\circ H\circ F$. Gordon and Green also showed that E maps simple Λ^*-modules (respectively Λ^*-projective covers) to simple Λ^*-modules (respectively Λ-projective covers). In addition they showed that if $\underline{r}$ and $\underline{r}^*$ are the Jacobson radicals for Λ and Λ^* respectively, then $E(\underline{r}^*M) = \underline{r}E(M)$ for any Λ^*-module. They proved these results in the case where $G = \mathbf{Z}$, but the proofs remain the same in the case where $G = \mathbf{Z_N}$. Thus E maps Λ^*-minimal projective resolutions to Λ-minimal projective resolutions.

We add to the results above by by showing that E maps indecomposable Λ^*-modules to indecomposable Λ-modules. In other words we will show that E maps Λ^*-string modules to Λ-string modules. More than that we will show that given a Λ-string module, M, there is a Λ^*-string module M^* such that $E(M^*) = M$. In Proposition 3.3 we will construct M^*. However, we begin with an illustration (see section 3.1 for the notation) of the action of E by drawing part of the representation for a Λ^*-string module R and the corresponding part of the representation for the Λ-string module $E(R)$. That R^* is a string is an immediate consequence of the fact that maximal paths and relations in Λ lift to maximal paths and relations in Λ^*. Later when we discuss the homomorphisms that appear in the minimal projective resolutions we will also describe the image of the homomorphism under E. In this way any constructive algorithms that we develop for computing the minimal projective resolutions of Λ^*-string modules will automatically become a constructive algorithms for computing minimal projective resolutions of Λ-string modules.

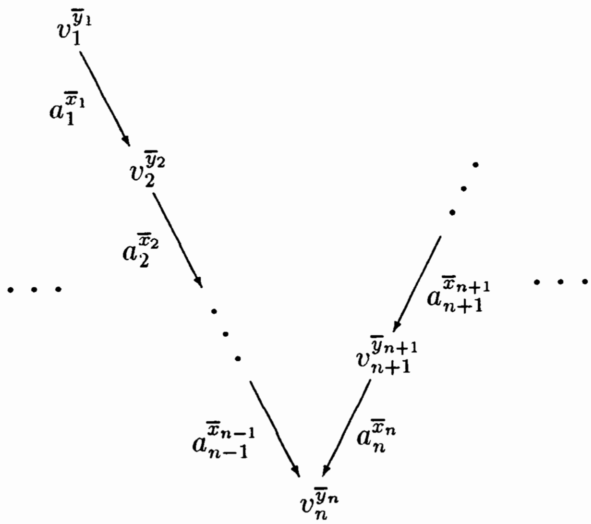
Proposition 3.3 *Given a Γ-string $S = s_1 \cdots s_n$ where the s_i are quiver and formal arrows, there exists a Γ^*-string S^* such that $E(M(S^*)) = M(S)$.*

Proof. We make the following definition: if $s = p^{-1}$, where p is a quiver arrow, and $\bar{l} \in Z_N$, we define $s^{\bar{l}} = (p^{\bar{l}})^{-1}$. We let a be the arrow in Γ such that $W(a) = \bar{1}$.

Then, if for each i, $w(s_i) = \overline{0}$, then we define $S^* := s_1^{\overline{0}} \cdots s_n^{\overline{0}}$. If $\{s_{t_1}, \ldots, s_{t_k}\}$ is the set of all arrows in S^* such that either $s_{t_i} = a$ or a^{-1}, then we define $S^* = s_1^{\overline{0}} \cdots s_{t_1}^{\overline{0}} s_{t_1+1}^{\overline{1}} \cdots s_{t_2}^{\overline{1}} s_{t_2+1}^{\overline{2}} \cdots s_k^{\overline{k-1}} s_{t_k+1}^{\overline{k}} \cdots s_n^{\overline{k}}$. In either case $E(M(S^*)) = M(S)$.

Remark: We could have begun the construction of S^* with $s_1^{\overline{j}}$ instead of $s_1^{\overline{0}}$ and changed the proof accordingly. We also could have used the inverses of any of the strings that we construct in this fashion. However, those are all of the strings S^* such that $E(M(S^*)) = M(S)$. ■

$J(M(R^*))$:



$H(J(M(R^*)))$:

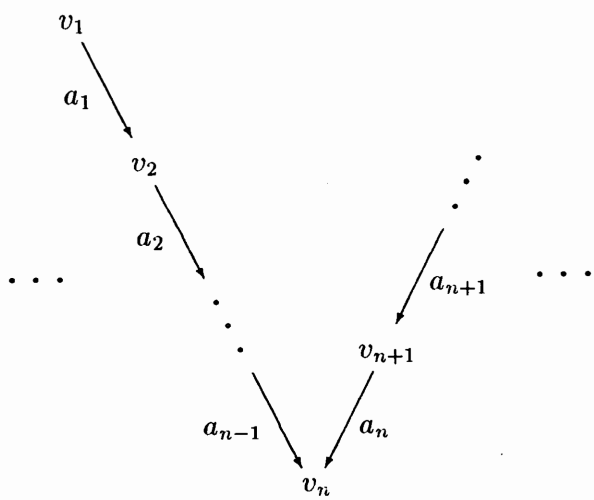


Figure 3.5:

Chapter 4

String Functions and their Images

In this chapter we introduce string functions and show that for every string module there is a string function with an isomorphic image and vice-versa. We will also introduce h-strings and underlying strings two convenient notations for describing a string function. In Chapter 5 we will show that string functions are the homomorphisms that appear in a minimal projective resolution of a string module.

For the remainder of this paper we define a *loop* to be an arrow with the same origin and terminus. Suppose the number of vertices in the quiver for a classical Brauer tree algebra is n, then we are going to form a set of n paths. We will call this set the set of *fixed maximal paths.* Recall that a maximal path, p, with origin v is a path which is not a relation, yet every path properly containing p is a relation. These paths always have the same origin and terminus, and there is always exactly two of them with the same origin. The set of *fixed maximal paths* will contain exactly one maximal path, which is not a loop, with origin v for each vertex v in the quiver.

Given a string S we are going to define a new formal path, an h-string hS. While the notation seems to suggest that an h-string is a string it is not always the case. There are formal paths that are strings but not h-strings and vice-versa. Using an h-string, hS, we will construct a homomorphism hS whose image is isomorphic to $M(S)$. (The abuse of notation will cause no confusion.) We will call these homomorphisms *string functions.*

We define an h-string in the same way that we define a string, except that we do not allow formal paths of length less than 1, and we do allow formal paths that are in the set of fixed maximal paths. However, we do not allow any formal paths that contain maximal paths as proper subpaths.

Before we can create hS, we must first *partition* S. We say that a formal path, S, of length greater than 1, is *partitioned* if it is written in the form $S = s_1 \cdots s_n$ where s_i is a formal path lying entirely within one cycle and for $1 \leq i < n$, s_i and s_{i+1} lie in different cycles. If $S = 1_v$ for some vertex v, then S is also partitioned.

Note that, since any quiver path lying in two cycles contains a relation, if s_i is a quiver path, then s_{i+1} is a formal inverse of a quiver path. Also if s_i is an inverse of a quiver path, then s_{i+1} is a quiver path. This means that all of the formal paths s_k where k is odd are paths in Γ, and all of the formal paths s_j where j is even are formal inverses of paths in Γ, or vice-versa. But there is a one to one correspondence between strings and h-strings.

Given a partitioned string $S = s_1 \cdots s_n$, we define the partitioned h-string, hS to be $s'_n \cdots s'_1$. Where s'_i is defined as follows. For $1 \leq i \leq n$, if s_i is a quiver path, then we define s'_i to be the quiver path such that $s_i s'_i$ is a maximal quiver path. If s_i is a formal inverse of a quiver path then define s'_i to be the formal path such that $s_i s'_i$ is the formal inverse of a maximal quiver path. If $s_i = 1_v$, then we define s'_i to be a fixed maximal path such that $o(s_i) = t(s_i) = v$. It is easy to see that hS is indeed an h-string.

We show that there is a one to one correspondence between strings and h-strings. Given a partitioned h-string, hS, we show how to construct a string function. A partitioned h-string hS can be written in one of four different ways which we list below:

$$
\begin{array}{ll}
1) & hS = P_1Q_1^{-1}P_2Q_2{}^{-}1\cdots P_mQ_m^{-1} \text{ or} \\
2) & hS = P_1Q_1^{-1}P_2Q_2^{-1}\cdots P_m \text{ or} \\
3) & hS = Q_1^{-1}P_2Q_2^{-1}\cdots P_mQ_m^{-1} \text{ or} \\
4) & hS = Q_1^{-1}P_2Q_2^{-1}\cdots P_m \qquad\qquad \text{where } P_i \text{ and } Q_i \text{ are quiver paths}
\end{array}
$$

.

For each of these four h-strings we define a string function hS. We do this in figures 4.1 - 4.4. Along with each definition we provide a diagram called an *underlying string* which assists in visualizing the string function. Also, whenever they are used, c_i and d_i are in the field K, c_i, $d_i \neq 0$, v_i and w_i are vertices in Γ (or Γ^*), $\overline{P}_i$ and $\overline{Q}_i$ are the images of P_i and Q_i in Λ, and $p_i = c_i\overline{P}_i$ and $q_i = d_i\overline{Q}_i$. Also, for brevity, we identify $(0,\ldots,0,\overline{v_i},0,\ldots,0)$ of $v_1\Lambda \oplus \cdots \oplus v_n\Lambda$ with $\overline{v_i}$. After understanding the definition of a string function the reader should note that the functor E introduced in Section 3.2 maps string functions to string functions, and that the corresponding h-strings are identical if the coefficients in Mod(Λ^*) are ignored.

To the h-string $P_1Q_1^{-1}P_2Q_2^{-1}\cdots P_mQ_m^{-1}$, and to the constants c_i and d_i, we associate a homomorphism hS: $v_1\Lambda \oplus \cdots \oplus v_m\Lambda \to w_1\Lambda \oplus \cdots \oplus w_{m+1}\Lambda$ where $hS(v_1)$ $= (c_1\overline{P}_1, d_1\overline{Q}_1, 0, \ldots, 0)$, for $i = 2\cdots(m-1)$ $hS(\overline{v}_i) = (0, \ldots, 0, c_i\overline{P}_i, d_i\overline{Q}_i, 0, \ldots, 0)$ where $c_i\overline{P}_i$ is in the $(i)^{th}$ coordinate, and $hS(\overline{v}_m) = (0,\ldots,0,c_m\overline{P}_m, d_m\overline{Q}_m)$.

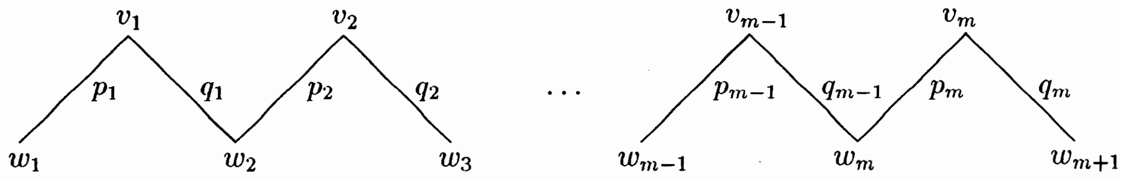


Figure 4.1:

To the h-string $P_1Q_1^{-1}P_2Q_2^{-1}\cdots P_m$, and to the constants c_i and d_i, we associate a homomorphism hS: $v_1\Lambda \oplus \cdots \oplus v_m\Lambda \to w_1\Lambda \oplus \cdots \oplus w_m\Lambda$ where $hS(\overline{v}_1) =$ $(c_1\overline{P}_1, d_1\overline{Q}_1, 0, \ldots, 0)$, for $i = 2\cdots(m-1)$ $hS(\overline{v}_i) = (0, \ldots, 0, c_i\overline{P}_i, d_i\overline{Q}_i, 0, \ldots, 0)$ where $c_i\overline{P}_i$ is in the $(i)^{th}$ coordinate, and $hS(\overline{v}_m) = (0,\ldots,0,c_m\overline{P}_m)$.

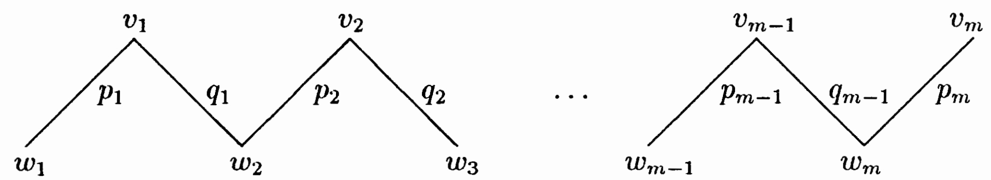


Figure 4.2:

To the h-string $P_1Q_1^{-1}P_2Q_2^{-1}\cdots P_m$, and to the constants c_i and d_i, we associate a homomorphism hS: $v_1\Lambda\oplus\cdots\oplus v_m\Lambda \to hS$: $v_1\Lambda\oplus\cdots\oplus v_m\Lambda \to w_1\Lambda\oplus\cdots\oplus w_m\Lambda$ where $hS(\overline{v}_1) = (d_1\overline{Q}_1, 0, \ldots, 0)$, for $i = 2\cdots(m-1)$ $hS(\overline{v}_i) = (0, \ldots, 0, c_i\overline{P}_i, d_i\overline{Q}_i, 0, \ldots, 0)$ where $c_i\overline{P}_i$ is in the $(i-1)^{st}$ coordinate, and $hS(\overline{v}_m) = (0,\ldots,0,c_m\overline{P}_m, d_m\overline{Q}_m)$.

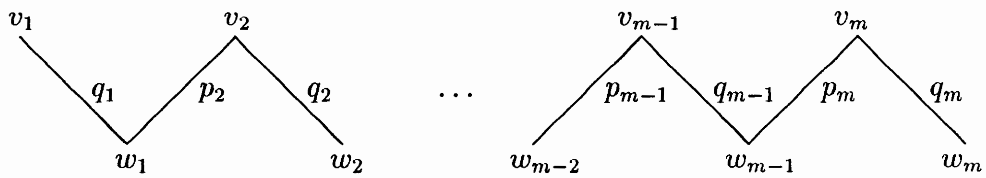


Figure 4.3:

To the h-string $Q_1^{-1}P_2Q_2^{-1}\cdots P_m$, and to the constants c_i and d_i, we associate a homomorphism hS: $v_1\Lambda\oplus\cdots\oplus v_m\Lambda \to hS$: $v_1\Lambda\oplus\cdots\oplus v_m\Lambda \to w_1\Lambda\oplus\cdots\oplus w_{m-1}\Lambda$ where $hS(\overline{v}_1) = (d_1\overline{Q}_1, 0, \ldots, 0)$, for $i = 2\cdots(m-1)$ $hS(\overline{v}_i) = (0, \ldots, 0, c_i\overline{P}_i, d_i\overline{Q}_i, 0, \ldots, 0)$ where $c_i\overline{P}_i$ is in the $(i-1)^{st}$ coordinate, and $hS(\overline{v}_m) = (0,\ldots,0,c_m\overline{P}_m)$.

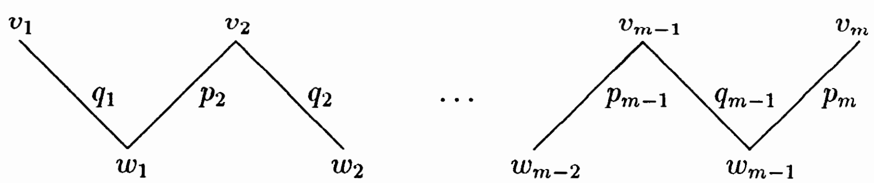


Figure 4.4:

We next prove that all of the string functions associated with a single h-string are isomorphic.

Lemma 4.1 *If f and g are string functions that are both associated with the same h-string, hS, then $Im\, f \simeq Im\, g$.*

Proof. Without loss of generality we can assume that the coefficients in the definition of f are all 1. We will only prove the lemma for string functions of the type described in figure 4.3. The proof is similar for string functions of the other three types. In figure 4.5 we give an underlying string for f and g.

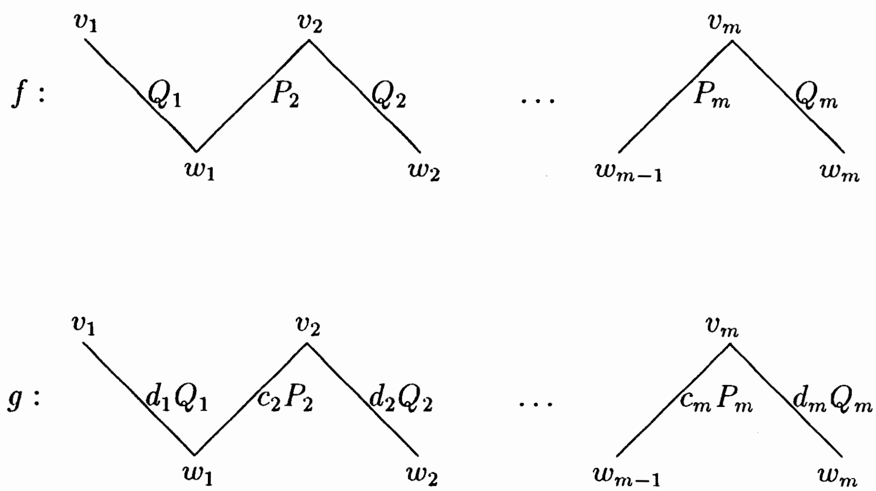


Figure 4.5:

To prove the lemma it suffices to show that there are isomorphisms X and Y such that the following diagram commutes.

$$\begin{tikzcd}
v_1\Lambda \oplus \cdots \oplus v_n\Lambda \arrow[rr, "f"] \arrow[d, "Y"] & & w_1\Lambda \oplus \cdots \oplus w_m\Lambda \arrow[d, "X"] \\
v_1\Lambda \oplus \cdots \oplus v_n\Lambda \arrow[rr, "g"] & & w_1\Lambda \oplus \cdots \oplus w_m\Lambda
\end{tikzcd}$$

Figure 4.6:

We define the isomorphisms X and Y by defining them on the idempotents $v_1, \ldots, v_m$. For $1 \leq i \leq m$ we will define $X(v_i) = x_i v_i$ and $Y(v_i) = y_i v_i$ where $x_i, y_i \in K^*$ need to be determined. To ensure that the diagram commutes it suffices to choose the x_i's and y_i's so that the diagrams commute on the v_i's.

For that to happen we need $x_1 d_1 = y_1$ and for $2 \leq i \leq m$, $x_i c_i = y_{i-1}$ and $x_i d_i = y_i$. Simplifying we get that for $2 \leq i \leq m$, $x_i d_i = x_{i+1} c_{i+1}$. Thus by choosing $x_1 = 1$ and solving recursively we can find the necessary homomorphisms X and Y.

■

Unfortunately using our definition two different h-strings can have associated string functions that have isomorphic images. So we need to identify when this can happen. We do this without proof in Lemma 4.2.

Lemma 4.2 *The string function hS is associated with only the h-strings hS and $(hS)^{-1}$.* ■

We conclude the chapter by proving

Proposition 4.3 *The image of the string function hS is $M(S)$.*

Proof. This is a sketch of the proof for a particular case. The proof clearly generalizes. However, the general case requires an even more burdensome notation.

We consider a graph $QUIV$ and a particular string, S. In figure 4.7 we have drawn a part of the graph $QUIV$. We want to show that the string function hS which corresponds to the h-string $hS = ((f_1 \cdots f_{n_f})(c_1 \cdots c_{n_c})^{-1}(a_1 \cdots a_{n_a})$ has image isomorphic to $M(S)$.

