COMPUTATIONAL REPRESENTATION THEORY – LECTURE V

Gerhard Hiss

Lehrstuhl D für Mathematik
RWTH Aachen University

Group Theory and Computational Methods
ICTS-TIFR, Bangalore, 05 – 14 November 2016

Contents

Notation

Throughout this lecture, let F be a field and $\mathfrak{A}$ a finite-dimensional F-algebra.

$J(\mathfrak{A})$: Jacobson radical of $\mathfrak{A}$
i.e. the annihilator of the simple $\mathfrak{A}$-modules
i.e. the intersection of the maximal right ideals of $\mathfrak{A}$

mod-$\mathfrak{A}$: category of finite-dimensional right $\mathfrak{A}$-modules

Presentations for Algebras

$F\langle X_1, \ldots, X_n \rangle$: free associative F-algebra in $X_1, \ldots, X_n$

Presentations for Algebras

$F\langle X_1, \ldots, X_n\rangle$: free associative F-algebra in $X_1, \ldots, X_n$

For $R \subset F\langle X_1, \ldots, X_n\rangle$ write

$$\langle X_1, \ldots, X_n \mid R \rangle := F\langle X_1, \ldots, X_n\rangle / I,$$

where I is the two-sided ideal generated by R.

Presentations for Algebras

$F\langle X_1, \ldots, X_n \rangle$: free associative F-algebra in $X_1, \ldots, X_n$

For $R \subset F\langle X_1, \ldots, X_n \rangle$ write

$$\langle X_1, \ldots, X_n \mid R \rangle := F\langle X_1, \ldots, X_n \rangle / I,$$

where I is the two-sided ideal generated by R.

Example: $\langle X_1, X_2 \mid X_1^2, X_2^2, X_1X_2 - X_2X_1 \rangle \cong F(C_2 \times C_2)$.

Presentations for Algebras

$F\langle X_1, \ldots, X_n \rangle$: free associative F-algebra in $X_1, \ldots, X_n$

For $R \subset F\langle X_1, \ldots, X_n \rangle$ write

$$\langle X_1, \ldots, X_n \mid R \rangle := F\langle X_1, \ldots, X_n \rangle / I,$$

where I is the two-sided ideal generated by R.

Example: $\langle X_1, X_2 \mid X_1^2, X_2^2, X_1X_2 - X_2X_1 \rangle \cong F(C_2 \times C_2)$.

$\mathfrak{A}$ is finitely presented if $\mathfrak{A} \cong \langle X_1, \ldots, X_n \mid R \rangle$ for some finite R.

Generators and Relations for Matrix Algebras

Suppose that F is finite, $\text{char}(F) = p$, and let $\mathfrak{A} \leq F^{d \times d}$ be a matrix algebra generated by $A_1, \ldots, A_l$.

Generators and Relations for Matrix Algebras

Suppose that F is finite, $\text{char}(F) = p$, and let $\mathfrak{A} \leq F^{d \times d}$ be a matrix algebra generated by $A_1, \ldots, A_l$.

Carlson and Matthews have developed and implemented an algorithm that computes

1. a finite presentation for $\mathfrak{A}$,

Generators and Relations for Matrix Algebras

Suppose that F is finite, $\mathrm{char}(F) = p$, and let $\mathfrak{A} \leq F^{d \times d}$ be a matrix algebra generated by $A_1, \ldots, A_l$.

Carlson and Matthews have developed and implemented an algorithm that computes

1. a finite presentation for $\mathfrak{A}$,
2. a matrix algebra isomorphic to the basic algebra of $\mathfrak{A}$,

GENERATORS AND RELATIONS FOR MATRIX ALGEBRAS

Suppose that F is finite, $\mathrm{char}(F) = p$, and let $\mathfrak{A} \leq F^{d \times d}$ be a matrix algebra generated by $A_1, \ldots, A_l$.

Carlson and Matthews have developed and implemented an algorithm that computes

1. a finite presentation for $\mathfrak{A}$,
2. a matrix algebra isomorphic to the basic algebra of $\mathfrak{A}$,
3. the Cartan matrix and the dimension of $\mathfrak{A}$.

GENERATORS AND RELATIONS FOR MATRIX ALGEBRAS

Suppose that F is finite, $\mathrm{char}(F) = p$, and let $\mathfrak{A} \leq F^{d\times d}$ be a matrix algebra generated by $A_1, \ldots, A_l$.

Carlson and Matthews have developed and implemented an algorithm that computes

1. a finite presentation for $\mathfrak{A}$,
2. a matrix algebra isomorphic to the basic algebra of $\mathfrak{A}$,
3. the Cartan matrix and the dimension of $\mathfrak{A}$.

Applications: Homomorphisms from $\mathfrak{A}$, cohomology, see also Lecture 4.

THE CARLSON-MATTHEWS ALGORITHM: BACKGROUND

Let $S_1, \ldots, S_r$ denote the simple $\mathfrak{A}$-modules (up to isomorphism).

The Carlson-Matthews Algorithm: Background

Let $S_1, \ldots, S_r$ denote the simple $\mathfrak{A}$-modules (up to isomorphism).

$\mathfrak{A}/J(\mathfrak{A}) \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$ with homogeneous components $\mathfrak{A}_i$;

The Carlson-Matthews Algorithm: Background

Let $S_1, \ldots, S_r$ denote the simple $\mathfrak{A}$-modules (up to isomorphism).

$\mathfrak{A}/J(\mathfrak{A}) \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$ with homogeneous components $\mathfrak{A}_i$; the $\mathfrak{A}_i$ are full matrix algebras over finite extension fields K_i of F.

The Carlson-Matthews Algorithm: Background

Let $S_1, \ldots, S_r$ denote the simple $\mathfrak{A}$-modules (up to isomorphism).

$\mathfrak{A}/J(\mathfrak{A}) \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$ with homogeneous components $\mathfrak{A}_i$; the $\mathfrak{A}_i$ are full matrix algebras over finite extension fields K_i of F.

In fact $\mathfrak{A}_i$ is the image of the action homomorphism

$$\varphi_i : \mathfrak{A} \to \mathrm{End}_F(S_i).$$

$\mathfrak{A}_i$, φ_i and K_i are constructed with the MeatAxe.

THE CARLSON-MATTHEWS ALGORITHM: BACKGROUND

Let $S_1, \ldots, S_r$ denote the simple $\mathfrak{A}$-modules (up to isomorphism).

$\mathfrak{A}/J(\mathfrak{A}) \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$ with homogeneous components $\mathfrak{A}_i$; the $\mathfrak{A}_i$ are full matrix algebras over finite extension fields K_i of F.

