Brauer Trees, Quivers, and Projective Indecomposable Modules

In this GAP/QPA project, we will discuss Brauer trees, the associated quivers, and the methods for recovering the projective indecomposable modules. We reference Zimmerman's text, *Representation Theory: A Homological Algebra Point of View.* Suppose that G has a block B with normal cyclic defect group D. Then, from Corollary 2.12.3, we know that each projective indecomposable B-module is uniserial. Let Λ be the Brauer tree for the block B. We will construct a quiver from the Brauer tree, then use a set of relations to define a path algebra. The projective indecomposable modules for this path algebra are exactly these uniserial projective indecomposable B-modules. Given a Brauer tree, it is easy to define a quiver in GAP, and using QPA, we are able to determine the PIM's.

It is well know that the "building blocks" of a Brauer tree for finite simple groups are lines and stars. Thus, we provide the GAP code for determining the PIM's from a star-shaped Brauer tree with five edges, as well as a line of length three. First, we examine the star-shaped Brauer tree. Labeling the center node as "1", choosing any outside vertex as "2" and continuing clockwise, we obtain a labeling on the set of vertices and an orientation on the edges:

$$(1,2) \to (1,3) \to (1,4) \to (1,5) \to (1,6) \to (1,2)$$

This yields the quiver as required, and thus, as evidenced by the following GAP code, the PIM's:

```
gap> LoadPackage("qpa");


Loading  GBNP 1.0.1 (Non-commutative Grbner bases)
by A.M. Cohen (http://www.win.tue.nl/~amc) and
   J.W. Knopper (J.W.Knopper@tue.nl).
Homepage: http://mathdox.org/products/gbnp/
```

```
Loading  QPA 1.23 (Quivers and Path Algebras)
by Edward Green (http://www.math.vt.edu/people/green) and
   Oeyvind Solberg (http://www.math.ntnu.no/~oyvinso/).
Homepage: http://www.math.ntnu.no/~oyvinso/QPA/

true
gap> Q:=Quiver(5, [[1,2,"a"],[2,3,"b"],[3,4,"c"],[4,5,"d"],[5,1,"e"]]);
<quiver with 5 vertices and 5 arrows>
gap> F:=GF(5);
GF(5)
gap> FQ:=PathAlgebra(F,Q);
<GF(5)[<quiver with 5 vertices and 5 arrows>]>
gap> gens:=GeneratorsOfAlgebra(FQ);
[ (Z(5)^0)*v1, (Z(5)^0)*v2, (Z(5)^0)*v3, (Z(5)^0)*v4, (Z(5)^0)*v5, (Z(5)^0)*a, (Z(5)^0)*b, (Z(5
gap> a:=gens[6];
(Z(5)^0)*a
gap> b:=gens[7];
(Z(5)^0)*b
gap> c:=gens[8];
(Z(5)^0)*c
gap> d:=gens[9];
(Z(5)^0)*d
gap> e:=gens[10];
(Z(5)^0)*e
```

```
gap> relations:=[a*b*c*d*e*a, b*c*d*e*a*b, c*d*e*a*b*c, d*e*a*b*c*d, e*a*b*c*d*e];

[ (Z(5)^0)*a*b*c*d*e*a, (Z(5)^0)*b*c*d*e*a*b, (Z(5)^0)*c*d*e*a*b*c, (Z(5)^0)*d*e*a*b*c*d, (Z(5

gap> gb:=GBNPGroebnerBasis(relations, FQ);

[ (Z(5)^0)*a*b*c*d*e*a, (Z(5)^0)*b*c*d*e*a*b, (Z(5)^0)*c*d*e*a*b*c, (Z(5)^0)*d*e*a*b*c*d, (Z(5

gap> A:=FQ/gb;

<GF(5)[<quiver with 5 vertices and 5 arrows>]/<two-sided ideal in <GF(5)[<quiver with 5 vertic

gap> Dimension(A);

30

gap> IndecProjectiveModules(A);

[ <[ 2, 1, 1, 1, 1 ]>, <[ 1, 2, 1, 1, 1 ]>, <[ 1, 1, 2, 1, 1 ]>, <[ 1, 1, 1, 2, 1 ]>, <[ 1, 1,
```

Now, we examine a line-shaped Brauer tree with four vertices. Labeling the left-most vertex as "1", and proceeding to the right, we obtain a set of vertices and the following maps between edges

$$(1,2) \to (2,3) \to (1,2) \qquad \text{and} \qquad (2,3) \to (3,4) \to (2,3).$$

This yields the quiver as required, and thus, as evidenced by the following GAP code, the PIM's:

```
gap> Q:=Quiver(3, [[1,2,"a"],[2,1,"b"],[2,3,"c"],[3,2,"d"]]);

<quiver with 3 vertices and 4 arrows>

gap> F:=GF(5);

GF(5)

gap> FQ:=PathAlgebra(F,Q);

<GF(5)[<quiver with 3 vertices and 4 arrows>]>

gap> gens:=GeneratorsOfPathAlgebra(FQ);

Error, Variable: 'GeneratorsOfPathAlgebra' must have a value

not in any function at line 6 of *stdin*

gap> gens:=GeneratorsOfAlgebra(FQ);
```

```
[ (Z(5)^0)*v1, (Z(5)^0)*v2, (Z(5)^0)*v3, (Z(5)^0)*a, (Z(5)^0)*b, (Z(5)^0)*c,
  (Z(5)^0)*d ]
gap> a:=gens[4];
(Z(5)^0)*a
gap> b:=gens[5];
(Z(5)^0)*b
gap> c:=gens[6];
(Z(5)^0)*c
gap> d:=gens[7];
(Z(5)^0)*d
gap> relations:=[a*b*a, d*c*d, a*c, d*b,b*a-c*d];
[ (Z(5)^0)*a*b*a, (Z(5)^0)*d*c*d, (Z(5)^0)*a*c, (Z(5)^0)*d*b,
  (Z(5)^0)*b*a+(Z(5)^2)*c*d ]
gap> gb:=GBNPGroebnerBasis(relations, FQ);
[ (Z(5)^0)*a*c, (Z(5)^2)*b*a+(Z(5)^0)*c*d, (Z(5)^0)*d*b, (Z(5)^0)*a*b*a,
  (Z(5)^0)*b*a*b ]
gap> A:=FQ/gb;
<GF(5)[<quiver with 3 vertices and 4 arrows>]/
<two-sided ideal in <GF(5)[<quiver with 3 vertices and 4 arrows>]>,
  (5 generators)>>
gap> Dimension(A);
10
gap> IndecProjectiveModules(A);
```