We let $S = (b_1 \cdots b_{n_b})(d_1 \cdots d_{n_d})^{-1}(e_1 \cdots e_{n_e})$ which is a sufficiently complicated to demonstrate all of the necessary proof techniques for the general case. We give the underlying string for hS in figure 4.8, and note that hS is a map from $t(f_{n_f})\Lambda \oplus t(a_{n_a})\Lambda$ to $t(e_{n_e})\Lambda \oplus t(d_{n_d})\Lambda$. It is defined as follows $hS((t(f_{n_f}),0)) = (f_1 \cdots f_{n_f}, c_1 \cdots c_{n_c})$, and $hS((0,t(a_{n_a})) = (0, a_1 \cdots a_{n_a})$. After unwinding the definitions one can see that the image of this map is isomorphic to $M(S)$. We give a representation of $M(S)$ in figure 4.9. There we have drawn all of the nonzero maps between basis elements. Note that each time a vertex v appears in the diagram it represents a different basis element in $M(S)_v$. ■

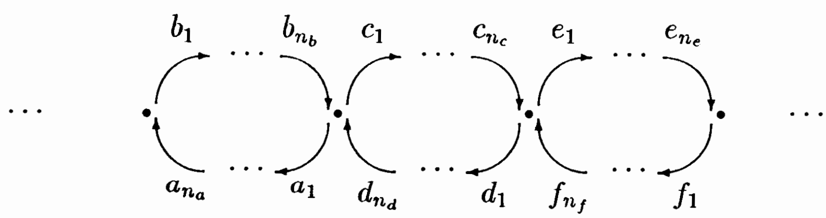


Figure 4.7:

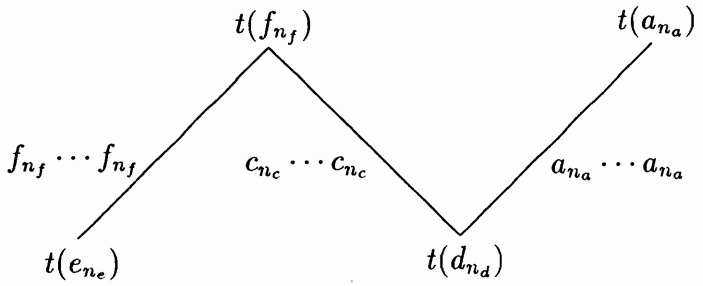


Figure 4.8:

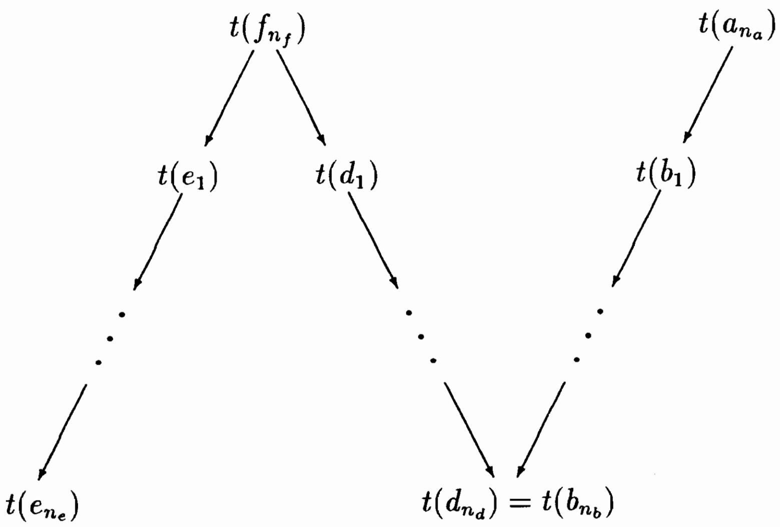


Figure 4.9:

Chapter 5

The Minimal Projective Resolutions of String Modules over Classical Brauer Tree Algebras

In this chapter we will prove that the periodicity of a string module is either $|\Gamma|$ or $2|\Gamma|$. To accomplish this we will develop algorithms for computing minimal projective resolutions for string modules over Λ. We remind the reader that $|\Gamma|$ is the number of vertices in Γ.

Let $\overline{r}$ be the Jacobson radical for Λ. Then we will be using the following definition for a minimal projective resolution of a module M_0: the exact sequence

$$\cdots P_2 \xrightarrow{f_1} P_1 \xrightarrow{f_0} M_0$$

is a minimal projective resolution of the module M_0 if $\mathrm{Ker}(f_i) \subseteq \overline{r}P_{i+1}$ and P_i is projective for $i \geq 1$.

5.1 The Picture Algorithm

Our goal is to give a simple 'walking' (or follow the arrows) algorithm for calculating the minimal projective resolution of an indecomposable module over Λ. But, before we are able to give the 'walking' algorithm we must first develop an intermediate algorithm which we will call the 'picture algorithm'. This algorithm can be applied to string modules over Λ. Hence it can also be applied to Λ^* since Λ^* is a Brauer tree algebra with exceptional number 1. The algorithm can in fact be applied to any path algebra whose quiver consists of cycles of arrows and employs the same type of relations that we have been using. In other words, we can relax the restriction that we look at a graph that is a tree of cycles.

Lemma 5.1 *Given a string function hS. The string function hT that is found using figures 5.1 - 5.10 is such that $Im(hT) = Ker(hS)$.*

Proof. We define two types of vertices *looped* and *nonlooped*. We say that a vertex, v, is a *looped* vertex if there is arrow, a, with $o(a) = t(a) = v$. We say that a vertex is a *nonlooped vertex* otherwise. The proof is broken up into ten cases. These ten cases correspond to ten different forms that an underlying string might assume. In each case we have given the underlying string for hS and directly above it the underlying string for hT. Immediately following each diagram we prove that $\mathrm{Im}(hT) = \mathrm{Ker}(hS)$. We do not concern ourselves with using only fixed maximal paths in the underlying strings. They are only used for ease of notation in other parts of the paper. Here it would be inconvenient. This will cause no problem when we return to examining h-strings. For if a maximal path does appear in an h-string our algorithms will only be concerned with the origin of the path.

Since our main concern will be in resolving simple modules, we can restrict our attention to the string functions where the field coefficients in the underlying string are all ones or alternate between +1 and -1. By that we mean $c_1, d_1, c_2, d_2, \ldots$ alternate between +1 and -1. The proof then becomes somewhat less burdensome with respect to notation. However, the approach remains the same for the more general case. After we have proven the lemma we will note the changes that occur when the sequence $c_1, d_1, c_2, d_2, \ldots$ alternates between +1 and -1.

We keep the same notation that we have been using in previous chapters. If P is a path which lies entirely in one cycle in Γ, then P' is the path such that PP' is a maximal path.

Case 1:

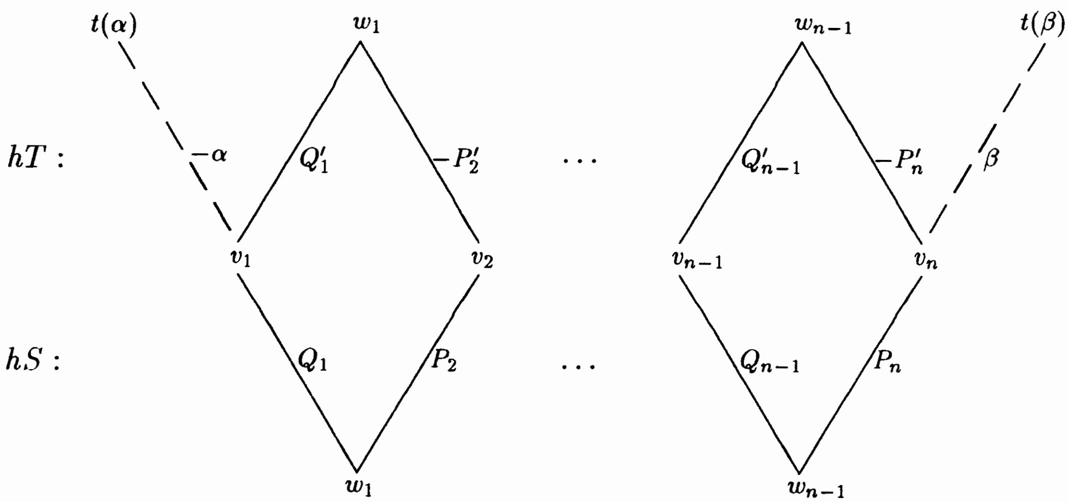


Figure 5.1: We let α be the arrow with origin v_1 which does not lie in the same cycle as Q_1 (or w_1). We let β be the arrow with origin v_n which does not lie in the same cycle as P_n (or w_{n-1}). Assume $n > 1$. The left (resp. right) dotted in portion of T will be present if v_1 (resp. v_n) is a nonlooped vertex.

For the remainder of the proof, when defined, we let q_i (resp. p_{i+1}) be a linear combination of distinct paths leaving v_i (resp. v_{i+1}) in the same cycle as Q_i (resp. P_{i+1}). We first assume that v_1 and v_n are looped. Then to prove hT maps onto $\operatorname{Ker} hS$ we will show that

$\{(0,\ldots,0,\overline{Q}'_i,-\overline{P}'_{i+1},0,\ldots,0) \ : \ 0 \le i < n$ and $\overline{Q}'_i$ is in the i^{th} component$\}$ is a generating set for Ker hS.

Let $x \in \operatorname{Ker} hS$. Then

$x = (\overline{q}_1, \overline{p}_2 + \overline{q}_2, \ldots, \overline{p}_{n-1} + \overline{q}_{n-1}, \overline{p}_n) = \sum_{i=1}^{n-1}(0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0)$. Since the product of two paths lying in different cycles is in I, we have that

$$\begin{aligned} hS(x) &= (\overline{Q}_1\overline{q}_1 + \overline{P}_2(\overline{p}_2+\overline{q}_2), \overline{Q}_2(\overline{p}_2+\overline{q}_2) + \overline{P}_3(\overline{p}_3+\overline{q}_3), \ldots, \overline{Q}_{n-1}(\overline{p}_{n-1}+\overline{q}_{n-1}) + \overline{P}_n\overline{p}_n) \\ &= (\overline{Q}_1\overline{q}_1 + \overline{P}_2\overline{p}_2, \overline{Q}_2\overline{q}_2 + \overline{P}_3\overline{p}_3, \ldots, \overline{Q}_{n-1}\overline{q}_{n-1} + \overline{P}_n\overline{p}_n) \\ &= (0,\ldots,0). \end{aligned}$$

So, in particular, for $0 \le i < n$ we have that

$$\begin{aligned} hS((0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0)) &= (0,\ldots,0,\overline{Q}_i\overline{q}_i + \overline{P}_{i+1}\overline{p}_i,0,\ldots,0) \\ &= (0,\ldots,0). \end{aligned}$$

Therefore, if we define $hS_i = \{y = (0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0) \ : \ y \in \operatorname{Ker} hS\}$, then $\bigcup_{i=1}^{m-1} hS_i$ is a generating set for $\operatorname{Ker} hS$. So we only need to show that for a fixed i $\{y = (0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0) \ : \ y \in \operatorname{Ker} hS\}$ is generated by $(0,\cdots,0,\overline{Q}'_i,-\overline{P}'_{i+1},0,\cdots,0)$.

For the remainder of the proof for $0 \le i < n$, we let b_i be the first arrow of Q_i, and we let a_{i+1} be the first arrow of P_{i+1}. If one of the terms of $\overline{Q}_i\overline{q}_i$ (resp. $\overline{P}_{i+1}p_{i+1}$) is zero in Λ^*, then it must be a Λ^*-multiple of $\overline{Q}_i\overline{Q}'_i\overline{b}_i$ (resp. $\overline{P}_{i+1}\overline{P}'_{i+1}\overline{a}_{i+1}$) and hence generated by $\overline{Q}_i\overline{Q}_i'\overline{b}_i$ (resp. $\overline{P}_{i+1}\overline{P}_{i+1}'\overline{a}_{i+1}$).

So if we subtract from $\overline{Q}_i\overline{q}_i + \overline{P}_{i+1}\overline{p}_{i+1}$ all terms that are zero in Λ^*, then we either have nothing left or we have a linear combination of terms that is zero in $w_i\Lambda^*$. Up to a scalar multiple there is only one such combination. It is $\overline{Q}_i\overline{Q}_i' - \overline{P}_{i+1}\overline{P}_{i+1}'$.

Since we also have for $0 \leq i < n$ $(\overline{Q}_i\overline{Q}_i' - \overline{P}_{i+1}\overline{P}_{i+1}')\overline{b}_i = \overline{Q}_i\overline{Q}_i'\overline{b}_i$ and $(\overline{Q}_i\overline{Q}_i' - \overline{P}_{i+1}\overline{P}_{i+1}')(-\overline{a}_i) = \overline{P}_{i+1}\overline{P}_{i+1}'\overline{a}_i$, we have $\overline{Q}_i\overline{Q}_i' - \overline{P}_{i+1}\overline{P}_{i+1}'$ generates $\{\overline{Q}_i\overline{q}_i + \overline{P}_{i+1}\overline{p}_{i+1} : (0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0)\,\mathrm{Ker}\,hS\}$. Therefore we have that the element $(0,\ldots,0,\overline{Q}_i',-\overline{P}_{i+1}',0,\ldots,0)$ generating hS_i for each i $0 \leq i < n$. We conclude that hT maps onto $\mathrm{Ker}\,hS$.

We now deal with the case where v_1 is nonlooped. We leave the case where v_n is nonlooped to the reader. We will deal with this case by briefly going back through the the preceding proof, noting the places where the proof changes.

First we must define p_1 and α. p_1 will be any linear combination of distinct paths starting at v_1 but not in the same cycle as Q_1. α will be the arrow leaving v_1 not in the same cycle as Q_1.

If $x \in \mathrm{Ker}\,hS$, then

$$x \;=\; (\overline{p}_1 + \overline{q}_1, \ldots, \overline{p}_{n-1} + \overline{q}_{n-1}, \overline{p}_n), \text{and}$$

$$\begin{aligned} hS(x) &= (\overline{Q}_1(\overline{p}_1 + \overline{q}_1) + \overline{P}_2(\overline{p}_2 + \overline{q}_2), \overline{Q}_2(\overline{p}_2 + \overline{q}_2) + \overline{P}_3(\overline{p}_3 + \overline{q}_3), \ldots, \overline{Q}_{n-1}(\overline{p}_{n-1} + \overline{q}_{n-1}) + \overline{P}_n \\ &= (\overline{Q}_1\overline{q}_1 + \overline{P}_2\overline{p}_2, \overline{Q}_2\overline{q}_2 + \overline{P}_3\overline{p}_3, \ldots, \overline{Q}_{n-1}\overline{q}_{n-1} + \overline{P}_n\overline{p}_n) \\ &= (0,\ldots,0). \end{aligned}$$

This implies that $(\overline{p}_1, 0, \ldots, 0) \in \mathrm{Ker}\,hS$. So if we define $hS_0 = \{y = (\overline{p}_1, 0, \ldots,) : y \in \mathrm{Ker}\,hS\}$. Then $\bigcup_{i=0}^{m-1} hS_i$ is a generating set for $\mathrm{Ker}\,hS$.

Finally if p is a path beginning with α then $\overline{Q}_1\overline{p} = 0$ in Λ. So $(-\overline{\alpha}, 0, \ldots, 0)$ generates hS_0. This completes the proof for case one.

Note that we used $(-\overline{\alpha}, 0, \ldots, 0)$ when we could just have easily used $(\overline{\alpha}, 0, \ldots, 0)$. We do this for the sake of consistency and ease of notation.

Case 2:

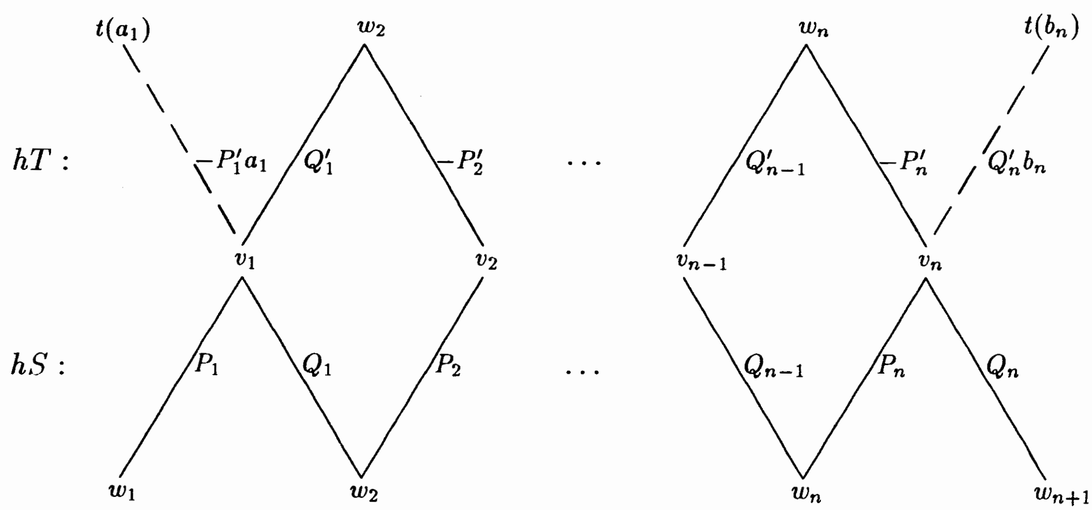


Figure 5.2: The left (resp. right) dotted in portion of hT will be present if P_1 (resp. Q_n) is not an arrow. Assume $(n > 1)$.

Case 2 uses the same format as Case 1 so we will move quickly through the proof noting only the differences.

If $x \in \mathrm{Ker} hS$, then

$$\begin{aligned} x &= (\overline{p}_1 + \overline{q}_1, \overline{p}_2 + \overline{q}_2, \ldots, \overline{p}_n + \overline{q}_n) \\ &= (\overline{p}_1, 0, \ldots, 0) + \sum_{i=1}^{n-1} (0, \ldots, 0, \overline{q}_i, \overline{p}_{i+1}) + (0, \ldots, 0, \overline{q}_n) \end{aligned}$$

Hence we have

$$
\begin{aligned}
hS(x) &= (\overline{P}_1(\overline{p}_1+\overline{q}_1), \overline{Q}_1(\overline{p}_1+\overline{q}_1)+\overline{P}_2(\overline{p}_2+\overline{q}_2), \ldots, \overline{Q}_{m-1}(\overline{p}_{n-1}+\overline{q}_{n-1})+\overline{P}_n(\overline{p}_n+\overline{q}_n), \overline{Q}_n(\\
&= (\overline{P}_1\overline{p}_1, \overline{Q}_1\overline{q}_1+\overline{P}_2\overline{p}_2, \ldots, \overline{Q}_{n-1}\overline{q}_{n-1}+\overline{P}_n\overline{p}_n, \overline{Q}_n\overline{q}_n)
\end{aligned}
$$

So once again for $1 \leq i < n$ we have $(0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0)$ $\in$KerhS. But in addition we have that $(\overline{p}_1,0,\ldots,0)$ and $(0,\ldots,0,\overline{q}_n)$ $\in$KerhS. Therefore, if we let $S_0 = \{(\overline{p}_1,0,\ldots,0) \in \text{Ker } hS\}$, $S_n = \{(0,\ldots,0,\overline{q}_n) \in \text{Ker } hS\}$, and for $1 \leq i < n$ we let $S_i = \{(0,\ldots,0,\overline{q}_i,\overline{p}_{i+1},0,\ldots,0) \in \text{Ker } hS\}$, then $\bigcup_{i=0}^{n} S_i$ is a generating set for Ker hS. As before for $1 \leq i < n$ $(0,\ldots,0,\overline{Q}'_i\overline{P}'_{i+1},0,\ldots,0)$ generates S_i. All that remains is for us find generators for S_0 and S_n.

If $\overline{P}_1\overline{p}_1$ (resp. $\overline{Q}_n\overline{q}_n$) is zero in Λ, then since P_1p_1 (resp. Q_nq_n) contains only paths lying in a single cycle it must be that each of the paths in P_1p_1 (resp. Q_nq_n) contains $P_1P'_1a_1$ (resp. $Q_nQ'_nb_n$). Therefor S_0 (resp. S_n) is generated by (-$\overline{P}'_1a_1$,0,...,0) (resp. $(0,\ldots,0,\overline{Q}'_nb_n)$). We have that hT maps onto Ker hS in either the case where P_1 (resp. Q_n) is an arrow or not.

However, if P_1 (resp Q_n) were the arrow a_1 (resp. b_n), then P_1 (resp. Q_n) would be a maximal path. Thus the map hT including the doted portions would not be a string function. Luckily, the dotted portions become unnecessary in that case. For then $\overline{P}'_1a_1 = \overline{Q}'_1\overline{Q}_1$ (resp. $\overline{Q}'_nb_n = \overline{P}'_n\overline{P}_n$). Hence,

$$
\begin{aligned}
(-\overline{P}'_1a_1,0,\ldots,0) &= (\overline{Q}'_1,-\overline{P}'_2,0,\ldots,0)(-\overline{Q}_1) \\
(resp.(0,\ldots,0,\overline{Q}'_nb_n) &= (0,\ldots,0,\overline{Q}'_{n-1},\overline{P}'_n)(\overline{P}_n))
\end{aligned}
$$

and so $(\overline{Q}'_1,-\overline{P}'_2,0,\ldots,)$ (resp. $(0,\ldots,\overline{Q},Q'_{n-1},\overline{P}'_n)$) generates $S_0 \bigcup S_1$ (resp.$S_{n-1} \bigcup S_n$).