In fact $\mathfrak{A}_i$ is the image of the action homomorphism

$$\varphi_i : \mathfrak{A} \to \mathrm{End}_F(S_i).$$

$\mathfrak{A}_i$, φ_i and K_i are constructed with the MeatAxe.

There is a subalgebra $\mathfrak{A}'$ of $\mathfrak{A}$ with $\mathfrak{A}' \cap J(\mathfrak{A}) = 0$, so that $\mathfrak{A}' \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$.

THE CARLSON-MATTHEWS ALGORITHM: BACKGROUND

Let $S_1, \ldots, S_r$ denote the simple $\mathfrak{A}$-modules (up to isomorphism).

$\mathfrak{A}/J(\mathfrak{A}) \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$ with homogeneous components $\mathfrak{A}_i$; the $\mathfrak{A}_i$ are full matrix algebras over finite extension fields K_i of F.

In fact $\mathfrak{A}_i$ is the image of the action homomorphism

$$\varphi_i : \mathfrak{A} \to \mathrm{End}_F(S_i).$$

$\mathfrak{A}_i$, φ_i and K_i are constructed with the MeatAxe.

There is a subalgebra $\mathfrak{A}'$ of $\mathfrak{A}$ with $\mathfrak{A}' \cap J(\mathfrak{A}) = 0$, so that $\mathfrak{A}' \cong \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_r$.

This subalgebra is also constructed during the algorithm.

THE CARLSON-MATTHEWS ALGORITHM: OUTLINE

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.

THE CARLSON-MATTHEWS ALGORITHM: OUTLINE

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.
2. For each i, compute, with the MeatAxe, a sequence e_{ij} of pairwise orthogonal primitive idempotents with $E_i = \sum_j e_{ij}$.

THE CARLSON-MATTHEWS ALGORITHM: OUTLINE

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.
2. For each i, compute, with the MeatAxe, a sequence e_{ij} of pairwise orthogonal primitive idempotents with $E_i = \sum_j e_{ij}$.
3. Construct elements $\beta_i \in e_{i1}\,\mathfrak{A}\,e_{i1}$, τ_i in $E_i \mathfrak{A} E_i$ such that $\langle \beta_i, \tau_i \rangle \cong \mathfrak{A}_i$; this gives generators for $\mathfrak{A}'$.

THE CARLSON-MATTHEWS ALGORITHM: OUTLINE

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.
2. For each i, compute, with the MeatAxe, a sequence e_{ij} of pairwise orthogonal primitive idempotents with $E_i = \sum_j e_{ij}$.
3. Construct elements $\beta_i \in e_{i1}\,\mathfrak{A}\,e_{i1}$, τ_i in $E_i \mathfrak{A} E_i$ such that $\langle \beta_i, \tau_i \rangle \cong \mathfrak{A}_i$; this gives generators for $\mathfrak{A}'$.
4. Determine ideal generators for $J(\mathfrak{A})$.

THE CARLSON-MATTHEWS ALGORITHM: OUTLINE

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.
2. For each i, compute, with the MeatAxe, a sequence e_{ij} of pairwise orthogonal primitive idempotents with $E_i = \sum_j e_{ij}$.
3. Construct elements $\beta_i \in e_{i1}\,\mathfrak{A}\,e_{i1}$, τ_i in $E_i\mathfrak{A}E_i$ such that $\langle \beta_i, \tau_i \rangle \cong \mathfrak{A}_i$; this gives generators for $\mathfrak{A}'$.
4. Determine ideal generators for $J(\mathfrak{A})$.
5. Determine the relations.

The Carlson-Matthews Algorithm: Outline

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.
2. For each i, compute, with the MeatAxe, a sequence e_{ij} of pairwise orthogonal primitive idempotents with $E_i = \sum_j e_{ij}$.
3. Construct elements $\beta_i \in e_{i1}\,\mathfrak{A}\,e_{i1}$, τ_i in $E_i \mathfrak{A} E_i$ such that $\langle \beta_i, \tau_i \rangle \cong \mathfrak{A}_i$; this gives generators for $\mathfrak{A}'$.
4. Determine ideal generators for $J(\mathfrak{A})$.
5. Determine the relations.

Put $e = \sum_i e_{i1}$. Then $e \mathfrak{A} e$ is the basic algebra of $\mathfrak{A}$.

THE CARLSON-MATTHEWS ALGORITHM: OUTLINE

Here is a very rough outline of the algorithm:

1. Compute, with the MeatAxe, a sequence E_i of pairwise orthogonal idempotents of $\mathfrak{A}$ such that
   - $\sum_i E_i = 1_{\mathfrak{A}}$,
   - $\varphi_j(E_i) = \delta_{ij} 1_{\mathfrak{A}_i}$.
2. For each i, compute, with the MeatAxe, a sequence e_{ij} of pairwise orthogonal primitive idempotents with $E_i = \sum_j e_{ij}$.
3. Construct elements $\beta_i \in e_{i1}\,\mathfrak{A}\,e_{i1}$, τ_i in $E_i \mathfrak{A} E_i$ such that $\langle \beta_i, \tau_i \rangle \cong \mathfrak{A}_i$; this gives generators for $\mathfrak{A}'$.
4. Determine ideal generators for $J(\mathfrak{A})$.
5. Determine the relations.

Put $e = \sum_i e_{i1}$. Then $e\mathfrak{A}e$ is the basic algebra of $\mathfrak{A}$.

Determine matrix representation of $e\mathfrak{A}e$ on $F^{1\times d}e$.

THE CARLSON-MATTHEWS ALGORITHM: STEP 1

1. Choose $E \in \mathfrak{A}$ at random.
2. For i from 2 to r do:
   - Compute minimal polynomial μ_i of $\varphi_i(E)$;
   - Replace E by $E\mu_i(E)$. (This is still in $\mathfrak{A}$.)
3. Now $\varphi_i(E) = 0$ for all $2 \leq i \leq r$.

The Carlson-Matthews Algorithm: Step 1

1. Choose $E \in \mathfrak{A}$ at random.
2. For i from 2 to r do:
   - Compute minimal polynomial μ_i of $\varphi_i(E)$;
   - Replace E by $E\mu_i(E)$. (This is still in $\mathfrak{A}$.)
3. Now $\varphi_i(E) = 0$ for all $2 \leq i \leq r$.
4. If $\varphi_1(E)$ is not invertible, go back to Step 1.