Note that we used ($-\overline{P}'_1 a_1$,0,...,0) instead of the more obvious ($\overline{P}'_1 a_1$,0,...,0). Again we do this for the sake of consistency and to ease later notation.

Case 3:

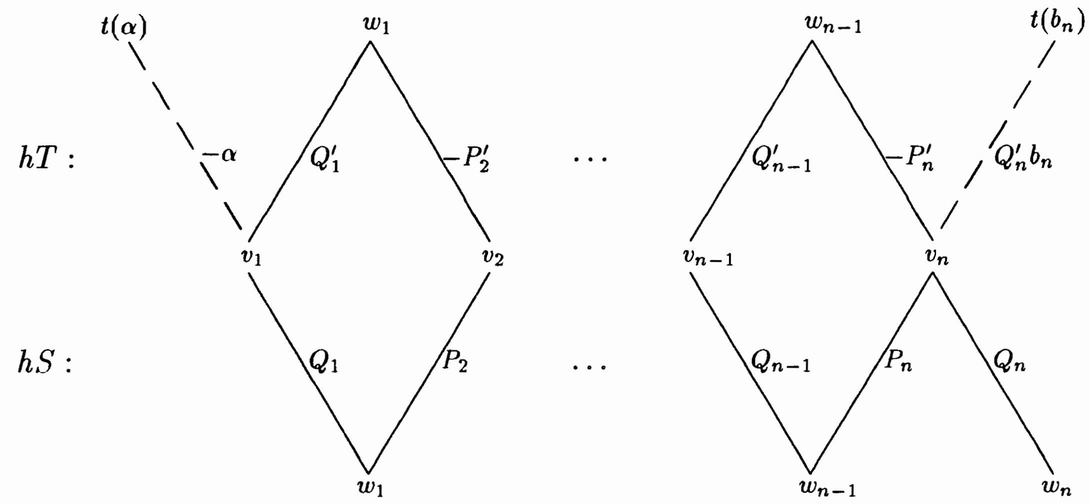


Figure 5.3: Once again α is the arrow leaving v_1 not in the same cycle as $\overline{Q}_1$. The left dotted in part of hT is included if v_1 is a nonlooped vertex. The right dotted in part of hT is included if Q_n is not an arrow. Assume ($n > 1$).

This case as well as the remaining cases use proof techniques similar to the ones presented in Cases 1 and 2. Hence we list the Cases without proof.

Case 4:

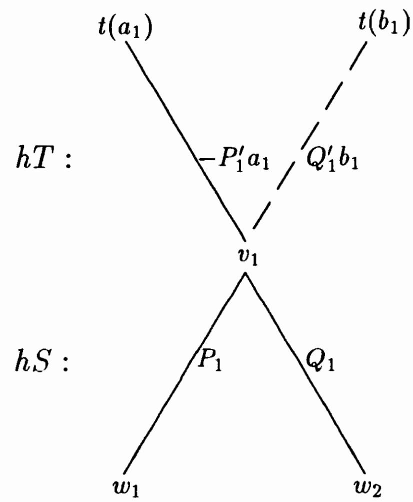


Figure 5.4: We assume that v_1 nonlooped. We also assume that if exactly one of P_1 or Q_1 is an arrow, then it is Q_1. The dotted in part will appear only when Q_1 is not an arrow.

Case 5:

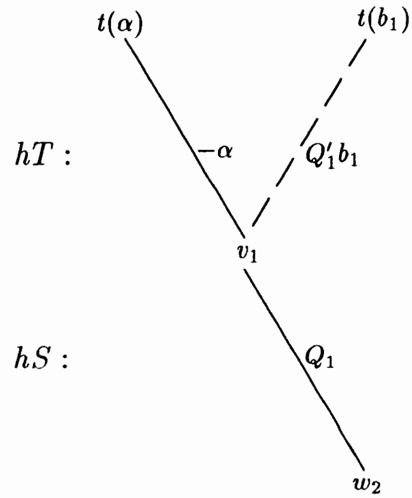


Figure 5.5: We assume that v_1 nonlooped. The dotted in part is included only if Q_1 is not an arrow. If Q_1 is a maximal path, we define Q_1' to be the vertex v_1.

Case 6:

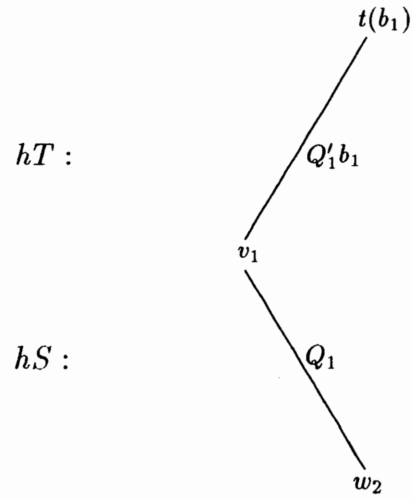


Figure 5.6: We assume that v_1 looped. If Q_1 is a maximal path, we define Q_1' to be the vertex v_1.

Case 7:

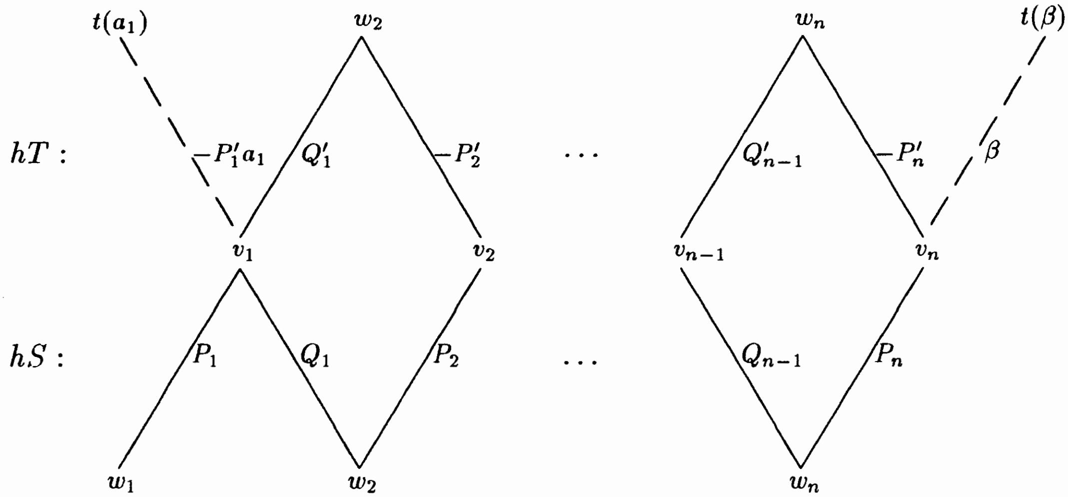


Figure 5.7: We assume that $(n > 1)$. The left dotted in part of hT is included if P_1 is not an arrow, the right dotted in part is included if v_n is a nonlooped vertex.

Case 8:

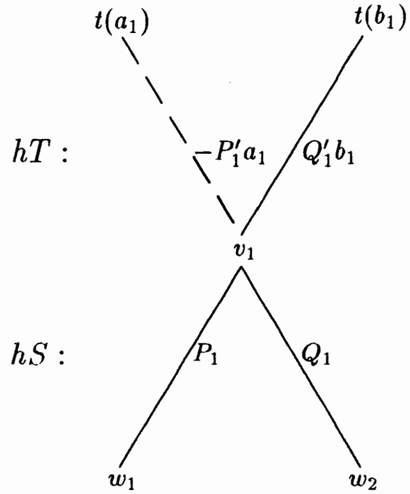


Figure 5.8: We assume that v_1 nonlooped. We assume that Q_1 is never an arrow. The dotted in part of hT will be included if P_1 is not an arrow.

Case 9:

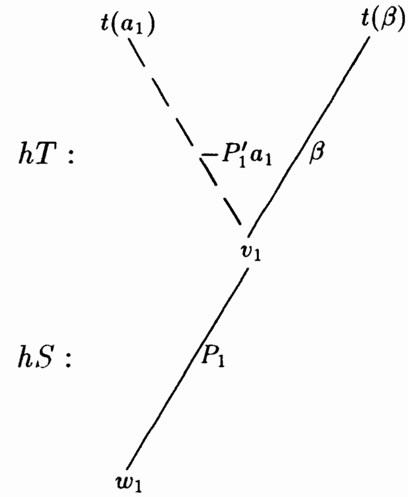


Figure 5.9: We assume that v_1 nonlooped. The dotted in part will be included if P_1 is not an arrow. If P_1 is a maximal path, we define P_1' to be the vertex v_1.

Case 10:

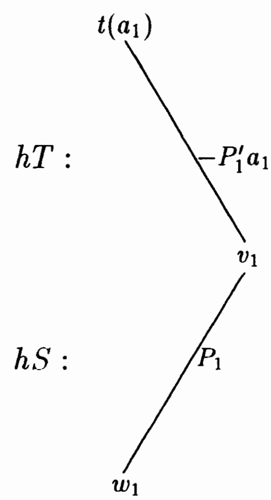


Figure 5.10: We assume that v_1 looped. If P_1 is a maximal path, we define P_1' to be the vertex v_1.

It can easily be shown that in each of the 10 cases above whenever the underlying string for hS has coefficients that are all $+1$, the coefficients in hT alternate between

+1 and -1. If instead the coefficients of hS alternate between +1 and -1, then we would have the coefficients of hT being all +1. ■

Lemma 5.2 *If $f : M \to N$ is a string function, then $Ker(f) \subseteq \overline{r}M$.*

Proof. Without much work it can be shown that $\overline{r}$ is generated by $\{\overline{a}$: a is an arrow in the quiver$\}$. By Lemma 5.1 $\ker(f) = \mathrm{Im}(g)$ for some string function g. By the definition of a string function it is clear that $\mathrm{Im}(g) \subseteq \overline{r}M$. ■

The Picture Algorithm

Given hS_0 we construct the minimal projective resolution of $M(S_0)$ recursively. hS_{i+1} is found by letting hS_i play the role of hS in Lemma 5.1. Use the appropriate case to find hT which is hS_{i+1}. Lemmas 5.1 and 5.2 prove that the algorithm constructs the desired resolution.

5.2 A Simpler Notation

In this section we will restrict our attention to classical Brauer tree algebras with exceptional number 1. In other words, we will be considering Λ^*. For in this case we can simplify the notation. Since we are able to construct the minimal projective resolutions for Λ-string modules from Λ^*-string modules this is a worthy endeavor. It will primarily be beneficial when we are constructing the minimal set of generators for the cohomology ring of Λ.

We define a *string-tuple* to be a 4-tuple (x, l, y, r) where $x, y \in \{up, down\}$ and l and r represent vertices of Γ^*, or the edges in the underlying Brauer tree on which the vertices lie. We also insist on a certain relationship between x and y. Recall that the quiver for a classical Brauer tree has an underlying Brauer tree, where each vertex lies on an edge of the underlying Brauer tree. Recall also that we use the same name for the edge and the vertex lying on it. Let $w_1 \cdots w_j$ be the edge walk from the edge l to the edge r ($w_1 = l$ and $w_j = r$.) Then y = down if and only if x = up and j is even or x = down and j is odd.

Proposition 5.3 *There is a 1-1 correspondence between string-tuples and h-strings.*

Proof. Let $hS = s_1 \cdots s_n$ be a partitioned h-string. We describe the string-tuple (x, l, y, r) that corresponds to hS and show how we will use the information in the string-tuple to uniquely determine hS. Let $l = o(s_1)$, $r = t(s_n)$, $x = up$ if s_1 is the formal inverse of a quiver path and *down* otherwise, and $y = up$ if s_n is a quiver path and *down* otherwise.

Recall that each vertex in Γ^* sits on an edge of the underlying Brauer tree for Γ^*. These vertices and the edges that they sit on share the same name. The edges $o(s_1)t(s_1)t(s_2)\cdots t(s_n)$ form a connected edge path. Since the exceptional number is 1 and since no s_i is a maximal path, the walk never doubles back on itself. Since this walk takes place on a tree, it is the unique walk from the edge $o(s_1)$ to the edge $t(s_n)$ with no repeating edges. So, given $l = o(s_1)$ and $r = t(s_n)$, we can build the sequence $o(s_1)t(s_1)t(s_2)\cdots t(s_n)$. It this sequence together with the variable x that will allow us to build hS.

In our construction whenever we are choosing a maximal path we will always choose from our collection of fixed maximal paths. We build the h-string $s'_1 \cdots s'_n$. If x = down, we choose s'_1 to be the quiver path lying in one cycle from $o(s_1)$ to $t(s_1)$. If x = up, we choose s'_1 to be the formal inverse of that quiver path. Clearly $s'_1 = s_1$. For $2 \leq i < n$, we choose s'_i to be either the quiver path lying in one cycle from $o(s_1)$ to $t(s_1)$ or its formal inverse. Beginning with s'_1 we alternate between choosing a formal inverse and a quiver path. By construction the h-string, $s'_1 \cdots s'_n$, is the h-string $s_1 \cdots s_n$.

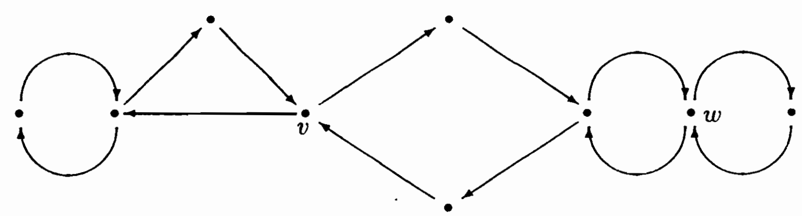
Note that we did not use y. The information contained in y could be used in a similar fashion to the way x was used to complete the description of hS. The two different approaches do not yield contradictory conclusions. In fact the variable y is unnecessary. However, its presense simplifies later proofs. So we leave it in. ■

Lemma 4.2 now takes the form of Corollary 5.4.

Corollary 5.4 *A string function associated with (x, l, y, r) can only be associated with the string-tuples (x, l, y, r) or (y, r, x, l). Hence up to an isomorphism of the image of a string function we have that each string function has only these two string-tuples associated with it.*

Using the string-tuple notation we are going to restate some of the results of the picture algorithm in Corollary 5.5. But first we need some new notation. We will say that the vertex v is *one arrow away from* the vertex w if there is an arrow from w to v. If $QUIV$ is a quiver for a classical Brauer tree algebra, v is a nonlooped vertex in $QUIV$, and w is any other vertex in $QUIV$, then we define a new smaller quiver $QUIV_{vw}$. We do this by considering the underlying Brauer tree. There are two subtrees that contain each edge. The quiver constructed using the subtree that contains v and w we will call $QUIV_{vw}$. We illustrate the idea in figure 5.11.
Definition: We will let $|QUIV_{vw}|$ be the number of vertices in $QUIV_{vw}$.

$QUIV :$



$QUIV_{vw} :$

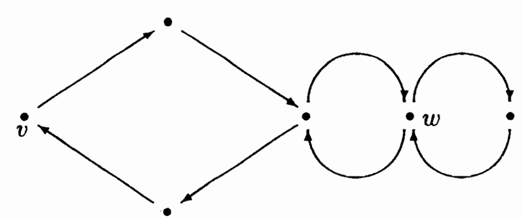


Figure 5.11:

Corollary 5.5 *Given that* $\{(x_i, l_i, y_i, r_i)\}_{i=0}^{\infty}$ *is a minimal projective resolution of the image of* (x_0, l_0, y_0, r_0)*, we give selective information about* (x_i, l_i, y_i, r_i) *in terms of* $(x_{i-1}, l_{i-1}, y_{i-1}, r_{i-1})$*. We do this by allowing* $(x_{i-1}, l_{i-1}, y_{i-1}, r_{i-1})$ *to play the role of hS in Lemma 5.1 and carefully considering the ten different cases. For each case we will list the cases that would need to be considered. We will be assuming the notation of Lemma 5.1.*

A If $l_{i-1} = r_{i-1}$ *when* l_i *is a nonlooped vertex, then the next string-tuple will be (up,*l_i*,up,*r_i*) where* l_i *and* r_i *will be the two vertices, say u and w, one arrow away from* l_{i-1}*. We are free to choose* l_i *to be u or w. We make a particular choice when we present the 'walking algorithm'. To verify this consider cases 5 and 9 of the picture algorithm. For B – F below we will assume that we do not have* $l_i = r_i$ *when* l_i *is a nonlooped vertex.*

B If x_{i-1} $=$ *down and* l_{i-1} *is looped, then* l_i *is one arrow away from* l_{i-1} *where* *the arrow is not a loop. Consider cases 2,4,7,8,9, and 10. We have that* $l_i = v_1$ *when* p_1 *is an arrow, and otherwise* $l_i = t(a_1)$ *where* a_1 *is the first arrow in* P_1*. In either case we have the result.*

C *If* x_{i-1} = *down and* l_{i-1} *is nonlooped, then* l_i *is one arrow away from* l_{i-1} *in* $\Gamma^*_{l_{i-1}r_{i-1}}$*. The same argument used in B works here if we also note that* P_1 *lies in* $\Gamma^*_{l_{i-1}r_{i-1}}$*.*

D *If* x_{i-1} = *up and* l_{i-1} *is nonlooped, then* x_i = *up and* l_i *is one arrow away from* l_{i-1} *not in* $\Gamma^*_{l_{i-1}r_{i-1}}$*. Consider cases 1, 3, and 5. Note that* α *is an arrow leaving* v_1 *not in the cycle containing* Q_1 *and hence not in* $\Gamma^*_{l_{i-1}r_{i-1}}$*.*

E *If* x_{i-1} = *up and* l_{i-1} *is looped, then* x_i = *down and* $l_i = l_{i-1}$*. See cases 1, 3, and 6.*

F *If* l_i *is looped and* $l_i \neq l_{i-1}$*, then* x_i = *up. We only get* x_i = *down if* $l_i = l_{i-1}$ *where* l_i *is looped or if* $l_i = v_1$ *is nonlooped. Consider all cases.*

G *All of the (obvious) analogs for B – F concerning* x_{i-1}*,* l_{i-1}*,* x_i*, and* l_i *hold for* y_{i-1}*,* r_{i-1}*,* y_i*, and* r_i*. The proofs look at different cases but the symmetry makes this easy.*

Let $v \leftrightarrow w$ represent the connected edge path in the underlying Brauer tree for Γ^* from v to w that contains no edge twice. Then we have the following corollary.

Corollary 5.6 *Let* $\{(x_i, l_i, y_i, r_i)\}_{i=0}^{\infty}$ *be the minimal projective resolution of the image of* (x_0, l_0, y_0, r_0)*. Then provided* $l_i \neq r_i$*,* x_i = *up if and only if* l_i *does not appear in* $l_{i-1} \leftrightarrow r_{i-1}$*. Also* y_i = *up if and only if* r_i *does not appear in* $l_{i-1} \leftrightarrow r_{i-1}$*. If* $l_i = r_i$*, then we may choose* x_i = *up and* y_i = *down or vice-versa.*

The reader should examine the picture algorithm to see that this proposition is valid. Do this by noting that the edges in $l_i \leftrightarrow r_i$ are the vertices that appear from left to right in the underlying string corresponding to (x_i, l_i, y_i, r_i).

5.3 The Walking Algorithm

Let $QUIV$ be the quiver associated with a classical Brauer tree algebra. If there are m vertices in $QUIV$ then we are going to label $2m$ evenly spaced points on a unit circle using the m vertices of $QUIV$. We will call this the $QUIV$-unit circle. Each vertex will be used exactly twice in the labeling process. *Definition:* We define any sequence that is formed by recording the labeled points in a clockwise direction as a *$QUIV$-walking sequence.* We will give two different algorithms for labeling the $QUIV$-unit circle the *closed curve algorithm* and the *arrow algorithm.* We will need to refer to each from time to time.

The Closed Curve Algorithm

We fix an embedding of $QUIV$ in a plane with a clockwise orientation. In other words we insist that every cycle has the arrows moving in a clockwise direction. A point will be considered to be in the exterior of $QUIV$ if it is not on the graph or bound by the graph. If d is the smallest distance between any two vertices in $QUIV$, then we place in the plane a simple closed curve, C, with a clockwise orientation. We insist that C follows the boundary of the exterior of $QUIV$ at a distance of $d/8$. See figure 5.12.

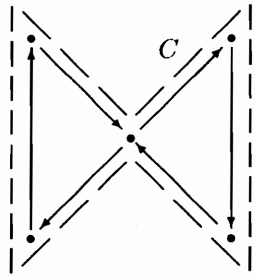


Figure 5.12:

Place the center of a circle, D, of radius $d/4$ any where on the curve C such that no vertex of $QUIV$ lies within D. Then, keeping the center of the circle on C, begin to slide the circle D in a clockwise direction. Choose any point on the $QUIV$-unit circle and label it the name of the first vertex to enter D. Continue to label the points on the $QUIV$-unit circle in a clockwise fashion using each successive vertex that enters D. Stop recording when D returns to its starting location.

Consider figure 5.13 and note that every vertex enters D exactly twice as it traverses C. Thus as we claimed we have labeled exactly $2m$ points on the $QUIV$-unit circle.

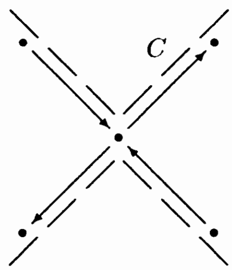


Figure 5.13:

The Arrow Algorithm

1. • Choose a looped vertex v.

   • Label any two successive points on the $QUIV$-unit circle v.

   • Choose the vertex one arrow away from v. We insist that the arrow not be a loop.

   • Go to step 2.

2. • If the chosen vertex is v, then stop. We have recorded every vertex twice. Otherwise

   • If the chosen vertex is a looped vertex, go to step 3.

   • If the chosen vertex is a nonlooped vertex, go to step 4.

3. • Continuing in a clockwise direction we label the next two points on the $QUIV$-unit circle as the chosen looped vertex.

   • Choose the vertex one arrow away from that looped vertex. We insist that the arrow is not a loop.

   • Go to step 2.

4. • Continuing in a clockwise direction we label the next point on the $QUIV$-unit circle as the chosen nonlooped vertex.

   • Choose the vertex one arrow away the nonlooped vertex not in the same cycle as the previously recorded vertex.

   • Go to step 2.

The Walking Algorithm

Recall that a Γ-walking sequence is any sequence formed by recording the points on the Γ-unit circle in a clockwise direction. A minimal projective resolution, $\{(x_i, l_i, y_i, r_i)\}_{i=0}^{\infty}$, for the image of some string function $hS_0 \in \Lambda^*$ can be computed by using the following algorithm.

1. Choose a string-tuple (x_0, l_0, y_0, r_0) that corresponds to hS_0.

2. Use Corollaries 5.5 and 5.6 to construct (or choose) (x_1, l_1, y_1, r_1).

3. Construct the Γ^*-walking sequence $\{l_i\}_{i=0}^{\infty}$ which begins with the Γ^*-walking sequence $\{l_0, l_1\}$.

4. Construct the Γ^*-walking sequence $\{r_i\}_{i=0}^{\infty}$ which begins with the Γ^*-walking sequence $\{r_0, r_1\}$.

5. Use Corollaries 5.6 to construct the sequences $\{x_i\}_{i=0}^{\infty}$ and $\{y_i\}_{i=0}^{\infty}$.

To validate this algorithm we only need to prove prove Proposition 5.7.

Proposition 5.7 *If Corollary 5.5 is used to construct the sequences $\{l_i\}_{i=0}^{\infty}$ and $\{r_i\}_{i=0}^{\infty}$, then those sequences can be chosen to be Γ^*-walking sequences.*

Proof. We will show that for $0 \leq i$, that $\{l_0, l_1, \ldots, l_{i+1}\}$ and $\{r_0, r_1, \ldots, r_{i+1}\}$ are Γ^*-walking sequences. The proof will proceed by induction on i. Using Corollary 5.5 to construct $\{l_i\}_{i=0}^{\infty}$ (resp. $\{r_i\}_{i=0}^{\infty}$), it is clear that for $i \geq 0$, l_i (resp. r_i) is always one arrow (possibly a loop) away from l_{i-1} (resp. r_{i-1}). Since any two vertices one arrow (possibly a loop) away from each other appear successively in a clockwise direction somewhere on the Γ^*-unit circle, we have that $\{l_0, l_1\}$ and $\{r_0, r_1\}$ are Γ^*-walking sequences. We now assume that $\{l_0, l_1, \ldots, l_i\}$ $\{r_0, r_1, \ldots, r_i\}$ are Γ^*-walking sequences and show that $\{l_0, l_1, \ldots, l_i, l_{i+1}\}$ and $\{r_0, r_1, \ldots, r_i, r_{i+1}\}$ are Γ^*-walking sequences. We do this by showing that the choice of l_{i+1} (resp. r_{i+1}) using Corollary 5.5 is consistent with the choice of l_{i+1} (resp. r_{i+1}) using the arrow algorithm.

We will first consider the proof when $l_i \neq r_i$. We note that when $l_i \neq r_i$, Corollary 5.5 allows no flexibility in the choice of l_{i+1} and r_{i+1}. In that case by symmetry we can simply show that the choice of l_{i+1} using Corollary 5.5 is consistent with the choice of l_{i+1} using the arrow algorithm. We consider case where l_i is a looped vertex separately from the case where l_i is a nonlooped vertex.

1. **l_i is looped.** If $l_{i-1} = l_i$, l_{i+1} is one arrow away from l_i. On the other hand, if $l_{i-1} \neq l_i$ then $l_{i+1} = l_i$. In either case the looped vertices appear in $\{l_i\}_{i=0}^{\infty}$ in a way that is consistent with the arrow algorithm.

2. **l_i is nonlooped.**

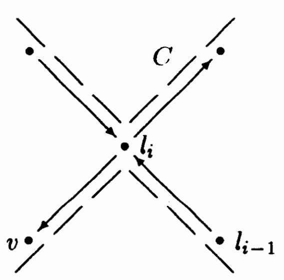


Figure 5.14:

Using figure 5.14 as a reference for our notation, our goal here is to show that Corollary 5.5 chooses l_{i+1} to be the vertex v. We consider two cases.

(a) We assume r_{i-1} is in $\Gamma^*_{l_i l_{i-1}}$ but that $r_{i-1} \neq l_i$. Therefore l_i is not in $l_{i-1} \leftrightarrow r_{i-1}$. So by Corollary 5.6, $x_i =$ up. Corollary 5.5 chooses l_{i+1}

to be one arrow away from l_i not in $\Gamma^*_{l_ir_i}$. Since r_i must be at most one arrow away from r_{i-1}, r_i must be in $\Gamma^*_{l_il_{i-1}}$. Therefore since $r_i \neq l_i$, we have $\Gamma^*_{l_ir_i} = \Gamma^*_{l_il_{i-1}}$. Hence $l_{i+1} = v$.

(b) We assume r_{i-1} is in $\Gamma^*_{l_iv}$. Then l_i is in $r_{i-1} \leftrightarrow l_{i-1}$ and so x_i = down. Therefore l_{i+1} is one arrow from l_{i-1} in $\Gamma^*_{l_ir_i}$. Since r_i is at most one arrow away from r_{i-1} which is in $\Gamma^*_{l_iv}$ and $r_i \neq l_i$, we have that r_i is in $\Gamma^*_{l_iv}$ and therefore $\Gamma^*_{l_ir_i} = l_iv$. So since x_i = down, l_{i+1} is one arrow away from l_i in $\Gamma^*_{l_ir_i} = \Gamma^*_{l_iv}$. Hence $l_{i+1} = v$.

We now consider the proof when $l_i = r_i$. It is clear from Corollary 5.5 that the Γ^*-walking sequences $\{l_{i-1}, l_i\}$ and $\{r_{i-1}, r_i\}$ are distinct with $l_i = r_i$ a vertex one arrow (not a loop) away from each of l_{i-1} and r_{i-1}. Since $l_i = r_i$, we may apply part A of Corollary 5.5. It says that we may choose l_{i+1} to be either of the two vertices that are one arrow away from l_i, and choose r_{i+1} to be the other vertex. Therefore using Corollary 5.5 we can choose l_{i+1} and r_{i+1} to be consistent with the choice that would be made by the arrow algorithm. This completes the inductive step of the argument. We have the result. ■

Let $\{(x_m, l_m, y_m, r_m)\}_{m=0}^{\infty}$ be the sequence of string- tuples corresponding to the string functions in the minimal projective resolution of the module $M(S) \in \Gamma^*$ which is constructed using the walking algorithm. *Definition:* We will call such a sequence a *walking sequence* for the module $M(S)$.

5.4 The Periodicity of a String Module

Let $\{hS_i\}_{i=0}^{\infty}$ be a minimal projective resolution of the string module $M(S_0)$ in Λ. If k is the least positive integer such that $M(S_k) \simeq M(S_0)$, then we say that the *periodicity* of $M(S_0)$ is k. Our main goal in this section is to give a method for determining the periodicity for any string module. We will prove that the periodicity is either $|\Gamma|$ or $2|\Gamma|$ for all string modules. In Theorem 5.13 we will give a test to determine the periodicity of a string module $M(S_0)$.

For the remainder of the chapter we will let N be the exceptional number for Γ. We let t be the number of times the arrow of weight $\overline{1}$ appears in S_0. Then using the construction of Proposition 3.3 we find S_0^* has associated to it a string-tuple of the form $(x_0, l_0^{\overline{0}}, y_0, r_0^{\overline{0}})$. We let $\{hS_i^*\}_{i=1}^{\infty}$ be the sequence of string functions in the minimal projective resolution of $M(S_0^*)$. Also let $\{(x_i, l_i^{\overline{m_i}}, y_i, r_i^{\overline{m_i}})\}_{i=0}^{\infty}$ be a Γ^*-walking sequence for $M(S_0^*)$.

Lemma 5.8

1. *If i is the least integer greater than 0 such that $E(M(S_i^*)) = M(S_0)$, then the periodicity of $M(S_0) = i$.*

2. *i satisfies (1) above if and only if S_i^* corresponds to a string-tuple of the form $(x_0, l_0^{\overline{0+j}}, y_0, r_0^{\overline{t+j}})$ or $(y_0, r_0^{\overline{t+j}}, x_0, l_0^{\overline{0+j}})$ where j is an integer.*

Proof. (1) follows from the fact that the functor E preserves minimal projective resolutions. (2) follows from the proof of Proposition 3.3 ■

Lemma 5.9 *If the superscripts of the vertices in a Γ^*-walking sequence are removed, then the Γ^*-walking sequence becomes a Γ-walking sequence. Also given the superscript of the first vertex of the Γ^*-walking sequence we can construct the Γ^*-walking sequence from the Γ-walking sequence.*

Proof. The first statement is an immediate consequence of the fact that Γ^* and Γ are locally homeomorphic. Letting $\overline{m_0}$ be the superscript of the first vertex of a

Γ^*-walking sequence. Then if $\{w_i\}_{i=0}^{\infty}$ is the Γ-walking sequence that we construct by dropping the superscripts from a Γ^*-walking sequence, we construct the original Γ^*-walking sequence recursively.

For $i > 0$, if $o(a) = w_{i-1}$ and $t(a) = w_i$, then $\overline{m}_i = \overline{m_{i-1}+1}$. Otherwise $\overline{m}_i = \overline{m}_{i-1}$. ∎

Lemma 5.10 *If Γ consists of a single vertex and two loops, then the periodicity for all string modules over Λ is 1 when the exceptional number is 1, and 2 otherwise.*

Proof. If the exceptional number is 1, then there is only one string module. If the exceptional number is 2, then there are two string modules with the image of one being the kernel of the other. ∎

For Lemmas 5.11 through 5.13 we assume that Γ is not the degenerate case that we just handled in Lemma 5.10.

Lemma 5.11 *Let k be the periodicity of the string module $M(S_0)$, then k divides $2|\Gamma|$.*

Proof. It suffices to show that $E(M(S_{2|\Gamma|})) \simeq M(S_0)$. Since the number of labeled points on the Γ-unit circle is $2|\Gamma|$, by Lemma 5.9, we have that $l_{2|\Gamma|}^{\overline{m_{2|\Gamma|}}} = l_0^{\overline{1}}$ (resp. $r_{2|\Gamma|}^{\overline{m_{2|\Gamma|}}} = r_0^{\overline{t+1}}$). Suppose $l_0^{\overline{0}} = r_0^{\overline{t}}$. Since $l_0^{\overline{0}} = r_0^{\overline{t}}$ by Lemma 5.8 we only need to show that $x_{2|\Gamma|} = x_0$ and $y_{2|\Gamma|} = y_0$ or $x_{2|\Gamma|} = y_0$ and $y_{2|\Gamma|} = x_0$. But when we are looking at strings-tuples in Γ^*. If we have (x, v, y, v), either x or y is up and the other is down. So we have $E(M(S^*_{2|\Gamma|})) = M(S_0)$.

Suppose $l_0^{\overline{0}} \neq r_0^{\overline{t}}$. At the beginning of this proof we showed that $l_{2|\Gamma|}^{\overline{m_{2|\Gamma|}}} = l_0^{\overline{1}}$ and that $r_{2|\Gamma|}^{\overline{m_{2|\Gamma|}}} = r_0^{\overline{t+1}}$. Therefore, if we prove that $x_{2|\Gamma|} = x_0$ and $y_{2|\Gamma|} = y_0$, we will have proven that $E(M(S^*_{2|\Gamma|})) = M(S_0)$. However, since the number of edges in $l_0^{\overline{0}} \leftrightarrow r_0^{\overline{t}}$ is the same as the number of edges in $l_0^{\overline{1}} \leftrightarrow r_0^{\overline{t+1}}$, and since that number together with x_0 (resp. $x_{2|\Gamma|}$) determines y_0 (resp. $y_{2|\Gamma|}$), we only need to show that $x_{2|\Gamma|} = x_0$.

Since, $l_0^{\overline{0}} \neq r_0^{\overline{t}}$, $l_1^{\overline{m_1}}$ can be determined by x_0 and $l_0^{\overline{0}} \leftrightarrow r_0^{\overline{t}}$. Similarly, since $l_{2|\Gamma|}^{\overline{1}} \neq r_{2|\Gamma|}^{\overline{t+1}}$, $l_{2|\Gamma|+1}^{\overline{m_{2|\Gamma|+1}}}$ can be determined by x_0 and $l_{2|\Gamma|}^{\overline{1}} \leftrightarrow r_{2|\Gamma|}^{\overline{t+1}}$. Since $l_{2|\Gamma|}^{\overline{1}} \leftrightarrow r_{2|\Gamma|}^{\overline{t+1}}$ can be found by adding $\overline{1}$ to each of the superscripts in $l_0^{\overline{0}} \leftrightarrow r_0^{\overline{t}}$, and since $l_{2|\Gamma|+1}^{\overline{m_{2|\Gamma|+1}}} = l_1^{\overline{m_1}}$ we must have $x_{2|\Gamma|} = x_0$. In other words, since those portions of Γ^* are homeomorphic, we must have $x_{2|\Gamma|} = x_0$. In either case we have the result. ■

Lemma 5.12 *If* $0 < k < 2|\Gamma|$, *then for* $k \neq |\Gamma|$, $E(M(S_k^*))$ *is not isomorphic to* $M(S_0)$. *If* $E(M(S_k^*))$ *is not isomorphic to* $M(S_0)$, *then the periodicity of* $M(S_0)$ *is* $2|\Gamma|$.

Proof. The second statement follows from the first statement and Lemma 5.11. We prove the first statement. We first consider the case where $l_0 = r_0 = v$ where v is some vertex. We assume that i is such that $E(M(S_i^*))$ is isomorphic to $M(S_0)$. Hence, we must have $l_i = r_i = v$. Also using Corollary 5.5 we can see that sequences $\{l_i\}_{i=0}^{\infty}$ and $\{r_i\}_{i=0}^{\infty}$ have different starting places on the Γ-unit circle. Hence, the walking sequences $l_0, l_1, \ldots, l_{i-1}, r_1, r_2, \ldots, r_i$ and $l_0, l_1, \ldots, l_{2|\Gamma|}$ are identical. Therefore $i = \frac{2|\Gamma|}{2} = |\Gamma|$. (Note that this could only happen if v is a nonlooped vertex, or Γ is the degenerate case we are not considering here).

Suppose $l_0 \neq r_0$. Then we will assume that i is such that $E(M(S_i^*))$ is isomorphic to $M(S)$ and proceed to identify the i's for which this can be true. The sequence $l_0, l_1, \ldots, l_{2|\Gamma|-1}$ appears on the Γ-unit circle if we record the points in a clockwise direction. Recall that although there are $2|\Gamma|$ points on the unit circle, we only use the $|\Gamma|$ vertex labels to label them. Thus the label l_0 appears twice in the sequence $l_0, l_1, \ldots, l_{2|\Gamma|-1}$. We locate the point labeled l_0 which is not the first point in the sequence. For the remainder of this lemma we refer to the point that we have just located as L_0. So that now we will refer to the vertex l_0 as l_0 or L_0. We define R_0 in a similar fashion. We have then that the second time the vertex $v = l_0$ appears in the Γ-walking sequence it will be labeled L_0. The third time it appears it will be labeled l_0 once again. With this new notation we can now describe the four possible scenarios that would lead to having the set of vertices $\{l_0, r_o\}$ being the same as the set of vertices $\{l_i, r_i\}$, a necessary condition for $E(M(S_i^*)) = M(S_0)$.

Case 1: $l_i = L_0$, $r_i = R_0$

Since $l_i = L_0$ appears in a different location on the Γ-unit circle than l_0, $l_{i+1} \neq l_1$. Similarly we have $r_{i+1} \neq r_1$. Yet if the periodicity of $M(S_0)$ is i, then we must have $E(M(S_{i+1}^*)) = M(S_1)$. So it must be that $r_{i+1} = l_1$ and $l_{i+1} = r_1$. We have four cases to consider. They each address what happens when only some of the equalities of equations 5.1 - 5.4 hold.

$$l_0 = l_1 \tag{5.1}$$

$$l_i = l_{i+1} \tag{5.2}$$

$$r_0 = r_1 \tag{5.3}$$

$$r_i = r_{i+1} \tag{5.4}$$

Case 1a: Suppose none of the equalities of equations 5.1 – 5.4 hold. In this case it must be that part of Γ must have the form given in figure 5.15. Since no tree of cycles can have such an arrangement, we see that this case can not occur.

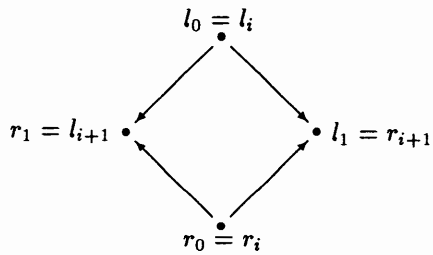


Figure 5.15:

Case 1b: Suppose exactly one of the equalities of equations 5.1 – 5.4 holds. Then part of Γ will have the form given in figure 5.16. No tree of cycle can have this form, so this case can not occur.

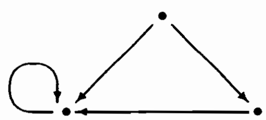


Figure 5.16:

Case 1c: Suppose exactly one of the following four pairs of equations hold:5.1 and 5.2, 5.3 and 5.4, equations 5.3 and 5.2, or 5.1 and 5.4. Then part of Γ will have the form given in figure 5.17. No tree of cycles can have this form, so this case can not occur.

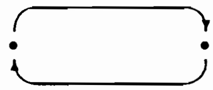


Figure 5.17:

Case 1d: Suppose exactly one of the following two pairs of equations hold: 5.1 and 5.3, 5.2 and 5.4.

Then it must be that Γ is the graph given in figure 5.18. Since $|\Gamma| = 2$ and i must divide $2|\Gamma|$, we know that $i = 2$ or $i = 1$. We need to show that i can not be 1. Suppose it is 1. Then we have $l_0 = l_i = l_1$ and $r_0 = r_i = r_1$. Since l_0 and r_0 are looped vertices, by Corollary 5.5 we have that $x_0 = y_0 =$ up and $x_1 = y_1 =$ down. So in that case $E(M(S_i^*))$ is not isomorphic to $M(S_0)$. We are finished with Case 1.

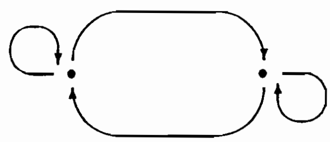


Figure 5.18:

Case 2: $l_i = r_0$ and $r_i = l_1$

Then $l_{i+1} \neq r_1$ while $r_{i+1} = l_1$. So that the set of vertices $\{l_1, r_1\}$ is not the same as the set of vertices $\{l_{i+1}, r_{i+1}\}$. Hence $E(M(S_{i+1}^*))$ is not isomorphic to $M(S)$. Which gives us that $E(M(S_i^*))$ is not isomorphic to $M(S)$.

Case 3: $l_i = R_0$ and $r_i = L_0$

This forces $l_{i+1} \neq r_0$ and $r_{i+1} \neq l_1$. So to have $E(M(S_{i+1})) = M(S_1)$. This in turn forces $l_{i+1} = l_1$ and $r_{i+1} = r_1$. Just as in Case 1 there are four cases to consider. But the cases and the arguments are the same as they were in Case 1.