The Carlson-Matthews Algorithm: Step 1

1. Choose $E \in \mathfrak{A}$ at random.
2. For i from 2 to r do:
   - Compute minimal polynomial μ_i of $\varphi_i(E)$;
   - Replace E by $E\mu_i(E)$. (This is still in $\mathfrak{A}$.)
3. Now $\varphi_i(E) = 0$ for all $2 \leq i \leq r$.
4. If $\varphi_1(E)$ is not invertible, go back to Step 1.
5. Compute the minimal polynomial μ of $\varphi_1(E)$.
   Note $\mu = \nu + a$ with $0 \neq a \in F$ and ν has no constant term.

THE CARLSON-MATTHEWS ALGORITHM: STEP 1

1. Choose $E \in \mathfrak{A}$ at random.
2. For i from 2 to r do:
   - Compute minimal polynomial μ_i of $\varphi_i(E)$;
   - Replace E by $E\mu_i(E)$. (This is still in $\mathfrak{A}$.)
3. Now $\varphi_i(E) = 0$ for all $2 \leq i \leq r$.
4. If $\varphi_1(E)$ is not invertible, go back to Step 1.
5. Compute the minimal polynomial μ of $\varphi_1(E)$.
   Note $\mu = \nu + a$ with $0 \neq a \in F$ and ν has no constant term.
6. Replace E by $-\nu(E)/a$; now $\varphi_1(E) = 1_{\mathfrak{A}_1}$.
7. Now $\varphi_j(E^2 - E) = 0$ for all j, i.e. $E^2 - E \in J(\mathfrak{A})$.

The Carlson-Matthews Algorithm: Step 1

1. Choose $E \in \mathfrak{A}$ at random.
2. For i from 2 to r do:
   - Compute minimal polynomial μ_i of $\varphi_i(E)$;
   - Replace E by $E\mu_i(E)$. (This is still in $\mathfrak{A}$.)
3. Now $\varphi_i(E) = 0$ for all $2 \leq i \leq r$.
4. If $\varphi_1(E)$ is not invertible, go back to Step 1.
5. Compute the minimal polynomial μ of $\varphi_1(E)$.
   Note $\mu = \nu + a$ with $0 \neq a \in F$ and ν has no constant term.
6. Replace E by $-\nu(E)/a$; now $\varphi_1(E) = 1_{\mathfrak{A}_1}$.
7. Now $\varphi_j(E^2 - E) = 0$ for all j, i.e. $E^2 - E \in J(\mathfrak{A})$.
8. If $E^2 - E \neq 0$, replace E by E^p; then
   $(E^p)^2 - E^p = (E^2 - E)^p$.
9. Repeat until $E^2 = E$; put $E_1 := E$. ($J(\mathfrak{A})$ is nilpotent.)

THE CARLSON-MATTHEWS ALGORITHM: STEP 1

1. Choose $E \in \mathfrak{A}$ at random.
2. For i from 2 to r do:
   - Compute minimal polynomial μ_i of $\varphi_i(E)$;
   - Replace E by $E\mu_i(E)$. (This is still in $\mathfrak{A}$.)
3. Now $\varphi_i(E) = 0$ for all $2 \leq i \leq r$.
4. If $\varphi_1(E)$ is not invertible, go back to Step 1.
5. Compute the minimal polynomial μ of $\varphi_1(E)$.
   Note $\mu = \nu + a$ with $0 \neq a \in F$ and ν has no constant term.
6. Replace E by $-\nu(E)/a$; now $\varphi_1(E) = 1_{\mathfrak{A}_1}$.
7. Now $\varphi_j(E^2 - E) = 0$ for all j, i.e. $E^2 - E \in J(\mathfrak{A})$.
8. If $E^2 - E \neq 0$, replace E by E^p; then
   $(E^p)^2 - E^p = (E^2 - E)^p$.
9. Repeat until $E^2 = E$; put $E_1 := E$. ($J(\mathfrak{A})$ is nilpotent.)
10. Continue with $(1_{\mathfrak{A}} - E_1)\mathfrak{A}(1_{\mathfrak{A}} - E_1)$.

PRESENTATIONS FOR MODULES

For a finite set $Y_1, \ldots, Y_m$ put

$$\mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m) := \text{free right } \mathfrak{A}\text{-module } \bigoplus_{i=1}^{m} Y_i \mathfrak{A}.$$

PRESENTATIONS FOR MODULES

For a finite set $Y_1, \ldots, Y_m$ put

$$\mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m) := \text{free right } \mathfrak{A}\text{-module } \bigoplus_{i=1}^{m} Y_i\mathfrak{A}.$$

For $R \subset \mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m)$ write

$$\langle Y_1, \ldots, Y_m \mid R \rangle := \mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m)/W,$$

where W is the submodule generated by R.

Presentations for Modules

For a finite set $Y_1, \ldots, Y_m$ put

$$\mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m) := \text{free right } \mathfrak{A}\text{-module } \bigoplus_{i=1}^{m} Y_i\mathfrak{A}.$$

For $R \subset \mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m)$ write

$$\langle Y_1, \ldots, Y_m \mid R \rangle := \mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_m)/W,$$

where W is the submodule generated by R.

An $\mathfrak{A}$-module V is finitely presented if $V \cong \langle Y_1, \ldots, Y_m \mid R \rangle$ for some finite R.

The VectorEnumerator

Let $\mathfrak{A} = \langle X_1, \ldots, X_n \mid R \rangle$ be finitely presented, and let $V = \langle Y_1, \ldots, Y_m \mid R' \rangle$ be a finite presentation for the $\mathfrak{A}$-module V.

The VectorEnumerator

Let $\mathfrak{A} = \langle X_1, \ldots, X_n \mid R \rangle$ be finitely presented, and let $V = \langle Y_1, \ldots, Y_m \mid R' \rangle$ be a finite presentation for the $\mathfrak{A}$-module V.

Theorem (Labonté, Linton)

There is an algorithm, the VectorEnumerator, which terminates, if and only if V is finite-dimensional.
In this case, the VectorEnumerator returns an F-basis $\mathcal{B}$ of V, and representing matrices for X_i w.r.t. $\mathcal{B}$.

The VectorEnumerator

Let $\mathfrak{A} = \langle X_1, \ldots, X_n \mid R \rangle$ be finitely presented, and let $V = \langle Y_1, \ldots, Y_m \mid R' \rangle$ be a finite presentation for the $\mathfrak{A}$-module V.

Theorem (Labonté, Linton)

There is an algorithm, the VectorEnumerator, which terminates, if and only if V is finite-dimensional.
In this case, the VectorEnumerator returns an F-basis $\mathcal{B}$ of V, and representing matrices for X_i w.r.t. $\mathcal{B}$.