Case 4: $l_i = r_0$ and $r_i = l_0$

This implies that the walking sequences $l_0, l_1, \ldots, l_{i-1}, r_1, r_2, \ldots, r_i$ and $l_0, l_1, \ldots, l_2|\Gamma|$ are identical. Therefore $i = \frac{2|\Gamma|}{2} = |\Gamma|$. (We note again that this could only happen if v is a nonlooped vertex, or Γ is the degenerate case we are not considering here). ■

For Lemma 5.13 we continue to use the notation that we developed in Lemma 5.12: the second time the label l_0 appears on the Γ-unit circle as we read the sequence

$\{l_i\}_{i=0}^{\infty}$ clockwise we relabel the point L_0. Similarly we define R_0. Thus if we say $l_{|\Gamma|} = r_0$ we mean more than the two vertices are equal we mean that those two labels correspond to the same point on the Γ-unit circle.

Theorem 5.13 *The periodicity of $M(S_0)$ is $|\Gamma|$ precisely when $l_{|\Gamma|} = r_0$ and $r_{|\Gamma|} = l_0$ and the exceptional number is 1. The periodicity of $M(S_0)$ is $2|\Gamma|$ in all other cases.*

Proof. The second statement follows from the first and Lemma 5.12. We prove the first statement. We have already shown that this is true when Γ contains only one looped vertex and when Γ contains exactly two vertices both of which are looped vertices. So we assume that we do not have either of these cases. We first let the exceptional number be 1. Then there is no need to form the graph Γ^*. Our walking algorithm works for string modules over Λ, and we can denote the string-tuples in the minimal projective resolution of $M(S_0)$ without using superscripts.

If the vertex l_0 is the vertex r_0, then by arguments that we made in Lemma 5.11 we have $E(M(S^*_{|\Gamma|})) \simeq M(S_0)$. So assume that the vertex l_0 is not the vertex r_0. Then for the result to hold we need $x_{|\Gamma|} = y_0$ and $y_{|\Gamma|} = x_0$. Since $l_{|\Gamma|} = r_0$ and $r_{|\Gamma|} = l_0$, we know that the point $l_{|\Gamma|+1}$ is the point r_1 and that the point $r_{|\Gamma|+1}$ is the point l_1. Again using similar arguments to those used in the proof of Lemma 5.11 we know that this can only happen if $x_{|\Gamma|} = y_0$ and $y_{|\Gamma|} = x_0$. So we have that if the points $l_{|\Gamma|} = r_0$ and $r_{|\Gamma|} = l_0$ and the exceptional number is 1, the periodicity is $|\Gamma|$.

We now show that if the exceptional number is not 1, the periodicity can not be n. By hypothesis and using Lemma 5.9 we have that the two Γ-walking sequences $l_0, \ldots, l_{2|\Gamma|}$ and $l_0, \ldots, l_{|\Gamma|}, r_1, \ldots, r_{|\Gamma|}$ are identical. Let a be the arrow in Γ of weight $\overline{1}$. Since there is a unique i such that $o(a) = l_i$ and $t(a) = l_{i+1}$, it must be that there is a $k \leq n$ such that either $o(a) = l_k$ and $t(a) = l_{k+1}$ or $o(a) = r_k$ and

$t(a) = l_{k+1}$. Thus the string-tuple for $hS^*_{|\Gamma|}$ has either the form $(x_{|\Gamma|}, r^{\bar{t}}_{|\Gamma|}, y_{|\Gamma|}, l^{\bar{1}}_{|\Gamma|})$ or $(x_{|\Gamma|}, r^{\overline{t+1}}_{|\Gamma|}, y_{|\Gamma|}, l^{\bar{0}}_{|\Gamma|})$. By Lemma 5.8 we know that in both of these cases $E(M(S^*_{|\Gamma|}))$ is not isomorphic to $M(S_0)$. We have the result. ∎

We define a *short* Γ*-walking sequence*, $\{v, v_1, \ldots, v_n, w\}$, to be a Γ-walking sequence in which $v, w \notin \{v_1, \ldots, v_n\}$.

Corollary 5.14 *Let $v \in \Gamma$. Let 1_v be the string S_0 that we have used in the lemmas for this section. Keeping the same notation that we have been using throughout, the periodicity of $M(1_v)$ is $|\Gamma|$ if and only if $|\Gamma_{l_0 l_1}| = |\Gamma_{r_0 r_1}|$ and the exceptional number is 1.*

Proof. By Theorem 5.13 we only need to show that when the exceptional number is 1, $l_0 = r_{|\Gamma|}$ and $r_0 = l_{|\Gamma|}$ if and only if $|\Gamma_{l_0 l_1}| = |\Gamma_{r_0 r_1}|$. Since in this case $l_0 = r_0$, this is equivalent to proving that $l_0 = l_{|\Gamma|}$ and $r_0 = r_{|\Gamma|}$ if and only if $|\Gamma_{l_0 l_1}| = |\Gamma_{r_0 r_1}|$.

Suppose $|\Gamma_{l_0 l_1}| = |\Gamma_{r_0 r_1}|$. Then since $l_0 = r_0$ and $l_1 \neq r_1$, by a simple counting argument it is easy to see that $2|\Gamma_{l_0 l_1}| = |\Gamma| + 1$. It is also not hard to see that the short Γ-walking sequence $\{l_0, l_1, \ldots, l_0\}$ has length $2|\Gamma_{l_0 l_1}| - 1 = |\Gamma|$. Therefore $l_0 = l_{|\Gamma|}$. Similarly we have $r_0 = r_{|\Gamma|}$.

Suppose $l_0 = l_{|\Gamma|}$ and $r_0 = r_{|\Gamma|}$. Since in every Γ-walking sequence of length $2|\Gamma|$ every vertex appears exactly twice, and since we know by previous experience that $l_{2|\Gamma|} = l_0$ experience, we have that $\{l_0, l_1, \ldots, l_{|\Gamma|}\}$ is a short Γ-walking sequence. Therefore it must be that the vertices of that walk are completely contained in $\Gamma_{l_0 l_1}$. In fact the sequence must have each of the vertices of $\Gamma l_0 l_1$ appearing exactly twice. Thus $|\Gamma| = 2|\Gamma_{l_0 l_1}| - 1$. Similarly $|\Gamma| = 2|\Gamma_{r_0 r_1}| - 1$. Thus we have that $|\Gamma_{l_0 l_1}| = |\Gamma_{r_0 r_1}|$. ∎

Chapter 6

A minimal set of generators for Ext(Λ)

In this chapter we construct a minimal set of generators for the cohomology ring or Ext-algebra of a classical Brauer tree algebra with respect to the Jacobson radical of Λ. By *minimal* we mean that there is no subset of the generating set that is also a generating set.

6.1 Definitions and Notations

For each $v \in \Gamma_0$, $M(1_v)$ is a string module. So we can apply the algorithms of Chapter 5 to compute $\mathbf{D}^v$, a minimal projective resolution of $M(1_v)$. We present the notation that we will use to describe $\mathbf{D}^v$ below.

$$\mathbf{D}^v : \cdots \longrightarrow D_2^v \xrightarrow{hS_2^v} D_1^v \xrightarrow{hS_1^v} D_0^v \xrightarrow{hS_0^v} M(1_v) \longrightarrow 0$$

In this chapter unless otherwise stated a bar over a symbol will mean the equivalence class of that element. Though there will be more than one equivalence class appearing in the chapter no confusion should arise. Let $\underline{r}$ be the Jacobson radical of Λ. Then $\underline{r}$ is generated by $\{\overline{a}$: where a is an arrow in $\Gamma.\}$. It is not hard to show that $\Lambda/\underline{r} \simeq \coprod_{v\in\Gamma_0} v\Lambda/_{\underline{r}(v\Lambda)}$, and that for each $v \in \Gamma_0$, $v\Lambda/_{\underline{r}(v\Lambda)} \simeq M(1_v)$. Let $\overline{\Lambda} = \coprod_{v\in\Gamma_0} M(1_v)$. Since the (natural) sum of two minimal projective resolutions is a minimal projective resolution, a minimal projective resolution of $\overline{\Lambda} \simeq \Lambda/\underline{r}$ is $\mathbf{D} = \coprod_{v\in\Gamma_0} \mathbf{D}^v$. In other words, if for $k \geq 0$, we define $d_k = \sum_{v\in\Gamma_0} hS_k^v$ and for $i \geq 0$, we define $D_i = \coprod_{v\in\Gamma_0} D_i^v$, then we have

$$\mathbf{D} : \cdots \longrightarrow D_2 \xrightarrow{d_2} D_1 \xrightarrow{d_1} D_0 \xrightarrow{d_0} \overline{\Lambda} \longrightarrow 0$$

is a minimal projective resolution of $\overline{\Lambda}$.

We are now in a position to define the Ext-algebra of Λ with respect to $\underline{r}$. As an abelian group we define $\mathrm{Ext}(\Lambda) = \coprod_{i=0}^{\infty} \mathrm{Ext}^i(\Lambda/\underline{r}, \Lambda/\underline{r})$. To construct $\mathrm{Ext}^i(\Lambda/\underline{r}, \Lambda/\underline{r})$ we begin with a projective resolution of $\Lambda/\underline{r}$. Since in our case Λ is artinian, we begin with a minimal resolution $\mathbf{D}$. We replace the module $\Lambda/\underline{r}$ in the resolution with the zero module. We then form the complex

$$\mathrm{Hom}(0, \Lambda/\underline{r}) \xrightarrow{d_0^*} \mathrm{Hom}(D_0, \Lambda/\underline{r}) \xrightarrow{d_1^*} \mathrm{Hom}(D_1, \Lambda/\underline{r}) \xrightarrow{d_2^*} \cdots$$

where d_0^* is the zero map and $d_i^*(\alpha) = \alpha \circ d_i$ for $i \geq 1$ and $\alpha \in \mathrm{Hom}(D_i, \Lambda/\underline{r})$. For $i \geq 1$ we define the abelian group $\mathrm{Ext}^i(\Lambda/\underline{r}, \Lambda/\underline{r}) = \mathrm{Ker}\, d_{i+1}^*/_{\mathrm{Im}\, d_i^*}$. If $e \in \mathrm{Ext}^i(\Lambda/\underline{r}, \Lambda/\underline{r})$, then we will say that e is of *degree* i.

Since $\mathbf{D}$ is minimal, and since Λ is artinian, general theory gives us that $\mathrm{Ext}^i(\Lambda/\underline{r}, \Lambda/\underline{r}) \simeq \mathrm{Hom}_\Lambda(D_i, \Lambda/\underline{r})$. Hence $\mathrm{Ext}(\Lambda) \simeq \coprod_{i=0}^{\infty} \mathrm{Hom}(D_i, \Lambda/\underline{r})$. Next we use the Yoneda product to place a multiplicative structure on $\mathrm{Ext}(\Lambda)$. Let $f \in \mathrm{Hom}_\Lambda(D_i, \Lambda/\underline{r})$ and $g \in \mathrm{Hom}_\Lambda(D_j, \Lambda/\underline{r})$. Since the all D_k's are projective, we can find a collection of homomorphisms $\{u_k\}_{i=0}^{\infty}$ that make the diagram of figure 6.1 commute. We call these homomorphisms liftings. We define $f * g = g \circ u_j$. Since

$\{f \in \mathrm{Hom}_\Lambda(D_i, \Lambda/\underline{r}) : i \geq 0\}$ is a basis for $\mathrm{Ext}(\Lambda)$, we can extend this definition linearly to complete the definition of multiplication of any two elements of $\mathrm{Ext}(\Lambda)$.

$$\begin{array}{ccccccccccc} D_{i+j} & \xrightarrow{d_{i+j}} & D_{i+j-1} & \longrightarrow & \cdots & \xrightarrow{d_{i+2}} & D_{i+1} & \xrightarrow{d_{i+1}} & D_i & & \\ u_j \downarrow & & u_{j-1} \downarrow & & & & u_1 \downarrow & & u_0 \downarrow & \overset{f}{\searrow} & \\ D_j & \xrightarrow{d_j} & D_{j-1} & \longrightarrow & \cdots & \xrightarrow{d_2} & D_1 & \xrightarrow{d_1} & D_0 & \xrightarrow{d_0} & \overline{\Lambda} \end{array}$$

Figure 6.1:

6.2 A minimal set of generators

Lemma 6.1 *Let $hS : v_1QUIV \oplus \cdots \oplus v_nQUIV \longrightarrow w_1QUIV \oplus \cdots \oplus w_mQUIV$ be a string function.*

1. *If $QUIV = \Gamma$, then the vertices $v_1, \ldots, v_n$ are distinct.*
2. *If $QUIV = \Gamma^*$ and we do not have both $n = 1$ and $m = 1$, then the vertices $v_1, \ldots, v_n, w_1, \ldots, w_m$ are distinct.*

Proof. Recall that for each of the vertices $v_1, \ldots, v_n, w_1, \ldots, w_m$ there is a corresponding edge in the underlying Brauer tree on which that vertex lies. The edges $v_1, \ldots, v_n, w_1, \ldots, w_m$ are the edges in some connected edge walk from $o(v_1)$ to $t(w_m)$. The edge walk can be found by reading the vertices that appear in the underlying string for hS from left to right. We let $e_1 \Longleftrightarrow e_{n+m}$ represent that edge walk. Since the walk is occurring on a tree, if that edge walk never doubles back on itself, the edges $v_1, \ldots, v_n, w_1, \ldots, w_m$ are clearly unique and we have the result. If the walk doubles back on itself we will have some edge E appearing in the walk twice in a row. This implies that hS contains a formal path, P, lying in one cycle that had the same terminus and origin.

If P lies in a nonexceptional cycle, then P must be a maximal path or an inverse of one. So if P lies in a nonexceptional cycle or is a maximal path in an exceptional cycle, (or an inverse of one), $hS = P$. Therefore we have (1) in the case where P lies in a nonexceptional cycle. Since every cycle in Γ^* is a nonexceptional cycle this constitutes a proof of (2). We now prove (1) in the case where P lies in an exceptional cycle with exceptional number not equal to 1, but is not a maximal path, nor an inverse of one. In this case the vertex E must also lie in the exceptional cycle. By the definition of a string function EEE can not be an edge walk contained in $e_1 \Longleftrightarrow e_{n+m}$ Therefore, since this is a connected edge walk occurring on a tree of edges, if $e_1 \Longleftrightarrow e_{n+m} = e_1 \Longleftrightarrow vEEw \Longleftrightarrow e_{n+m}$, then the edge walks $e_1 \Longleftrightarrow v$ and $w \Longleftrightarrow e_{n+m}$ must not contain any edges (or vertices) that lie in the exceptional cycle. Hence the walks $e_1 \Longleftrightarrow E$ and $E \Longleftrightarrow e_{n+m}$ are both connected edge walks that do not double back on themselves. If an edge x appears in both of those walks, then the unique connected edge walk that doesn't double back on itself from x to E has the same number of edges as the one from E to x. Thus if $x = e_i = e_j$, and if i is odd, j must be even and vice-versa. Therefore, since the vertices $v_1, \ldots, v_n$ appear in $e_1 \Longleftrightarrow e_{n+m}$ as every other edge, they must be distinct. ■

For each $k \geq 0$ and each $v \in \Gamma_0$, we will define $D_k^v = \coprod_{w \in G_k^v} w\Lambda$ where G_k^v is the appropriate set of vertices. If $w \in G_k^v$, then we will say that w appears in D_k^v. We also define an index set $G_k = \{(v,w) : (v,w) \text{ is a formally defined symbol}, v \in \Gamma_0, \text{ and } w \in G_k^v\}$. Using this notation we have that $D_k \simeq \coprod_{(v,w) \in G_k} w\Lambda$. To identify the term sitting in the (v,w) component of D_k we will append the subscript (v,w) to that term.

We are now ready to create a very nice basis for $\mathrm{Ext}(\Lambda)$. Let $v, y \in \Gamma_0$, $w \in G_k^v$, and $x \in G_k^y$, then we define the function $f_k^{(v,w)} : D_k \to \Lambda/\underline{r}$ to be the function such that

$$f_k^{(v,w)}((0,\ldots,0,\overline{w}_{(v,w)},0,\ldots,0)) = (0,\ldots,0,w,0,\ldots,0) \quad \text{where and on the right w is in the } M(w) \text{ component of } \Lambda/\underline{r}\text{, and}$$

$$f_k^{(v,w)}((0,\ldots,0,\overline{x}_{(y,x)},0,\ldots,0)) = 0 \quad \text{where } (y,x) \neq (v,w)$$

By Lemma 6.1 the component $w\Lambda$ appears only once in D_k^v. Therefore, this function is well defined. We say that $f_k^{(v,w)}$ only *nontrivially projects* the vertex w that appears in D_k^v to $\Lambda/\underline{r}$.

Lemma 6.2 *$\{f_k^{(v,w)} : v \in \Gamma_0,\ k \geq 0,$ and $w \in G_k^v\}$ is a K-basis for $Ext(\Lambda)$.*

Proof. Consider the paragraph above and note that all homomorphisms, g, from $v\Lambda$ to $\Lambda/\underline{r}$ are defined as follows $g(\overline{v}) = c(0,\ldots,0,v,0,\ldots,0)$ where v is a vertex in the $M(1_v)$ component of $\Lambda/\underline{r}$ and c is a constant. ■

We call the basis of Lemma 6.2 the *standard basis* and we refer to the vectors in this basis as *standard basis vectors.*

Lemma 6.3 *Let $x, v, w \in \Gamma_0$ with $w \in G_i^x$. Then*

$$f_0^{(v,v)} * f_0^{(w,w)} = \left\{ \begin{array}{ll} f_0^{(v,v)} & \text{if } v = w \\ 0 & \text{otherwise} \end{array} \right\} \text{ and}$$

$$f_0^{(v,v)} * f_i^{(x,w)} = \left\{ \begin{array}{ll} f_i^{(x,w)} & \text{if } x = v \\ 0 & \text{otherwise} \end{array} \right\} \text{ and}$$

$$f_0^{(x,w)} * f_0^{(v,v)} = \left\{ \begin{array}{ll} f_i^{(x,w)} & \text{if } w = v \\ 0 & \text{otherwise} \end{array} \right\}.$$

Proof. This is a good exercise to get the reader acclimated to the notation for subsequent arguments. ■

Lemma 6.4 *Let $g \in Hom_\Lambda(D_j, \Lambda/\underline{r})$. Then a collection of liftings $\{u_r\}_{r=0}^\infty$ for the product $f_i^{(v,w)} * g$ may be chosen so that for $0 \leq k \leq j$, $u_k(D_{i+k}^v)$ sits in the D_k^w component of D_k, and $u_k(D_{i+k}^x) = \{0\}$ for $x \neq v$.*

Proof. The proof is by induction on k. We define $D_{-1} = \overline{\Lambda}$, and for $x \in \Gamma_0$, $D_{-1}^x = M(x)$. We first prove the result for $k = 0$. By definition $f_i^{(v,w)}(D_i^x) = \{0\}$ for $x \neq v$, and $f_i^{(v,w)}(D_i^v)$ sits in the D_{-1}^w component of D_{-1}. Hence u_0 must be chosen so that $d_0(u_0(D_i^x)) = \{0\}$ for $x \neq v$ and $d_0(u_0(D_i^v))$ sits in the D_{-1}^w component of D_{-1}. Therefore, since $d_0 = \sum_{v \in \Gamma_0} hS_0^v$, $hS_0^w(D_0^x) = \{0\}$ for $x \neq w$, and $hS_0^w(D_0^w)$ sits in the D_{-1}^w component of D_{-1}, $u_0(D_i^v)$ is forced to lie in the D_0^w component of D_0, and u_0 may be chosen so that $u_0(D_i^x) = \{0\}$ for $x \neq v$.

We now induct. Assume that we have the result for $k = s$. Then the proof for $k = s+1$ is nearly identical to the one just given for $k = 0$. Simply replace $f_i^{(v,w)}$ in the argument with $d_{i+s+1} \circ u_s$ and adjust the indices accordingly. ■

We call the lifting of Lemma 6.4 a *standard lifting* for the product.

Corollary 6.5 $f_i^{(v,x)} * f_j^{(y,w)} = 0$ *when* $x \neq y$.

Proof. Let $\{u_k\}_{k=0}^\infty$ be a standard lifting for the product. By Lemma, 6.4 $u_j(D_{i+j}^z) = \{0\}$ for $z \neq v$, and $u_j(D_{i+j}^v)$ sits in the D_j^w component of D_j. Therefore, since $f_j^{(y,w)}$ only nontrivially projects the vertex w that appears in D_j^y to $\Lambda/\underline{r}$, when $y \neq w$, we must have $u_j \circ f_j^{(y,w)} = 0$. This gives us the result. ■

Corollary 6.6

Let $\{u_k\}_{k=0}^\infty$ be a standard lifting for the product $f_i^{(v,x)} * f_j^{(x,w)} \in Ext(\Lambda)$. If $u_j(0, \ldots, 0, \overline{w}_{(v,w)}, 0, \ldots, 0) = ($other components$, c\overline{w}_{(x,w)},$ other components$)$ where c is a nonzero constant, then $f_i^{(v,x)} * f_j^{(x,w)} = cf_{i+j}^{(v,w)}$. Otherwise $f_i^{(v,x)} * f_j^{(x,w)} = 0$.

Proof. This follows from Lemma 6.4 and because the only elements not in the kernel of $f_j^{(x,w)}$ must have a scalar multiple of w in the $w\Lambda$ component of D_j^x. ∎

It is well known that Λ is self-injective and of finite representation type and hence $\text{Ext}(\Lambda)$ is finitely generated as an algebra. Therefore we can define H to be a minimal set of generators for $\text{Ext}(\Lambda)$ which is also a subset of the standard basis. We will show that H is unique.

Lemma 6.7 *$f_k^{(v,w)} \notin H$ if and only if $cf_k^{(v,w)} = f_i^{(v,x)} * f_j^{(x,w)}$ for some i, j with $i + j = k$, and $c \neq 0$.*

Proof. This follows from Corollaries 6.5 and 6.6 and because H is a subset of a basis. ∎

Corollary 6.8 *All standard basis vectors in $Hom_\Lambda(D_0, \Lambda/\underline{r})$ and $Hom_\Lambda(D_1, \Lambda/\underline{r})$ are in H.*

Proof. Apply Lemmas 6.3 and 6.7. ∎

Lemma 6.9 *Let $\{u_k\}_{k=0}^{\infty}$ be a standard lifting for $f_i^{(\alpha,\beta)} * f_j^{(\beta,\gamma)} \in Ext(\Lambda)$. For some fixed k, $0 \leq k < j$, suppose $v \in G_{i+k}^{\alpha}$, $w \in G_{i+k+1}^{\alpha}$, $x \in G_k^{\beta}$, $y \in G_{k+1}^{\beta}$, $u_k((0,\ldots,0,v_{(\alpha,v)},0,\ldots,0)) =$ (other components,$(c_1\overline{a})_{(\beta,x)}$,other components), $d_{i+k+1}((0,\ldots,0,w_{(\alpha,w)},0,\ldots,0)) =$ (other components,$(c_2\overline{b})_{(\alpha,v)}$,other components), and $d_{k+1}((0,\ldots,0,y_{(\beta,y)},0,\ldots,0)) =$ (other components,$(c_3\overline{g})_{(\beta,x)}$,other components) where a is a path of any length (including 0) that does not contain a relation, and b, g are paths of length greater than 0. Also assume that v, w, x, y, a, b, g all lie in the same cycle, and c_1, c_2, c_3 are constants not equal to 0.*

1. If ab is a path that does not contain a maximal path, then $u_{k+1}((0,\dots,0,w_{(\alpha,w)},0,\dots,0)) =$ (other components,$(c_4\overline{h})_{(\beta,y)}$,other components) where h is the path such that $gh = ab$, and c_4 is the constant such that $c_3c_4 = c_1c_2$.

Proof. We have $u_k(d_{i+k+1}((0,\dots,0,v_{(\alpha,w)},0,\dots,0))) =$ (other components,$(c_1c_2\overline{ab}+\overline{P})_{(\beta,x)}$, other components) where P is a path not in the same cycle as ab. Also, since hS^{β}_{k+1} is a string function d_{k+1} maps at most two components of D^{β}_{k+1} to the $x\Lambda$ component of D^{β}_{k}. However, only one of those (possibly) two components, $z\Lambda$, is such that z is in the same cycle as ab. That component is the $y\Lambda$ component. Therefore, since ab does not contain a maximal path, and since $d_{k+1}((0,\dots,0,y_{y_\beta},0,\dots,0)) =$ (other components,$(c_3\overline{g})_{x_\beta}$,other components), we must have $u_{k+1}((0,\dots,0,w_{w_\alpha},0,\dots,0)) =$ (other components,$(c_4\overline{h})_{y_\beta}$,other components), where h is a path such $gh = ab$, and c_4 is a constant such that $c_3c_4 = c_1c_2$.
■

Let hS be a string function. Then we will say that the vertices $o(S)$ and $t(S)$ are the end vertices of hS. We will refer to all other vertices, w, appearing in the underlying string for hS, and with $v\Lambda$ appearing in the domain for hS as middle vertices of hS. We note that we will allow a vertex that appears twice in the underlying string to be both a middle vertex and an end vertex. However, we will say that an end vertex, v, is an upper end vertex if $v\Lambda$ appears in the domain for hS. You will note that by Lemma 6.1 upper end vertices are never middle vertices. Note also that both of the cycles that a middle vertex lies within must have length at least 2. By the *length of a cycle* we mean the number of arrows in the cycle if the cycle is not exceptional and $N\cdot$(the number of arrows in the cycle) otherwise.

Figure 6.2 we supply as a reference for Theorem 6.10 and Lemma 6.11. We never refer to it and simply intend for the reader to use it as a thinking device.

$$\begin{array}{ccccccc} D_k & \xrightarrow{d_k} & D_{k-1} & \xrightarrow{d_{k-1}} & D_{k-2} & & \\ \downarrow u_2 & & \downarrow u_1 & & \downarrow u_0 & \searrow f_{k-2}^{(v,w)} & \\ D_2 & \xrightarrow{d_2} & D_1 & \xrightarrow{d_1} & D_0 & \xrightarrow{d_0} & \overline{\Lambda} \end{array}$$

Figure 6.2:

For the following lemma we let the *length of a cycle* be the number of arrows in the cycle when the cycle is not the exceptional cycle, and $N\cdot$(the number of arrows in the cycle) when the cycle is the exceptional cycle with exceptional number N.

Theorem 6.10 *Let $v \in G_2^v$ with v a middle vertex of hS_2^v. Then $f_2^{(v,v)} \notin H$ if and only if one of the two cycles that v sits in has length two.*

Proof. Since v is a middle vertex of hS_2^v, each cycle that v sits in must have length greater than 1. Let a and b be the two arrows with origin v. Let a' and b' be the paths (not vertices) such that aa' and bb' are maximal paths. Using this notation we draw part of hS_1^v and hS_2^v in figure 6.3.

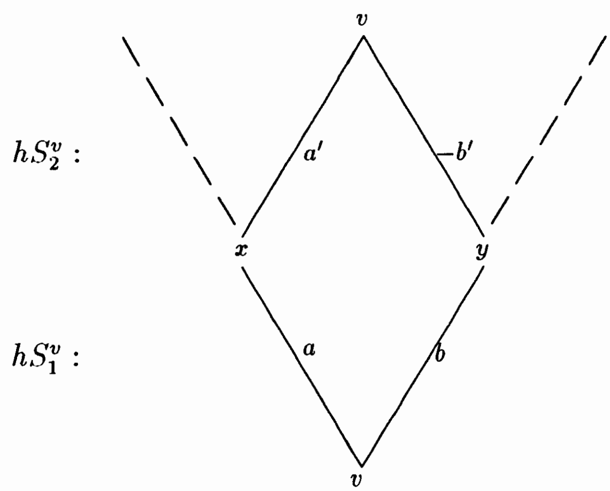


Figure 6.3:

To argue both directions of the 'if and only if' statement we will use the following fact. By Lemmas 6.3 and 6.7 $f_2^{(v,v)} \notin H$ if and only if $f_2^{(v,v)} = cf_1^{(v,w)} * f_1^{(w,v)}$ for some $w \in \Gamma_0$ and some constant $c \neq 0$. Without loss of generality we first assume that the cycle containing aa' has length two. Then a' is an arrow and we have figure 6.4.

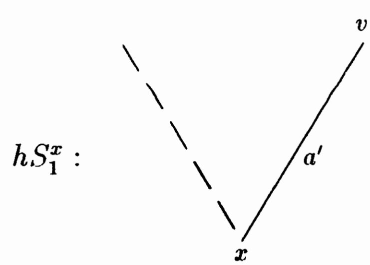


Figure 6.4:

Let $\{u_k\}_{k=0}^{\infty}$ be a standard lifting for $f_1^{(v,x)} * f_1^{(x,v)}$.
Since $u_0((0,\ldots,0,x_{(v,x)},0,\ldots,0)) = (0,\ldots,0,x_{(x,x)},0,\ldots,0)$, by Lemma 6.9 $u_1((0,\ldots,0,v_{(v,v)},0,\ldots,0)) = (0,\ldots,0,v_{(x,v)},0,\ldots,0)$. Hence by Corollary 6.4 $f_1^{(v,x)} * f_1^{(x,v)} = f_2^{(v,v)}$. This gives us that $f_2^{(v,v)} \notin H$.

We now assume that both cycles which v sits in have length greater than two. Let α and β be the first arrows of the paths a' and b', and let A and B be the paths such that $\alpha A = a'$ and $\beta B = b'$. Using this notation we give hS_1^x and hS_1^y in figure 6.5.

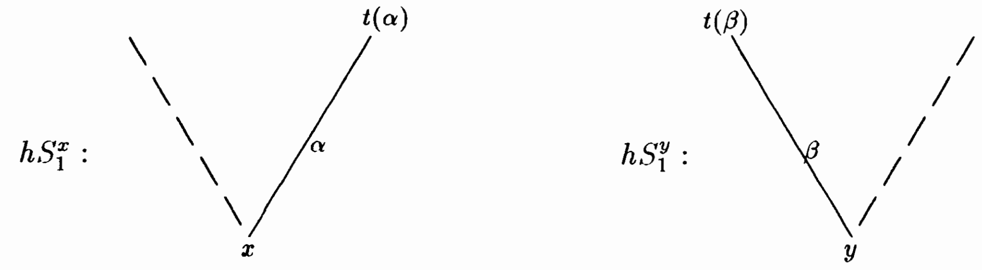


Figure 6.5:

Therefore, $f_2^{(v,v)}$ will be in H unless either $f_2^{(v,v)} = cf_1^{(v,x)} * f_1^{(x,v)}$ or $f_2^{(v,v)} = cf_1^{(v,y)} * f_1^{(y,v)}$ where $c \neq 0$. We show that the first equality is not possible, and symmetry gives us that the second equality is not possible. Let $\{u_k\}_{k=0}^{\infty}$ be a standard lifting for $f_1^{(v,x)} * f_1^{(x,v)}$. Since $u_0((0,\ldots,0,x_{(v,x)},0,\ldots,0) = (0,\ldots,0,x_{(x,x)},0,\ldots,0)$, by Lemma 6.9 we have $u_1((0,\ldots,0,v_{(v,v)},0,\ldots,0) =$ (other components, $A_{(x,v)}$, other components). Thus by Corollary 6.4 we have $f_1^{(v,x)} * f_1^{(x,v)} = 0$. ■

Lemma 6.11 *For $k > 2$, if $w \in G_k^v$, where w is a middle vertex of hS_k^v, then $f_k^{(v,w)} = f_{k-2}^{(v,w)} * f_2^{(w,w)}$. Hence $f_k^{(v,w)} \notin H$.*

Proof. We first address the existence of $f_2(w,w)$ and $f_{k-2}^{(v,w)}$. Since w is a middle vertex, a quick study of its resolution using Corollary 5.1 establishes the existence of $f_2^{(w,w)}$. Similarly, Corollary 5.1 can be used to see that if the middle vertex w appears in D_k^v, it must also have appeared in D_{k-2}^v. Thus $f_{k-2}^{(v,w)}$ must exist.

In figure 6.6 we illustrate a portion of the string functions $hS^v_{k-1}, hS^v_k, hS^w_1, hS^w_2$. There we are letting P, Q be paths of length greater than zero in the same cycles as the arrows a and b respectively. P', Q', a', and b' will be the paths such that PP', QQ', aa', and bb' are maximal paths. Also we let $c = \pm 1$. As you consider the figure recall that if we choose all of the coefficients in the underlying string for the first homomorphism of a resolution to be $+1$, then the coefficients of all subsequent underlying strings are either all $+1$'s or alternate between $+1$ and -1.

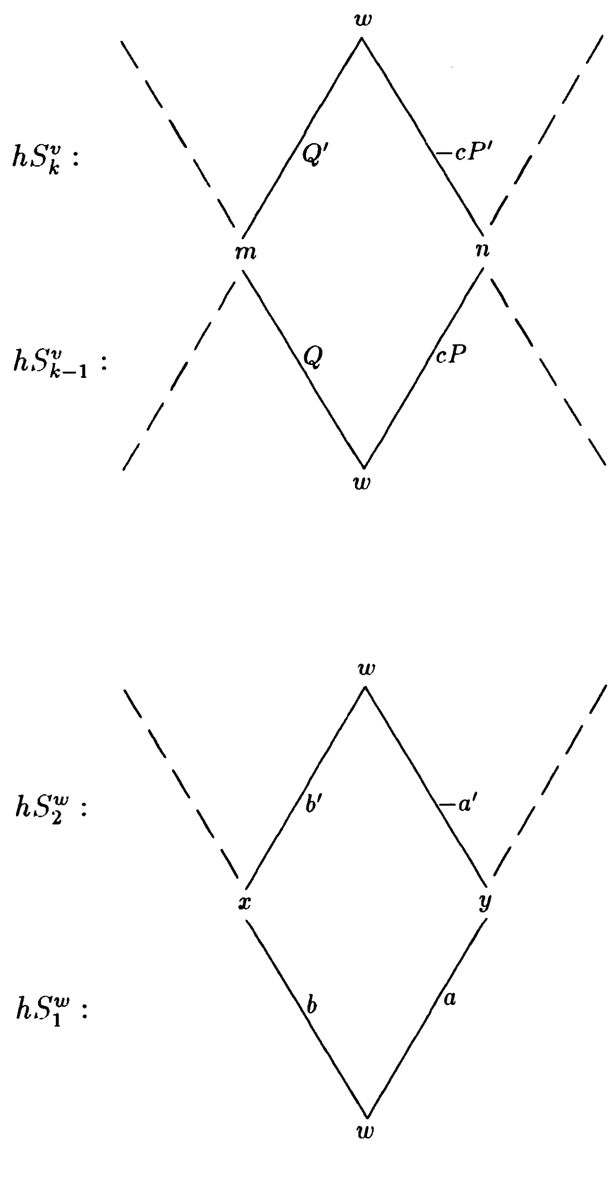


Figure 6.6:

From figure 6.6 we see that

$$\begin{aligned}
d_1((0,\ldots,0,\overline{x}_{(w,x)},0,\ldots,0)) &= (0,\ldots,0,\overline{b}_{(w,w)},0,\ldots,0),\\
d_2((0,\ldots,0,\overline{w}_{(w,w)},0,\ldots,0)) &= (0,\ldots,0,\overline{b}'_{(w,x)},-\overline{a}'_{(w,y)},0,\ldots,0),\\
d_{k-1}((0,\ldots,0,\overline{m}_{(v,m)},0,\ldots,0)) &= (\text{other components},\overline{Q}_{(v,w)},\text{other components})\\
&\text{and}\\
d_k((0,\ldots,0,\overline{w}_{(v,w)},0,\ldots,0)) &= (0,\ldots,0,\overline{Q}'_{(v,m)},-c\overline{P}'_{(v,n)},0,\ldots,0).
\end{aligned}$$

Note that $u_0((0,\ldots,0,\overline{w}_{(v,w)},0,\ldots,0)) = (0,\ldots,0,\overline{w}_{(w,w)},0,\ldots,0)$, and apply Lemma 6.9 to see that $u_1((0,\ldots,0,\overline{m}_{(v,m)},0,\ldots,0)) = (\text{other components},\overline{B}_{(w,x)},\text{other components})$ where B is the path such that $bB = wQ = Q$. Since $BQ' = b'$, we can reapply Lemma 6.9 to see that $u_1((0,\ldots,0,\overline{w}_{(v,w)},0,\ldots,0)) = (\text{other components},\overline{w}_{(w,w)},\text{other components})$. By Corollary 6.4 we have the result. ■

Following Lemma 6.11, the only standard basis vectors whose presence in H have not be determined algorithmically are of the form $f_k^{(v,w)}$ where w is an upper end vertex of hS_k^v for $k \geq 2$. Those algorithmic conditions are given in Corollary 6.19. Actually the conditions given in that Corollary are for determining when a standard basis vector of the form $f_k^{(l_0,l_k)}$ (or $f_k^{(r_0,r_k)}$) is in H where $(x_m,l_m,y_m,r_m)_{m=0}^{\infty}$ is a walking sequence for $M(1_{l_0})$ and $x_m = \text{up}$ (or $y_m = \text{up}$). However those are exactly the vectors of the form $f_k^{(v,w)}$ where w is an upper end vertex of hS_k^v. To obtain Corollary 6.19 we are going to construct algorithmic conditions for determining when the standard basis vectors of the form $f_k^{(l_0,l_k)}$ (or $f_k^{(r_0,r_k)}$) are elements of the minimal generating set H^* for $\mathrm{Ext}(\Lambda^*)$ where $(x_m,l_m,y_m,r_m)_{m=0}^{\infty}$ is a walking sequence for $M(1_{l_0}) \in \mathrm{Mod}(\Lambda^*)$ and $x_k = \text{up}$ (or $y_k = \text{up}$). Lemma 6.12 justifies this approach. From Lemma 6.13 to Theorem 6.18 we are are going to leave off the superscripts for the vertices of Γ^* when no confusion can arrise. For that portion of the paper all standard basis vectors will be assumed to be in $\mathrm{Ext}(\Lambda^*)$.

Lemma 6.12 *Let $w \in G_k^v \subseteq \Gamma$, $x \in G_i^v \subseteq \Gamma$, and $w \in G_j^x \subseteq \Gamma$. Then $cf_k^{(v,w)} = f_i^{(v,x)} * f_j^{(x,w)}$ if and only if $cf_k^{(v^{\overline{0}},w^{\overline{t_1}})} = f_i^{(v^{\overline{0}},x^{\overline{t_2}})} * f_j^{(x^{\overline{t_2}},w^{\overline{t_1}})}$ where $\overline{t_1}$ and $\overline{t_2}$ are the unique elements of $\boldsymbol{Z}_N$ such that $w^{\overline{t}_1} \in G_k^{v^{\overline{0}}} \subseteq \Gamma^*$ and $x^{\overline{t}_2} \in G_i^{v^{\overline{0}}} \subseteq \Gamma^*$,*

Proof. Note that by Lemma 6.1 $\overline{t}_1$ and $\overline{t}_2$ are unique. Let $\{hS_m^v\}_{m=0}^{\infty}$ and $\{hS_m^{v^{\overline{0}}}\}_{m=0}^{\infty}$ be the string functions in a minimal projective resolution of $M(1_v) \in \mathrm{Mod}(\Lambda)$ and $M(1_{v^{\overline{0}}}) \in \mathrm{Mod}(\Lambda^*)$ respectively. Let $\{u_m\}_{m=0}^{\infty}$ and $\{u_m^*\}_{m=0}^{\infty}$ be standard liftings for $f_i^{(v,x)} * f_j^{(x,w)}$ and $f_i^{(v^{\overline{0}},x^{\overline{t_2}})} * f_j^{(x^{\overline{t}_2},w^{\overline{t_1}})}$ respectively.