Taking $V = \langle Y \mid \emptyset \rangle$, The VectorEnumerator computes the (right) regular representation of $\mathfrak{A}$.

THE VECTORENUMERATOR

Let $\mathfrak{A} = \langle X_1, \ldots, X_n \mid R \rangle$ be finitely presented, and let $V = \langle Y_1, \ldots, Y_m \mid R' \rangle$ be a finite presentation for the $\mathfrak{A}$-module V.

THEOREM (LABONTÉ, LINTON)

There is an algorithm, the VectorEnumerator, which terminates, if and only if V is finite-dimensional.
In this case, the VectorEnumerator returns an F-basis $\mathcal{B}$ of V, and representing matrices for X_i w.r.t. $\mathcal{B}$.

Taking $V = \langle Y \mid \emptyset \rangle$, The VectorEnumerator computes the (right) regular representation of $\mathfrak{A}$.

The VectorEnumerator is a linear version of the Todd-Coxeter algorithm for finitely presented groups.

COXETER GROUPS

Let $M := (m_{ij})_{1 \leq i,j \leq r}$ be a symmetric matrix with $m_{ij} \in \mathbb{Z}$ satisfying $m_{ii} = 1$ and $m_{ij} > 1$ for $i \neq j$.

COXETER GROUPS

Let $M := (m_{ij})_{1 \leq i,j \leq r}$ be a symmetric matrix with $m_{ij} \in \mathbb{Z}$ satisfying $m_{ii} = 1$ and $m_{ij} > 1$ for $i \neq j$.

The group

$$W := W(m_{ij}) := \left\langle s_1, \ldots, s_r \mid (s_i s_j)^{m_{ij}} = 1 \right\rangle_{\text{group}},$$

is called the Coxeter group of M, the elements $s_1, \ldots, s_r$ are the Coxeter generators of W.

COXETER GROUPS

Let $M := (m_{ij})_{1 \leq i,j \leq r}$ be a symmetric matrix with $m_{ij} \in \mathbb{Z}$ satisfying $m_{ii} = 1$ and $m_{ij} > 1$ for $i \neq j$.

The group

$$W := W(m_{ij}) := \left\langle s_1, \ldots, s_r \mid (s_i s_j)^{m_{ij}} = 1 \right\rangle_{\text{group}},$$

is called the Coxeter group of M, the elements $s_1, \ldots, s_r$ are the Coxeter generators of W.

The relations $(s_i s_j)^{m_{ij}} = 1$ ($i \neq j$) are called braid relations.

Coxeter Groups

Let $M := (m_{ij})_{1 \leq i,j \leq r}$ be a symmetric matrix with $m_{ij} \in \mathbb{Z}$ satisfying $m_{ii} = 1$ and $m_{ij} > 1$ for $i \neq j$.

The group

$$W := W(m_{ij}) := \left\langle s_1, \ldots, s_r \mid (s_i s_j)^{m_{ij}} = 1 \right\rangle_{\text{group}},$$

is called the Coxeter group of M, the elements $s_1, \ldots, s_r$ are the Coxeter generators of W.

The relations $(s_i s_j)^{m_{ij}} = 1$ $(i \neq j)$ are called braid relations.

In view of $s_i^2 = 1$, they can be written as $s_i s_j s_i \cdots = s_j s_i s_j \cdots$

COXETER GROUPS

Let $M := (m_{ij})_{1 \leq i,j \leq r}$ be a symmetric matrix with $m_{ij} \in \mathbb{Z}$ satisfying $m_{ii} = 1$ and $m_{ij} > 1$ for $i \neq j$.

The group

$$W := W(m_{ij}) := \left\langle s_1, \ldots, s_r \mid (s_i s_j)^{m_{ij}} = 1 \right\rangle_{\text{group}},$$

is called the Coxeter group of M, the elements $s_1, \ldots, s_r$ are the Coxeter generators of W.

The relations $(s_i s_j)^{m_{ij}} = 1$ ($i \neq j$) are called braid relations.

In view of $s_i^2 = 1$, they can be written as $s_i s_j s_i \cdots = s_j s_i s_j \cdots$

The finite real reflection groups are Coxeter groups.

COXETER GROUPS

Let $M := (m_{ij})_{1 \leq i,j \leq r}$ be a symmetric matrix with $m_{ij} \in \mathbb{Z}$ satisfying $m_{ii} = 1$ and $m_{ij} > 1$ for $i \neq j$.

The group

$$W := W(m_{ij}) := \left\langle s_1, \ldots, s_r \mid (s_i s_j)^{m_{ij}} = 1 \right\rangle_{\text{group}},$$

is called the Coxeter group of M, the elements $s_1, \ldots, s_r$ are the Coxeter generators of W.

The relations $(s_i s_j)^{m_{ij}} = 1$ ($i \neq j$) are called braid relations.

In view of $s_i^2 = 1$, they can be written as $s_i s_j s_i \cdots = s_j s_i s_j \cdots$

The finite real reflection groups are Coxeter groups.

E.g. $S_n = \langle s_1, \ldots, s_{n-1} \mid s_i^2, (s_i s_{i+1})^3, (s_i s_j)^2 \text{ for } |i - j| > 1 \rangle$.

The Iwahori-Hecke Algebra

Let W be a Coxeter group with Coxeter matrix (m_{ij}).

The Iwahori-Hecke Algebra

Let W be a Coxeter group with Coxeter matrix (m_{ij}).
For $q \in F$, the algebra

$$H_{F,q}(W) := \left\langle T_{s_1}, \ldots, T_{s_r} \mid T_{s_i}^2 = q1 + (q-1)T_{s_i}, \text{ braid rel's} \right\rangle_{F\text{-alg.}}$$

is the Iwahori-Hecke algebra of W over F with parameter q.

The Iwahori-Hecke Algebra

Let W be a Coxeter group with Coxeter matrix (m_{ij}).
For $q \in F$, the algebra

$$H_{F,q}(W) := \left\langle T_{s_1}, \ldots, T_{s_r} \mid T_{s_i}^2 = q1 + (q-1)T_{s_i}, \text{ braid rel's} \right\rangle_{F\text{-alg.}}$$

is the Iwahori-Hecke algebra of W over F with parameter q.
Braid rel's: $T_{s_i}T_{s_j}T_{s_i}\cdots = T_{s_j}T_{s_i}T_{s_j}\cdots$ (m_{ij} factors on each side)

THE IWAHORI-HECKE ALGEBRA

Let W be a Coxeter group with Coxeter matrix (m_{ij}).
For $q \in F$, the algebra

$$H_{F,q}(W) := \left\langle T_{s_1}, \ldots, T_{s_r} \mid T_{s_i}^2 = q1 + (q-1)T_{s_i}, \text{ braid rel's } \right\rangle_{F\text{-alg.}}$$

is the Iwahori-Hecke algebra of W over F with parameter q.
Braid rel's: $T_{s_i}T_{s_j}T_{s_i}\cdots = T_{s_j}T_{s_i}T_{s_j}\cdots$ (m_{ij} factors on each side)

FACT

*$H_{F,q}(W)$ has **finite** dimension $|W|$.*

THE IWAHORI-HECKE ALGEBRA

Let W be a Coxeter group with Coxeter matrix (m_{ij}).
For $q \in F$, the algebra

$$H_{F,q}(W) := \left\langle T_{s_1}, \ldots, T_{s_r} \mid T_{s_i}^2 = q1 + (q-1)T_{s_i}, \text{ braid rel's } \right\rangle_{F\text{-alg.}}$$

is the Iwahori-Hecke algebra of W over F with parameter q.
Braid rel's: $T_{s_i} T_{s_j} T_{s_i} \cdots = T_{s_j} T_{s_i} T_{s_j} \cdots$ (m_{ij} factors on each side)

FACT

*$H_{F,q}(W)$ has **finite** dimension $|W|$.*

These Iwahori-Hecke algebras play a crucial role in the representation theory of finite groups of Lie type.

The Iwahori-Hecke Algebra

Let W be a Coxeter group with Coxeter matrix (m_{ij}).
For $q \in F$, the algebra

$$H_{F,q}(W) := \left\langle T_{s_1}, \ldots, T_{s_r} \mid T_{s_i}^2 = q1 + (q-1)T_{s_i}, \text{ braid rel's} \right\rangle_{F\text{-alg.}}$$

is the Iwahori-Hecke algebra of W over F with parameter q.
Braid rel's: $T_{s_i} T_{s_j} T_{s_i} \cdots = T_{s_j} T_{s_i} T_{s_j} \cdots$ (m_{ij} factors on each side)

Fact

*$H_{F,q}(W)$ has **finite** dimension $|W|$.*

These Iwahori-Hecke algebras play a crucial role in the representation theory of finite groups of Lie type.
If $F = \mathbb{Q}(\mathbf{u})$ for an indeterminate $\mathbf{u}$, then $H_{F,\mathbf{u}}$ is called the generic Iwahori-Hecke algebra associated to W.

COMPLEX REFLECTION GROUPS

A complex reflection group is a finite group W generated by pseudo reflections in $\mathrm{GL}_d(\mathbb{C})$.

Complex Reflection Groups

A complex reflection group is a finite group W generated by pseudo reflections in $\mathrm{GL}_d(\mathbb{C})$.

A pseudo reflection is an element of $\mathrm{GL}_d(\mathbb{C})$ of finite order with fixed space of dimension $d - 1$.

Complex Reflection Groups

A complex reflection group is a finite group W generated by pseudo reflections in $\mathrm{GL}_d(\mathbb{C})$.

A pseudo reflection is an element of $\mathrm{GL}_d(\mathbb{C})$ of finite order with fixed space of dimension $d - 1$.

Shephard and Todd classified the irreducible complex reflection groups. Apart from a (3-parameter) infinite family there are 34 exceptional groups.

Complex Reflection Groups

A complex reflection group is a finite group W generated by pseudo reflections in $\mathrm{GL}_d(\mathbb{C})$.

A pseudo reflection is an element of $\mathrm{GL}_d(\mathbb{C})$ of finite order with fixed space of dimension $d-1$.

Shephard and Todd classified the irreducible complex reflection groups. Apart from a (3-parameter) infinite family there are 34 exceptional groups.

Many of them have a Coxeter like presentation, e.g.

$$G_{25} = \langle r, s, t \mid r^3 = s^3 = t^3 = 1, rsr = srs, sts = tst, rt = tr \rangle.$$

COMPLEX REFLECTION GROUPS

A complex reflection group is a finite group W generated by pseudo reflections in $\mathrm{GL}_d(\mathbb{C})$.

A pseudo reflection is an element of $\mathrm{GL}_d(\mathbb{C})$ of finite order with fixed space of dimension $d-1$.

Shephard and Todd classified the irreducible complex reflection groups. Apart from a (3-parameter) infinite family there are 34 exceptional groups.

Many of them have a Coxeter like presentation, e.g.

$$G_{25} = \langle r, s, t \mid r^3 = s^3 = t^3 = 1, rsr = srs, sts = tst, rt = tr \rangle.$$

One can thus associate a Hecke algebra to them, called Cyclotomic Hecke Algebra (Ariki, Koike; Broué, Malle).

The VectorEnumerator: An Application

W finite complex reflection group, given by a Coxeter like presentation on S (order + braid relations)

The VectorEnumerator: An Application

W finite complex reflection group, given by a Coxeter like presentation on S (order + braid relations)

Let $\mathbf{u} := (u_{s,j} \mid s \in S, 0 \leq j \leq |s| - 1)$ be a vector of indeterminates, $F := \mathbb{Q}(\mathbf{u})$ rational function field.

THE VECTORENUMERATOR: AN APPLICATION

W finite complex reflection group, given by a Coxeter like presentation on S (order + braid relations)

Let $\mathbf{u} := (u_{s,j} \mid s \in S, 0 \leq j \leq |s| - 1)$ be a vector of indeterminates, $F := \mathbb{Q}(\mathbf{u})$ rational function field.

$$H_{F,\mathbf{u}} := \langle T_s, s \in S \mid \text{ braid relations, } \prod_{j=0}^{|s|-1} (T_s - u_{s,j}) \rangle$$

is the cyclotomic Hecke algebra associated to (W, S).

The VectorEnumerator: An Application

W finite complex reflection group, given by a Coxeter like presentation on S (order + braid relations)

Let $\mathbf{u} := (u_{s,j} \mid s \in S, 0 \leq j \leq |s| - 1)$ be a vector of indeterminates, $F := \mathbb{Q}(\mathbf{u})$ rational function field.

$$H_{F,\mathbf{u}} := \langle T_s, s \in S \mid \text{ braid relations}, \prod_{j=0}^{|s|-1} (T_s - u_{s,j}) \rangle$$

is the cyclotomic Hecke algebra associated to (W, S).