By the proof of Lemma 3.3 if we disregard the superscripts in a string $S_m^{v^{\overline{0}}}$, then the strings $S_m^{v^{\overline{0}}}$ and S_m^v are the same. Similarly if we disregard the superscripts in the h-string $hS_m^{v^{\overline{0}}}$, then the h-strings $hS_m^{v^{\overline{0}}}$ and hS_m^v are the same. Since by Section 3 $E(M(S_m^{v^{\overline{0}}})) = M(S_m^v)$ for all v and m, u_m^* and u_m will be the same if we disregard superscripts, and that is the essence of the proof. ■

Lemma 6.13 *Let $\{(x_m, l_m, y_m, r_m)\}_{m=0}^{\infty}$ be a walking sequence for $M(1_{l_0}) \in$ Mod(Λ^*).*

1. *l_s is not in $l_{s-1} \leftrightarrow r_{s-1}$ if and only if r_{s-1} is not in $\Gamma^*_{l_s l_{s+1}}$.*

2. *r_s is not in $l_{s-1} \leftrightarrow r_{s-1}$ if and only if l_{s-1} is not in $\Gamma^*_{r_s r_{s+1}}$.*

Proof. Since $l_{s-1} \leftrightarrow r_{s-1}$ is an edge walk on the underlying Brauer tree, and since it never doubles back on itself, it is clear from the diagram in figure 6.7 that the vertex $r_{s-1} \in \Gamma^*_{l_s l_{s+1}}$ if and only if $l_s \in l_{s-1} \leftrightarrow r_{s-1}$. The proof of the second statement follows in a similar fashion. ■

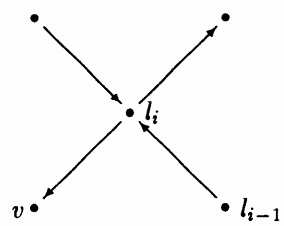


Figure 6.7:

Recall that a *short* Γ^**-walking sequence* $\{v, v_1, \ldots, v_n, w\}$ is a Γ^*-walking sequence in which $v, w \notin \{v_1, \ldots, v_n\}$.

Lemma 6.14 *In the construction of the walking sequence,* $\{(x_m, l_m, y_m, r_m)\}_{m=0}^{\infty}$, *for* $M(1_{l_0}) \in Mod(\Lambda^*)$ *we may choose* $x_s = up$ *if and only if* $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_s l_{s+1}}|$. *Similarly we may choose* $y_s = up$ *if and only if* $|\Gamma^*_{r_0 r_1}| \geq |\Gamma^*_{r_s r_{s+1}}|$.

Proof. We prove the first statement. The second statement can be proven similarly. By Corollary 5.6, whenever $l_s = r_s$, we may choose x_s = up. Using the definition of a short Γ^*-walking sequence, one can see that $l_s = r_s$ if and only if the short Γ^*-walking sequence $\{l_s, l_{s+1}, \ldots, l_s\}$ and $\{l_s, l_{s+1}, \ldots, r_s\}$ have the same length. However, the length of $\{l_s, l_{s+1}, \ldots, r_s\}$ is the same as the length of $\{l_0, l_1, \ldots, r_0\}$ which is $|\Gamma^*_{l_0 l_1}|$, and the length of $\{l_s, l_{s+1}, \ldots, l_s\}$ is $|\Gamma^* l_s l_{s+1}$. Therefore we have that $l_s = r_s$ if and only if $|\Gamma^*_{l_0 l_1}| = |\Gamma^*_{l_s l_{s+1}}|$.

Suppose $l_s \neq r_s$. Then by Corollary 5.6 x_s = up if and only if l_s does not appear in $l_{s-1} \leftrightarrow r_{s-1}$. Using Lemma 6.13 we then have x_s = up if and only if r_{s-1} is not in $\Gamma^*_{l_s l_{s+1}}$. In addition, r_{s-1} not appearing in $\Gamma^*_{l_s l_{s+1}}$ is equivalent to r_{s-1} not appearing in the short Γ^*-walking sequence $\{l_s, l_{s+1}, \ldots, l_s\}$. However, r_{s-1} is not in the sequence $\{l_s, l_{s+1}, \ldots, l_s\}$ if and only if the short Γ^*-walking sequence

$\{l_s, l_{s+1}, \ldots, r_{s-1}\}$ has length greater than the length of $\{l_s, l_{s+1}, \ldots, l_s\}$ which is $2|\Gamma^*_{l_s l_{s+1}}|$. Since $\{l_s, l_{s+1}, \ldots, r_{s-1}\}$ has the same length as $\{l_2, l_3, \ldots, r_0, r_1\}$ which is $2|\Gamma^*_{l_0 l_1}| - 1$, we have $x_s = $ up if and only if $2|\Gamma^*_{l_0 l_1}| - 1 > 2|\Gamma^*_{l_s l_{s+1}}|$. Hence, $x_s = $ up if and only if $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_s l_{s+1}}|$.

Combining our results when $l_s = r_s$ and those we just obtained when $l_s \neq r_s$, we have that we may choose $x_s = $ up if and only if $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_s l_{s+1}}|$. ■

Corollary 6.15 *If we keep the same notation as in Lemma 6.14, and assume that* $|\Gamma^*_{l_0 l_1}| < |\Gamma^*_{l_s l_{s+1}}|$, *then* $f_s^{(l_0, l_s)} \notin H^*$. *In the same way we have that if* $|\Gamma^*_{r_0 r_1}| < |\Gamma^*_{r_s r_{s+1}}|$ *then* $f_s^{(r_0, r_s)} \notin H^*$.

Proof. The result follows immediately from Lemma 6.14. ■

Lemma 6.16 *Let* $v \in \Gamma^*$. *Let* $\{(x_m, l_m, y_m, r_m)\}_{m=0}^{\infty}$ *be a walking sequence for* $M(1_v) = M(1_{l_0}) \in Mod(\Lambda^*)$. *Let* $k > 1$, $i + j = k$, *and let* c *be a nonzero scalar. If* $l_k \neq r_k$ *and* $cf_k^{(v, l_k)} = f_i^{(v,x)} * f_j^{(x, l_k)}$ *and if* $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ *is a walking sequence for* $M(x)$, *then*

1. *We may choose* $\hat{l}_j = \hat{l}_k$ *with* $\hat{x}_j = up$,

2. $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_i l i+1}|$,

3. $|\Gamma^*_{l_i l_{i+1}}| \geq |\Gamma^*_{l_k l k+1}|$.

If $l_k \neq r_k$ *and* $cf_k^{(v, r_k)} = f_i^{(v,x)} * f_j^{(x, r_k)}$ *and if* $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ *is a walking sequence for* $M(x)$, *then*

1. *We may choose* $\hat{r}_j = \hat{r}_k$ *with* $\hat{x}_j = up$,

2. $|\Gamma^*_{r_0 r_1}| \geq |\Gamma^*_{r_i r i+1}|$,

3. $|\Gamma^*_{r_i r_{i+1}}| \geq |\Gamma^*_{r_k r k+1}|$.

Proof. We prove the first set of statements. The other set can be proven using symmetry. Let $\{u_m\}_{m=0}^{\infty}$ be a standard lifting for the product $f_i^{(v,x)} * f_j^{(x,l_k)}$. We first prove (1). For ease of notation we let $w = l_k$. Since the product $f_i^{(v,x)} * f_j^{(x,l_k)} = cf_k^{(v,w)} \neq 0$, we know by Corollary 6.6 that $u_j((0,\dots,0,\overline{w}_{(v,w)},0,\dots,0)) =$ (other components, $c\overline{w}_{(x,w)}$, other components). Since l_k is an upper end vertex of hS_k^v, we know that $d_k((0,\dots,0,\overline{w}_{(v,w)},0,\dots,0)) = (0,\dots,0,c_1\overline{a}_{(v,o(a))},0,\dots,0)$ for some nonzero scalar c_1 and path a. Hence $u_{j-1}(d_k((0,\dots,0,\overline{w}_{(v,w)},0,\dots,0))) = (0,\dots,0,c_2\overline{ab}_{(v,o(a))},0,\dots,0)$ for some nonzero scalar c_2 and some path b. If w is a middle vertex of hS_j^x, then we know $d_j(u_j((0,\dots,0,\overline{w}_{(x,w)},0,\dots,0)) = (0,\dots,0,c_3\overline{A}_{(x,o(A))},c_4\overline{B}_{(x,o(B))},0,\dots,0)$ where c_3,c_4 are nonzero scalars, and A and B are paths that do not contain relations, i.e., $\overline{A},\overline{B} \neq 0$. This leads us to the contradiction $d_j(u_j((0,\dots,0,\overline{w}_{(v,w)},0,\dots,0))) \neq u_{j-1}(d_k((0,\dots,0,\overline{w}_{(v,w)},0,\dots,0)))$. Thus we can choose $\hat{l}_j = w$ with $\hat{x}_j =$ up.

We now prove (2) and (3). Using (1) we now choose the walking sequence, $\{(\hat{x}_m,\hat{l}_m,\hat{y}_m,\hat{r}_m)\}_{m=0}^{\infty}$, for $M(x)$ so that $\hat{l}_j = w = l_{i+j} = l_k$. And $\hat{x}_j = up$. We will show that the walking sequence $\{(\hat{x}_m,\hat{l}_m,\hat{y}_m,\hat{r}_m)\}_{m=0}^{\infty}$ can in fact be chosen so that the sequences $\{\hat{l}_m\}_{m=0}^{\infty}$ and $\{l_{i+m}\}_{m=0}^{\infty}$ are identical with $x_j =$ up. Since they are walking sequences, it suffices to show that the sequences $\{\hat{l}_{j-1},\hat{l}_j\}$ and $\{l_{j-1},l_j\}$ can be chosen to be identical.

In figure 6.8 we draw part of the underlying strings for hS_{i+j}^v and hS_j^x we do this by appealing to the following facts: $w = l_k$ and $x_k =$ up, we may choose $\hat{l}_j = w$ with $\hat{x}_j =$ up, while $u_j((0,\dots,0,\overline{w}_{(v,w)},0,\dots,0)) =$ (other components, $c\overline{w}_{(x,w)}$, other components) by Corollary 6.6. In the figure c_5 and c_6 are nonzero scalars. Since $l_k \neq r+k$, P is not a maximal path. If Q is not a maximal path, then by Lemma 6.9 P and Q must lie in the came cycle. Hence, by Corollary 5.1 l_{i+j-1} (resp. $\hat{l}_{j-1}$) must be the unique vertex such that l_{i+j} (resp. $\hat{l}_j$) is one arrow away from l_{i+j-1} (resp. $\hat{l}_{j-1}$) in the same cycle as P (resp. Q). Since P and Q are in the same cycle and since $\hat{l}_j = l_{i+j}$, we have $\hat{l}_{j-1} = l_{i+j-1}$. Therefore the walking sequence $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ can be chosen so that $\{\hat{l}_m\}_{m=0}^{j} = \{l_{i+m}\}_{m=0}^{j}$ are identical.

As an immediate consequence we have that $l_i = \hat{l}_0 = x$. By Lemma 6.1 the vertices appearing in an underlying string for a module in Mod(Γ^*) are distinct. Therefore, since $f_i^{(v,x)}$ is defined, we know that $x_i =$ up. By Lemma 6.14 we conclude that $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_i l_{i+1}}|$. We have proven (2).

We had shown earlier that $\hat{x}_j =$ up. So again using that $\{\hat{l}_m\}_{m=0}^{j} = \{l_{i+m}\}_{m=0}^{j}$ are identical and Lemma 6.14 we have that $|\Gamma^*_{l_i l_{i+1}}||\Gamma^*_{\hat{l}_0 \hat{l}_1}| \geq |\Gamma^*_{\hat{j}_i \hat{j}_{i+1}}| = |\Gamma^*_{k_i k_{i+1}}|$. We have now proven (3).

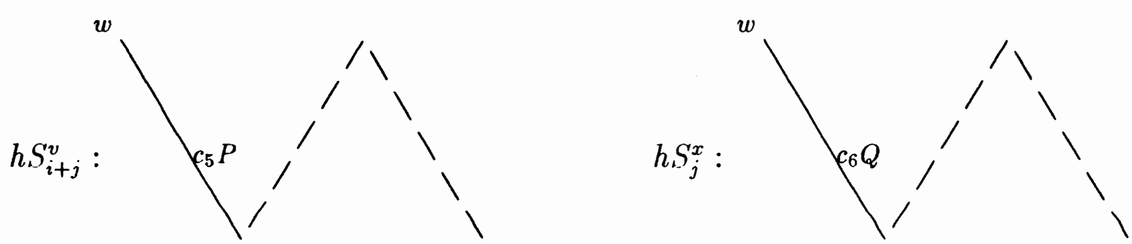


Figure 6.8:

■

Lemma 6.17 *Let $v \in \Gamma^*$ and let $\{(x_m, l_m, y_m, r_m)\}_{m=0}^{\infty}$ be a Γ^*-walking sequence for $M(1_v) \in$ Mod(Λ^*).*

1. *If in accordance with Corollary 5.6 we can choose $x_k =$ up and for some $i < k$, we can choose $x_i =$ up. Let $j = k - i$. Choose $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ be the*

walking sequence for $M(1_{l_i})$ *such that* $\{\hat{l}_m\}_{m=0}^{\infty} = \{l_{i+m}\}_{m=0}^{\infty}$. *If in accordance with Corollary 5.6 we can choose the sequences* $\{\hat{x}_m\}_{m=0}^{\infty}$ *and* $\{x_{i+m}\}_{m=0}^{\infty}$ *to be identical with* $x_i = x_k = up$*, then for some nonzero scalar* c *we have* $cf_k^{(l_0,l_k)} = f_i^{(l_0,l_i)} * f_j^{(\hat{l}_0,\hat{l}_j)} = f_i^{(l_0,l_i)} * f_j^{(l_i,l_k)}$.

2. *If in accordance with Corollary 5.6 we can choose* $y_k = up$ *and for some* $i < k$*, we can choose* $y_i = up$*. Let* $j = k - i$*. Choose* $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ *be the walking sequence for* $M(r_i)$ *such that* $\{\hat{r}_m\}_{m=0}^{\infty} = \{r_{i+m}\}_{m=0}^{\infty}$. *If in accordance with Corollary 5.6 we can choose the sequences* $\{\hat{y}_m\}_{m=0}^{\infty}$ *and* $\{y_{i+m}\}_{m=0}^{\infty}$ *to be identical with* $y_i = y_k = up$*, then for some nonzero scalar* c *we have* $cf_k^{(r_0,r_k)} = f_i^{(r_0,r_i)} * f_j^{(\hat{r}_0,\hat{r}_j)} = f_i^{(r_0,r_i)} * f_j^{(r_i,r_k)}$.

Proof. We prove (1). (2) follows using symmetry. For notations sake we let $w = \hat{l}_0 = l_i$. By Corollary 6.6 we only need to show that $u_j((0,\ldots,0,\overline{l_{k(v,l_k)}},0,\ldots,0)) =$ (other components, $c\overline{l_{k(w,l_k)}}$, other components). Using induction on m we will prove the following more general statement. For $0 \le m \le j$ if $x_{i+m} =$ up, then $u_m((0,\ldots,0,\overline{l_{i+m(v,l_{i+m})}},0,\ldots,0)) =$
(other components, $c_m\overline{l_{i+m(w,l_{i+m})}}$, other components), and if $x_{i+m} =$ down, then $u_{m-1}((0,\ldots,0,\overline{l_{i+m(v,l_{i+m})}},0,\ldots,0)) =$
(other components, $c_{m-1}\overline{l_{i+m(w,l_{i+m})}}$, other components). We know from experience that $u_0((0,\ldots,0,\overline{l_{i(v,l_i)}},0,\ldots,0)) = (0,\ldots,0,\overline{l_{i(w,l_i}},0,\ldots,0)$. Since $x_i =$ up, we have that the general statement when $m = 0$. We now assume that it is true for $m = t$, $0 \le t < j$.

We consider four cases, and, in each case, we appeal to figures 5.1 – 5.10 of Corollary 5.1 to create parts of the underlying strings hS_{i+t}^v,hS_{i+t+1}^v,hS_t^w, and hS_{t+1}^w that occur in that case. We draw these in figures 6.9 – 6.12. Figure 6.9 represents the case where x_{i+t} = up and x_{i+t+1} = down. Figure 6.10 represents the case where x_{i+t} = up and x_{i+t+1} = up. Figure 6.9 represents the case where x_{i+t} = down and x_{i+t+1} = down. Figure 6.9 represents the case where x_{i+t} = down and x_{i+t+1} = up. In the first case the inductive assumption gives us the result immediately. In the next two cases only one application of Lemma 6.9 gives us the result. In the last case we need to apply Lemma 6.9 twice in order to obtain the result. We will only write down the proof for the last case.

In this case since x_{t+i} = down, we know that
$u_{t-1}((0,\ldots,0,\overline{l_{(t+i)}}_{(v,l_{t+i})},0,\ldots,0)) =$
(other components, $c_t\overline{l_{(t+i)}}_{(w,l_{t+i})}$, other components). By Lemma 6.9 we have that
$u_t((0,\ldots,0,\overline{y}_{(v,y)},0,\ldots,0)) =$
(other components, $k_3\overline{B}_{(w,z)}$, other components). where k_3 is some nonzero scalar and B is the path such that $P = QB$. Note that $(BP')a = (Q')a$. Hence, we can reapply Lemma 6.9 to see that $u_{t+i+1}((0,\ldots,0,\overline{l}_{(t+i+1)(v,l_{t+i+1})},0,\ldots,0)) =$ (other components, $c_{t+i+1}\overline{l}_{(t+i+1)(x,l_{t+i+1})}$, other components). Since x_{t+i+1} = up, we have proven the inductive step of the argument.

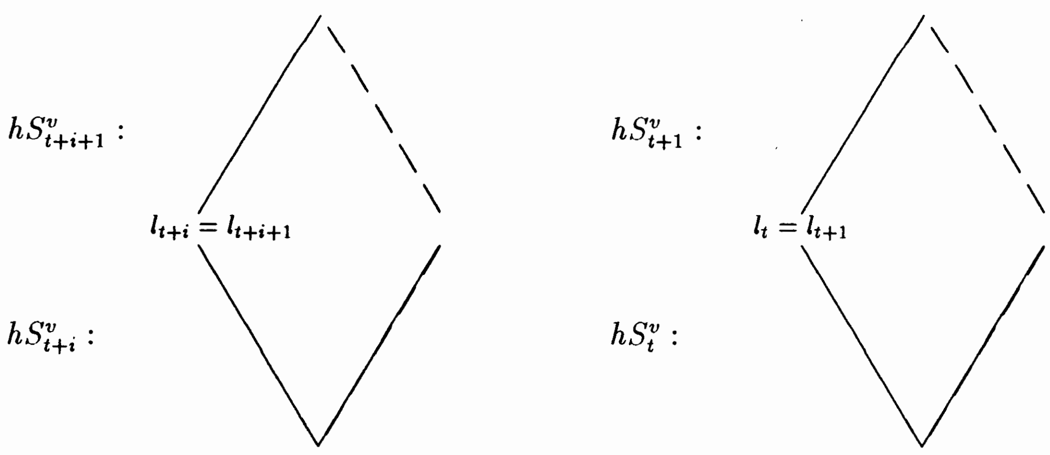


Figure 6.9:

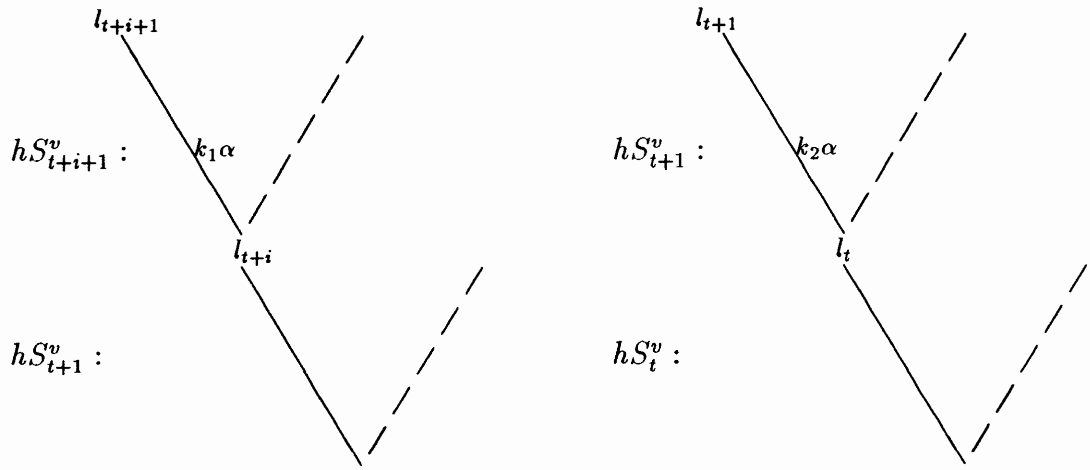


Figure 6.10: We let k_1, k_2 be nonzero scalars, and let α be an arrow.