Conjecture (Broué, Malle, Rouquier): $\dim H_{F,\mathbf{u}} = |W|$.

THE VECTORENUMERATOR: AN APPLICATION

W finite complex reflection group, given by a Coxeter like presentation on S (order + braid relations)

Let $\mathbf{u} := (u_{s,j} \mid s \in S, 0 \leq j \leq |s| - 1)$ be a vector of indeterminates, $F := \mathbb{Q}(\mathbf{u})$ rational function field.

$$H_{F,\mathbf{u}} := \langle T_s, s \in S \mid \text{ braid relations, } \prod_{j=0}^{|s|-1} (T_s - u_{s,j}) \rangle$$

is the cyclotomic Hecke algebra associated to (W, S).

Conjecture (Broué, Malle, Rouquier): $\dim H_{F,\mathbf{u}} = |W|$.

Jürgen Müller proved this for some exceptional cyclotomic Hecke algebras using the VectorEnumerator over $\mathbb{Q}(\mathbf{u})$.

The VectorEnumerator: An Application

W finite complex reflection group, given by a Coxeter like presentation on S (order + braid relations)

Let $\mathbf{u} := (u_{s,j} \mid s \in S, 0 \leq j \leq |s| - 1)$ be a vector of indeterminates, $F := \mathbb{Q}(\mathbf{u})$ rational function field.

$$H_{F,\mathbf{u}} := \langle T_s, s \in S \mid \text{ braid relations}, \prod_{j=0}^{|s|-1} (T_s - u_{s,j}) \rangle$$

is the cyclotomic Hecke algebra associated to (W, S).

Conjecture (Broué, Malle, Rouquier): $\dim H_{F,\mathbf{u}} = |W|$.

Jürgen Müller proved this for some exceptional cyclotomic Hecke algebras using the VectorEnumerator over $\mathbb{Q}(\mathbf{u})$.

Ivan Marin and collaborators proved many more instances.

HOMOMORPHISMS

Let $V, W \in \text{mod-}\mathfrak{A}$.

Homomorphisms

Let $V, W \in \text{mod-}\mathfrak{A}$.

Recall: An $\mathfrak{A}$-homomorphism from V to W is a linear map $\varphi : V \to W$, such that

$$(v\varphi)\mathfrak{a} = (v\mathfrak{a})\varphi \qquad (1)$$

for all $v \in V$, $\mathfrak{a} \in \mathfrak{A}$.

Homomorphisms

Let $V, W \in$ mod-$\mathfrak{A}$.

Recall: An $\mathfrak{A}$-homomorphism from V to W is a linear map $\varphi : V \to W$, such that

$$(v\varphi)\mathfrak{a} = (v\mathfrak{a})\varphi \qquad (1)$$

for all $v \in V$, $\mathfrak{a} \in \mathfrak{A}$.

$\mathrm{Hom}_{\mathfrak{A}}(V, W)$: set of $\mathfrak{A}$-homomorphism from V to W

Homomorphisms

Let $V, W \in$ mod-$\mathfrak{A}$.

Recall: An $\mathfrak{A}$-homomorphism from V to W is a linear map $\varphi : V \to W$, such that

$$(v\varphi)\mathfrak{a} = (v\mathfrak{a})\varphi \quad (1)$$

for all $v \in V$, $\mathfrak{a} \in \mathfrak{A}$.

$\mathrm{Hom}_{\mathfrak{A}}(V, W)$: set of $\mathfrak{A}$-homomorphism from V to W

Application (Lux and Szőke): Let V and W be indecomposable, and let $\varphi_1, \ldots, \varphi_n$ be a basis of $\mathrm{Hom}_{\mathfrak{A}}(V, W)$.

Homomorphisms

Let $V, W \in \text{mod-}\mathfrak{A}$.

Recall: An $\mathfrak{A}$-homomorphism from V to W is a linear map $\varphi : V \to W$, such that

$$(v\varphi)\mathfrak{a} = (v\mathfrak{a})\varphi \qquad (1)$$

for all $v \in V$, $\mathfrak{a} \in \mathfrak{A}$.

$\text{Hom}_{\mathfrak{A}}(V, W)$: set of $\mathfrak{A}$-homomorphism from V to W

Application (Lux and Szőke): Let V and W be indecomposable, and let $\varphi_1, \ldots, \varphi_n$ be a basis of $\text{Hom}_{\mathfrak{A}}(V, W)$. Then: V and W are isomorphic, if and only if one of the φ_i is an isomorphism.

Computing Homomorphisms, I

$\mathrm{Hom}_{\mathfrak{A}}(V, W)$ can be computed: Equation (1) leads to a system of linear equations.

Computing Homomorphisms, I

$\text{Hom}_{\mathfrak{A}}(V, W)$ can be computed: Equation (1) leads to a system of linear equations.

Let $\mathfrak{A} = F\langle \mathfrak{a}_1, \ldots, \mathfrak{a}_l \rangle$ as F-algebra, $\dim(V) = m$, $\dim(W) = n$, and let the action of $\mathfrak{A}$ on V be given by $A_1, \ldots, A_l \in F^{m \times m}$ and on W by $B_1, \ldots, B_l \in F^{n \times n}$.

Computing Homomorphisms, I

$\mathrm{Hom}_{\mathfrak{A}}(V, W)$ can be computed: Equation (1) leads to a system of linear equations.

Let $\mathfrak{A} = F\langle \mathfrak{a}_1, \ldots, \mathfrak{a}_l \rangle$ as F-algebra, $\dim(V) = m$, $\dim(W) = n$, and let the action of $\mathfrak{A}$ on V be given by $A_1, \ldots, A_l \in F^{m \times m}$ and on W by $B_1, \ldots, B_l \in F^{n \times n}$.

Then

$$\mathrm{Hom}_{\mathfrak{A}}(V, W) \cong \{U \in F^{m \times n} \mid A_i U = U B_i \text{ for all } 1 \leq i \leq l\}. \quad (2)$$

Computing Homomorphisms, I

$\text{Hom}_{\mathfrak{A}}(V, W)$ can be computed: Equation (1) leads to a system of linear equations.

Let $\mathfrak{A} = F\langle \mathfrak{a}_1, \ldots, \mathfrak{a}_l \rangle$ as F-algebra, $\dim(V) = m$, $\dim(W) = n$, and let the action of $\mathfrak{A}$ on V be given by $A_1, \ldots, A_l \in F^{m\times m}$ and on W by $B_1, \ldots, B_l \in F^{n\times n}$.

Then

$$\text{Hom}_{\mathfrak{A}}(V, W) \cong \{U \in F^{m\times n} \mid A_i U = U B_i \text{ for all } 1 \leq i \leq l\}. \quad (2)$$

Taking the entries of U as unknowns, (2) is a system of lmn equations in mn unknowns.

Computing Homomorphisms, I

$\text{Hom}_{\mathfrak{A}}(V, W)$ can be computed: Equation (1) leads to a system of linear equations.

Let $\mathfrak{A} = F\langle \mathfrak{a}_1, \ldots, \mathfrak{a}_l \rangle$ as F-algebra, $\dim(V) = m$, $\dim(W) = n$, and let the action of $\mathfrak{A}$ on V be given by $A_1, \ldots, A_l \in F^{m \times m}$ and on W by $B_1, \ldots, B_l \in F^{n \times n}$.

Then

$$\text{Hom}_{\mathfrak{A}}(V, W) \cong \{U \in F^{m \times n} \mid A_i U = U B_i \text{ for all } 1 \leq i \leq l\}. \quad (2)$$

Taking the entries of U as unknowns, (2) is a system of lmn equations in mn unknowns.

This was the first approach taken be G. Schneider in 1990. It is restricted to small values of l, m, n.

COMPUTING HOMOMORPHISMS, II

C. Leedham-Green and J. Cannon develop an algorithm that performs better, implemented in MAGMA by M. Smith.

COMPUTING HOMOMORPHISMS, II

C. Leedham-Green and J. Cannon develop an algorithm that performs better, implemented in MAGMA by M. Smith.

Lux and Szőke reduce the number of unknowns by using a (short) presentation of V.

COMPUTING HOMOMORPHISMS, II

C. Leedham-Green and J. Cannon develop an algorithm that performs better, implemented in MAGMA by M. Smith.

Lux and Szőke reduce the number of unknowns by using a (short) presentation of V.

Suppose $V = \langle Y_1, \ldots, Y_r \mid R \rangle$ with R finite, i.e. V is given by a finite presentation.

Computing Homomorphisms, II

C. Leedham-Green and J. Cannon develop an algorithm that performs better, implemented in Magma by M. Smith.

Lux and Szőke reduce the number of unknowns by using a (short) presentation of V.

Suppose $V = \langle Y_1, \ldots, Y_r \mid R \rangle$ with R finite, i.e. V is given by a finite presentation.

Then

$$\mathrm{Hom}_{\mathfrak{A}}(V, W) \cong \{\psi \in \mathrm{Hom}_{\mathfrak{A}}(\mathrm{FM}_{\mathfrak{A}}(Y_1, \ldots, Y_r), W) \mid R \subseteq \mathrm{Ker}(\psi)\}.$$

Computing Homomorphisms, III

Simplest case: $V = \langle Y \mid R \rangle$ is cyclic.

Computing Homomorphisms, III

Simplest case: $V = \langle Y \mid R \rangle$ is cyclic.
Let $w_1, \ldots, w_n$ be a basis of W.

Computing Homomorphisms, III

Simplest case: $V = \langle Y \mid R \rangle$ is cyclic.
Let $w_1, \ldots, w_n$ be a basis of W.
Let $\psi \in \mathrm{Hom}_{\mathfrak{A}}(Y\mathfrak{A}, W)$ be defined by $Y\psi = \sum_{j=1}^{n} u_j w_j$ with unknown coefficients u_j.

COMPUTING HOMOMORPHISMS, III

Simplest case: $V = \langle Y \mid R \rangle$ is cyclic.
Let $w_1, \ldots, w_n$ be a basis of W.
Let $\psi \in \mathrm{Hom}_{\mathfrak{A}}(Y\mathfrak{A}, W)$ be defined by $Y\psi = \sum_{j=1}^{n} u_j w_j$ with unknown coefficients u_j.
Let $s = Y\mathfrak{a} \in R$ for some $\mathfrak{a} \in \mathfrak{A}$. Suppose that

$$w_j\mathfrak{a} = \sum_{k=1}^{n} a_{jk} w_k,$$

i.e. $A = (a_{jk}) \in F^{n \times n}$ is the matrix of the action of $\mathfrak{a}$ on W.

COMPUTING HOMOMORPHISMS, III

Simplest case: $V = \langle Y \mid R \rangle$ is cyclic.
Let $w_1, \ldots, w_n$ be a basis of W.
Let $\psi \in \mathrm{Hom}_{\mathfrak{A}}(Y\mathfrak{A}, W)$ be defined by $Y\psi = \sum_{j=1}^{n} u_j w_j$ with unknown coefficients u_j.
Let $s = Y\mathfrak{a} \in R$ for some $\mathfrak{a} \in \mathfrak{A}$. Suppose that

$$w_j\mathfrak{a} = \sum_{k=1}^{n} a_{jk} w_k,$$

i.e. $A = (a_{jk}) \in F^{n\times n}$ is the matrix of the action of $\mathfrak{a}$ on W.
Then $s\psi = 0$, yields the n equations

$$\sum_{j=1}^{n} u_j a_{jk} = 0 \qquad \text{for all } k = 1, \ldots, n.$$

Direct Decompositions

Let $V \in \text{mod-}\mathfrak{A}$. Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V) := \text{Hom}_{\mathfrak{A}}(V, V)$ (this is an F-algebra, the endomorphism ring of V).

Direct Decompositions

Let $V \in \text{mod-}\mathfrak{A}$. Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V) := \text{Hom}_{\mathfrak{A}}(V, V)$ (this is an F-algebra, the endomorphism ring of V).

Suppose

$$V = V_1 \oplus V_2 \oplus \cdots \oplus V_l,$$

with non-zero $\mathfrak{A}$-submodules V_i.

Direct Decompositions

Let $V \in \text{mod-}\mathfrak{A}$. Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V) := \text{Hom}_{\mathfrak{A}}(V, V)$ (this is an F-algebra, the endomorphism ring of V).

Suppose

$$V = V_1 \oplus V_2 \oplus \cdots \oplus V_l,$$

with non-zero $\mathfrak{A}$-submodules V_i.

Let $\pi_i \in \mathfrak{E}$ denote the projection to V_i.
Then $\pi_i^2 = \pi_i$, i.e., π_i is an idempotent in $\mathfrak{E}$.

Direct Decompositions

Let $V \in \text{mod-}\mathfrak{A}$. Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V) := \text{Hom}_{\mathfrak{A}}(V, V)$ (this is an F-algebra, the endomorphism ring of V).

Suppose

$$V = V_1 \oplus V_2 \oplus \cdots \oplus V_l,$$

with non-zero $\mathfrak{A}$-submodules V_i.