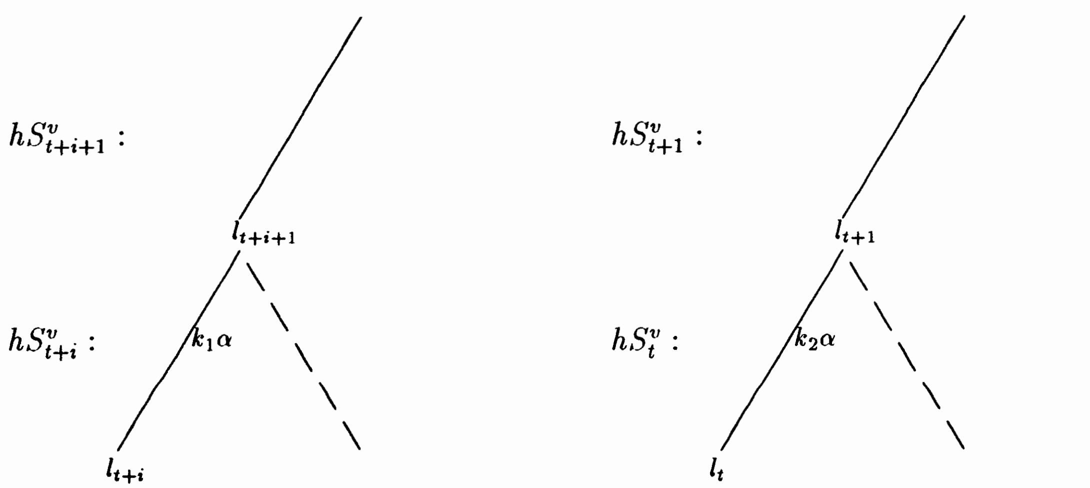


Figure 6.11: We let k_1, k_2 be nonzero scalars, and let α be an arrow.

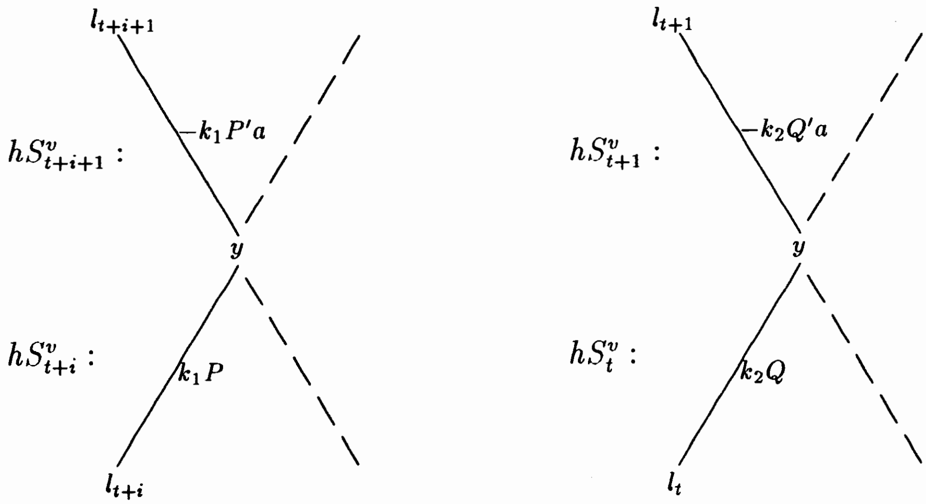


Figure 6.12: k_1, k_2 are nonzero scalars, and P and Q are paths lying in the same cycle which do not contain any relations. a is the first arrow of P, and P' and Q' are the paths (or vertices) such that PP' and QQ' are maximal paths.

■

Theorem 6.18 *Let* $\{(x_m, l_m, y_m, r_m)\}_{m=0}^{\infty}$ *be a walking sequence for* $M(1_{l_0}) \in$ *Mod*(Γ^*).

1. $f_k^{(l_0,l_k)} \notin H^*$ *if and only if* $|\Gamma^*_{l_0l_1}| < |\Gamma^*_{l_kl_{k+1}}|$ *or there is an* i, $0 < i < k$ *such that* $|\Gamma^*_{l_0l_1}| \geq |\Gamma^*_{l_il_{i+1}}| \geq |\Gamma^*_{l_kl_{k+1}}|$.

2. $f_k^{(r_0,r_k)} \notin H^*$ *if and only if* $|\Gamma^*_{r_0r_1}| < |\Gamma^*_{r_kr_{k+1}}|$ *or there is an* i, $0 < i < k$ *such that* $|\Gamma^*_{r_0r_1}| \geq |\Gamma^*_{r_ir_{i+1}}| \geq |\Gamma^*_{r_kr_{k+1}}|$.

Proof. We will prove (1). (2) follows from symmetry. Suppose $|\Gamma^*_{l_0l_1}| < |\Gamma^*_{l_kl_{k+1}}|$. Then by Corollary 6.15 $f_k^{(l_0,l_k)} \notin H$. Suppose, on the other hand, that there is an i, $0 < i < k$ such that $|\Gamma^*_{l_0l_1}| \geq |\Gamma^*_{l_il_{i+1}}| \geq |\Gamma^*_{l_kl_{k+1}}|$. Then we can find a t, $0 < t < k$ such that $|\Gamma^*_{l_0l_1}| \geq |\Gamma^*_{l_tl_{t+1}}| \geq |\Gamma^*_{l_jl_{j+1}}|$ for all j where $0 < j \leq k$, and $|\Gamma^*_{l_0l_1}| \geq |\Gamma^*_{l_jl_{j+1}}|$. Let $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ be the walking sequence for $M(1_{l_t})$ such that $\{\hat{l}_m\}_{m=0}^{\infty}$ and

$\{\hat{l}_{m+t}\}_{m=0}^{\infty}$ are the the same sequence. By Lemma 6.17 we can show that $f_k^{(l_0,l_k)} \notin H^*$ by showing that we can choose the sequences $\{\hat{x}_a\}_{a=0}^{k-t}$ and $\{x_{t+a}\}_{a=0}^{k-t}$ to be the same. Let $0 < a \leq k-t$. By our choice of t we have $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_{t+a} l_{t+a+1}}|$ if and only if $|\Gamma^*_{l_t l_{t+1}}| \geq |\Gamma^*_{l_{t+a} l_{t+a+1}}|$. Therefore we have $|\Gamma^*_{\hat{l}_0 \hat{l}_1}| \geq |\Gamma^*_{\hat{l}_a \hat{l}_{a+1}}|$ if and only if $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_{t+a} l_{t+a+1}}|$. Hence by Lemma 6.14, for $0 < a \leq k-t$, we can choose $x_{t+a} =$ up if and only if we can choose $\hat{x}_a =$ up. In addition since $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_k l_{k+1}}|$, we can choose $\hat{x}_{k-t} = x_k =$ up. Since $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_t l_{t+1}}|$, we can also choose $x_t = \hat{x}_0 =$ up. Finally, since we can always choose $\hat{x}_0 =$ up, we have $f_k^{(l_0,l_k)} \notin H*$. We have one direction of (1).

Suppose $f_k^{(l_0,l_1)} \notin H^*$ and $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_k l_{k+1}}|$. If $l_k \neq r_k$, then by Lemma 6.16 we have for some i, $0 < i < k$, $|\Gamma^*_{l_0 l_1}| \geq |\Gamma^*_{l_i l i+1}| \geq |\Gamma^*_{l_k l_{k+1}}|$. So suppose $l_k = r_k$, and that $cf_k^{(l_0,l_k)} = f_i^{(l_0,x)} * f_j^{(x,l_k)}$. Then by (1) of Lemma 6.16 we may choose $\hat{l}_j = l_k$ with $\hat{x}_j =$ up. Therefore partial underlying strings for hS_j^x, and hS_k^v and hS_j^x are as given in figure 6.8 with the dotted portions ommitted. By Lemma 6.9, since $u_j((0,\ldots,0,\overline{l_{k(l_0,l_k)}},0,\ldots,0)) = (0,\ldots,0,c\overline{l_{k(x,l_k)}},0,\ldots,0)$, and since P is maximal, we have Q is maximal. Thus $hS_k^v = hS_j^x$ and hence $hS_{i+t}^v = hS_t^x$ for $0 \leq t \leq j$. Thus the walking sequence for $M(x)$, $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}$, can be chosen so that the sequences $\{(\hat{x}_m, \hat{l}_m, \hat{y}_m, \hat{r}_m)\}_{m=0}^{\infty}$ and $\{(x_{m+i}, l_{m+i}, y_{m+i}, r_{m+i})\}_{m=0}^{\infty}$ are the same with $l_i = \hat{l}_0 = \hat{r}_0 = r_i$. By the proof of Lemma 6.14, since $l_i = r_i$ and $l_k = r_k$, we have $|\Gamma^*_{l_0 l_1}| = |\Gamma^*_{l_i l_{i+1}}|$ and $|\Gamma^*_{l_0 l_1}| = |\Gamma^*_{l_k l_{k+1}}|$. Thus we also have the result when $l_k = r_k$. ■

Let N be the exceptional number for Γ. We define

$$\|\Gamma_{vw}\| = \begin{cases} |\Gamma_{vw}| \text{ if } \Gamma_{vw} \text{ does not contain the exceptional cycle} \\ |\Gamma_{vw}| + (N-1)|\Gamma| \text{ otherwise.} \end{cases}$$

Corollary 6.19 *Let* $\{x_m, l_m^{\bar{a}_m}, y_m, r_m^{\bar{b}_m}\}_{m=0}^{\infty}$ *be a walking sequence for* $M(1_{l_0^{\bar{a}_m}}) \in$ *Mod*(Λ^*). *Then*

1. $f_k^{(l_0,l_k)} \notin H \subseteq$ *Ext*(Λ) *if and only if* $|\Gamma_{l_0l_1}| \geq |\Gamma_{l_kl_{k+1}}|$ *or there is an* i, $0 < i < k$ *such that*

$$\|\Gamma_{l_0l_1}\| \geq \|\Gamma_{l_il_{i+1}}\| \geq \|\Gamma_{l_kl_{k+1}}\|.$$

2. $f_k^{(r_0,r_k)} \notin H \subseteq$ *Ext*(Λ) *if and only if* $|\Gamma_{l_0l_1}| \geq |\Gamma_{l_kl_{k+1}}|$ *or there is an* i, $0 < i < k$ *such that*

$$\|\Gamma_{r_0r_1}\| \geq \|\Gamma_{r_ir_{i+1}}\| \geq \|\Gamma_{r_kr_{k+1}}\|.$$

Proof. It follows from the definition of $\|\Gamma_{l_ml_{m+1}}\|$ that $\|\Gamma_{l_ml_{m+1}}\| = |\Gamma^*_{l_m^{\bar{a}_m} l_{m+1}^{\bar{a}_{m+1}}}|$. Thus by Theorem 6.18 and Lemma 6.12, we have (1). (2) follows similarly. ∎

Corollary 6.19 is stronger than it appears at first. It seems to imply that we need to calculate a Γ^*-walking sequence for $M(1_{l_0^{m_0}})$. However, that is unnecessary. We can derive the sequences $\{l_m\}_{m=0}^{\infty}$ and $\{r_m\}_{m=0}^{\infty}$ without calculating the sequences $\{l_m^{\bar{a}_m}\}_{m=0}^{\infty}$ and $\{r_m^{\bar{b}_m}\}_{m=0}^{\infty}$.

Let d_0 be a Λ-string function corresponding to the h-string S_0 where S_0 is a maximal path with origin and terminus l_0. Let S_1 be the h-string corresponding to d_1 the string function acquired by applying the picture algorithm to d_0. Then let $\hat{x}_1 = o(S_1)$ (the origin of S_1). Similarly let $\hat{y}_1 = t(S_1)$. Our choice for these two vertices may be the reverse from the choices that we would have made had we gone to Λ^* for our calculations. However, the symmetry makes this irrelevant. The reader should review Lemma 5.3.

Since Λ^*-walking sequences and Λ-walking sequences are identical when we ignore the superscripts, the Λ-walking sequences $\{\hat{x}_m\}_{m=0}^{\infty}$ and $\{\hat{y}_m\}_{m=0}^{\infty}$ are the walking sequences $\{x_m\}_{m=0}^{\infty}$ and $\{y_m\}_{m=0}^{\infty}$.

6.3 Summary and Examples

We can view $\mathrm{Ext}(\Lambda)$ as a quotient of a path algebra, $K\Upsilon$. Below we describe how to use Theorem 6.10 and Corollary 6.19 to construct Υ. It is clear from Corollary 6.6 that the generators of the form $f_k^{(v,w)}$, for $k \geq 1$, behave like arrows from a verex v to a vertex w, and that the generators $f_0^{(v,v)}$ behave like a vertex v. Even though we have used the label v in the graph Γ we will use it again without confusion in the graph Υ. We will use the label $f_k^{(v,w)}$ for the arrow from the vertex v to the vertex w corresponding to $f_k^{(v,w)}$.

It is clear from Corollary 6.8 that, for each vertex $v \in \Gamma$, $f_0^{(v,v)} \in H$. Hence, each of the vertices appearing in Γ will also appear in Υ. Using Theorem 6.10 we now decide which of the vertices v will be an origin for an arrow of the form $f_2^{(v,v)}$. We do this by considering the lengths of the two cycles in which v lies in Γ. According to the theorem the arrow $f_2^{(v,v)}$ must be included in Υ whenever both of the cycles have length greater than two. Recall that for the theorem we defined the length of a cycle to be the number of arrows in the cycle unless the cycle is the exceptional cycle. In that case, the length of the cycle is $N\cdot$(the number of arrows in the cycle) where N is the exceptional number.

Let v $(= l_0 = r_0)$ be a vertex in Γ. Let $\{l_i\}_{i=0}^{\infty}$ and $\{r_i\}_{i=0}^{\infty}$ be the two Γ walking sequences that begin with v. Then we now need to use Corollary 6.19 to determine when an arrow $f_k^{(l_0,l_k)}$ or an arrow $f_k^{(r_0,r_k)}$ is to be included in Υ. We know by Lemma 6.11 that after doing this for each vertex $v \in \Gamma$ we will have completed the construction of the graph of Υ. We warn the reader that the algorithm may yield the following: $f_k^{(v,l_k)} \in H$ and $f_k^{(v,r_k)} \in H$ where $l_k = r_k$. This does not mean that there will be two arrows of degree k from v to l_k. The notation for the generators was well defined. It simply means that $f_k^{(v,l_k)} = f_k^{(v,r_k)}$.

The reader will be pleased to note that for any Γ-walking sequences $\{l_i\}_{i=0}^{\infty}$, $l_{2|\Gamma|+k} = l_k$. Thus $|\Gamma_{l_k l_{k+1}}| = |\Gamma_{l_{2|\Gamma|+k} l_{2|\Gamma|+k+1}}$. So by Corollary 6.19 $f_{2|\Gamma|+k}^{(l_0,l_{2|\Gamma|+k})} \notin H$ for $k \geq 1$. Thus the algorithm terminates rather quickly.

Example: In figure 6.13 we give a graph Γ and in figure 6.14 we give the constructed graph Υ. We give the reader that one of the Γ-walking sequences for v is $vvwwxyyx$. We also list the sizes of the partial graphs in the order in which they should be considered for determining the arrows leaving the vertex v using that walk.

$$\begin{aligned}
&\|\Gamma_{vv}\| = 1 + 2|\Gamma| = 1 + 2 \cdot 4 = 9 \\
&\|\Gamma_{vw}\| = 4 \\
&\|\Gamma_{ww}\| = 1 \\
&\|\Gamma_{wx}\| = 4 + 2|\Gamma| = 12 \\
&\|\Gamma_{xy}\| = 2 \\
&\|\Gamma_{yy}\| = 1 \\
&\|\Gamma_{yx}\| = 12 \\
&\|\Gamma_{xv}\| = 11
\end{aligned}$$

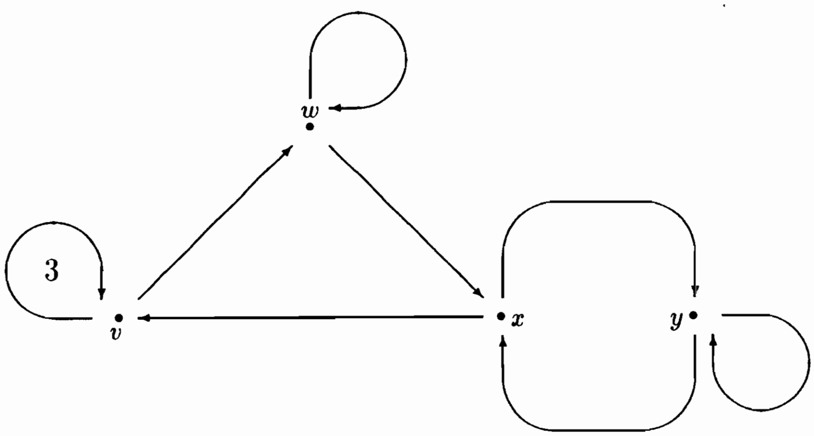


Figure 6.13: Γ

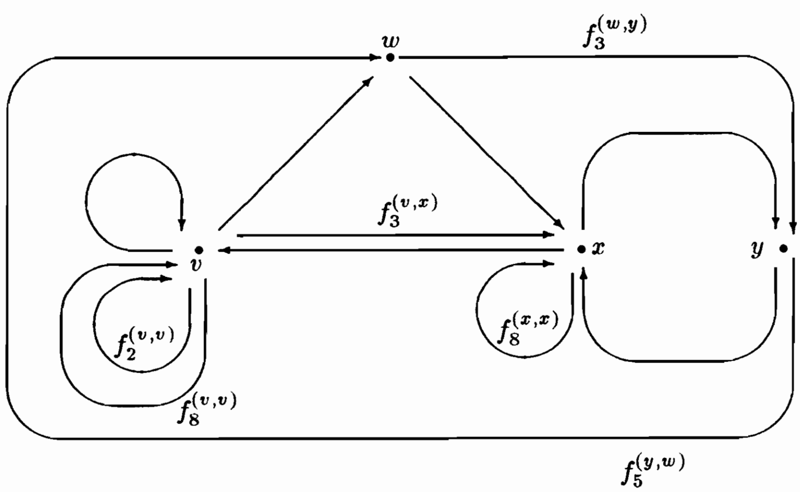


Figure 6.14: Υ

Bibliography

[1] J. Alperin, G. Janusz, *Resolutions and periodicity* , Proc. Amer. Math. Soc. **37** (1973) 403–406.

[2] R. Brauer, *Investigations on group characters*, Ann. of Math. **42** (1941) 936–958.

[3] P. Brown, *The Ext-algebra of a representation-finite biserial algebra*, preprint

[4] M.C.R. Butler, C.M. Ringel, *Auslander–Reiten Sequences with Few Middle Terms and Applications to String Algebras*, J. of Alg. **76 no.1** (1982) 138–152.

[5] E. Dade, *Blocks with cyclic defect groups* , Ann. of Math. **84** (1966) 20–48.

[6] P. Gabriel, *Indecomposable Representations II*, Symposia Mathematica Soc. **11** (1973) 81–104. New York-London: Academic Press 1973

[7] E. Green, *Graphs with Relations, Coverings and Group-Graded Algebras*, Trans. American Math. Soc., **279** (1983), 297–310.

[8] E. Green, *Group-Graded Algebras and the Zero Relation Problem.*

[9] E. Green, R. Gordon, *Representation Theory of Graded Artin Algebras*, Comm. in Alg. **15 (1 and 2)** (1987) 145–179.

[10] E. Green, D. Zacharia, *The cohomology ring of a monomial algebra* , Manuscripta Mathematica **85** (1994) 11–23.

[11] J.A. Green, *Walking around the Brauer tree* , J. Austral. Math. Soc. Ser. A **17** (1974) 197–213.

[12] D. Higman, *Indecomposable representations at characteristic p*, Duke Math. J. **21** (1954) 377–381.

[13] G.J. Janusz, *Indecomposable modules for finite groups* , Ann. of Math. **89** (1969) 209–241.

[14] F. Kasch, M. Kneser, H. Kupisch, *Unzerlegbare modulare Darstellungen endlicher Gruppen mit zyklischer p-Sylow-Gruppe* , Arch. Math. (Basel) **8** (1957) 320–321.

[15] H. Meltzer, A. Skowroński, *Group algebras of finite representation type*, Mathematische Zeitschrift **182 (no. 1)** (1983) 129–148.

[16] Y. Ogawa, *Ext for blocks with cyclic defect groups*, J. of Alg. **91** (191984) 328–333.

[17] I. Reiten, *Almost split sequences for group algebras of finite representation type*, Trans. Amer. Math. Soc. **233** (1977) 125–136.

Vita

Lee Andrew Chasen was born in Atlantic City, New Jersey in 1967 where he spent the first fourteen years of his life. At the age of fourteen he moved to Elizabethville, PA. There he graduated from Upper Dauphin High School in 1985. In 1989, he graduated from Bloomsburg University of Bloomsburg, PA with a B.S. in mathematics. Lee then went to Virginia Polytechnic Institute and State University where he received his Ph.D. in mathematics in 1995.

Lee Andrew Chasen