Let $\pi_i \in \mathfrak{E}$ denote the projection to V_i.
Then $\pi_i^2 = \pi_i$, i.e., π_i is an idempotent in $\mathfrak{E}$.

The left ideal $\mathfrak{E}\pi_i$ may be identified with $\text{Hom}_{\mathfrak{A}}(V, V_i)$.

Direct Decompositions

Let $V \in \text{mod-}\mathfrak{A}$. Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V) := \text{Hom}_{\mathfrak{A}}(V, V)$ (this is an F-algebra, the endomorphism ring of V).

Suppose

$$V = V_1 \oplus V_2 \oplus \cdots \oplus V_l,$$

with non-zero $\mathfrak{A}$-submodules V_i.

Let $\pi_i \in \mathfrak{E}$ denote the projection to V_i.
Then $\pi_i^2 = \pi_i$, i.e., π_i is an idempotent in $\mathfrak{E}$.

The left ideal $\mathfrak{E}\pi_i$ may be identified with $\text{Hom}_{\mathfrak{A}}(V, V_i)$.

Proposition (Fitting correspondence)

1. $\mathfrak{E} = \mathfrak{E}\pi_1 \oplus \mathfrak{E}\pi_2 \oplus \cdots \oplus \mathfrak{E}\pi_l$.
2. *$V_i \cong V_j$ as $\mathfrak{A}$-modules, if and only if $\mathfrak{E}\pi_i \cong \mathfrak{E}\pi_j$ as left ideals.*
3. *V_i is indecomposable if and only if π_i is primitive.*

Lux and Szőke's Algorithm: Background

K. Lux and M. Szőke: algorithm to find the indecomposable components V_i of $V \in \text{mod-}\mathfrak{A}$.

Lux and Szőke's Algorithm: Background

K. Lux and M. Szőke: algorithm to find the indecomposable components V_i of $V \in$ mod-$\mathfrak{A}$.

Put $\mathfrak{E} := \mathrm{End}_{\mathfrak{A}}(V)$, write $\bar{\ } : \mathfrak{E} \to \mathfrak{E}/J(\mathfrak{E}) =: \bar{\mathfrak{E}}$ (natural map).

LUX AND SZŐKE'S ALGORITHM: BACKGROUND

K. Lux and M. Szőke: algorithm to find the indecomposable components V_i of $V \in$ mod-$\mathfrak{A}$.

Put $\mathfrak{E} := \mathrm{End}_{\mathfrak{A}}(V)$, write $\bar{\ } : \mathfrak{E} \to \mathfrak{E}/J(\mathfrak{E}) =: \bar{\mathfrak{E}}$ (natural map).

Suppose $\bar{\mathfrak{E}} = S_1 \oplus \cdots \oplus S_n$ is the decomposition of $\bar{\mathfrak{E}}$ into simple left ideals.

LUX AND SZŐKE'S ALGORITHM: BACKGROUND

K. Lux and M. Szőke: algorithm to find the indecomposable components V_i of $V \in$ mod-$\mathfrak{A}$.

Put $\mathfrak{E} := \mathrm{End}_{\mathfrak{A}}(V)$, write $\bar{\ } : \mathfrak{E} \to \mathfrak{E}/J(\mathfrak{E}) =: \bar{\mathfrak{E}}$ (natural map).

Suppose $\bar{\mathfrak{E}} = S_1 \oplus \cdots \oplus S_n$ is the decomposition of $\bar{\mathfrak{E}}$ into simple left ideals.

Let $\varepsilon_i' \in \mathfrak{E}$ be non-nilpotent with $\bar{\mathfrak{E}}\bar{\varepsilon}_i' = S_i$, $1 \leq i \leq n$.

LUX AND SZŐKE'S ALGORITHM: BACKGROUND

K. Lux and M. Szőke: algorithm to find the indecomposable components V_i of $V \in \text{mod-}\mathfrak{A}$.

Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V)$, write $\bar{}: \mathfrak{E} \to \mathfrak{E}/J(\mathfrak{E}) =: \bar{\mathfrak{E}}$ (natural map).

Suppose $\bar{\mathfrak{E}} = S_1 \oplus \cdots \oplus S_n$ is the decomposition of $\bar{\mathfrak{E}}$ into simple left ideals.

Let $\varepsilon_i' \in \mathfrak{E}$ be non-nilpotent with $\bar{\mathfrak{E}}\bar{\varepsilon}_i' = S_i$, $1 \leq i \leq n$.

Then for suitable powers ε_i of ε_i' the following are satisfied:

1. $\bar{\mathfrak{E}}\bar{\varepsilon}_i = S_i$,
2. $\mathfrak{E}\varepsilon_i$ is a left PIM of $\mathfrak{E}$,
3. $\mathfrak{E} = \mathfrak{E}\varepsilon_1 \oplus \cdots \oplus \mathfrak{E}\varepsilon_n$.

Lux and Szőke's Algorithm: Background

K. Lux and M. Szőke: algorithm to find the indecomposable components V_i of $V \in \text{mod-}\mathfrak{A}$.

Put $\mathfrak{E} := \text{End}_{\mathfrak{A}}(V)$, write $\bar{\ } : \mathfrak{E} \to \mathfrak{E}/J(\mathfrak{E}) =: \bar{\mathfrak{E}}$ (natural map).

Suppose $\bar{\mathfrak{E}} = S_1 \oplus \cdots \oplus S_n$ is the decomposition of $\bar{\mathfrak{E}}$ into simple left ideals.

Let $\varepsilon_i' \in \mathfrak{E}$ be non-nilpotent with $\bar{\mathfrak{E}}\bar{\varepsilon}_i' = S_i$, $1 \leq i \leq n$.

Then for suitable powers ε_i of ε_i' the following are satisfied:

1. $\bar{\mathfrak{E}}\bar{\varepsilon}_i = S_i$,
2. $\mathfrak{E}\varepsilon_i$ is a left PIM of $\mathfrak{E}$,
3. $\mathfrak{E} = \mathfrak{E}\varepsilon_1 \oplus \cdots \oplus \mathfrak{E}\varepsilon_n$.

Thus $V = V_i \oplus \cdots \oplus V_n$ with the indecomposables $V_i = V\varepsilon_i$.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.
4. Compute $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathfrak{E}$ such that $\bar{\mathcal{C}}_i$ is a basis for S_i.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.
4. Compute $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathfrak{E}$ such that $\bar{\mathcal{C}}_i$ is a basis for S_i.
5. Choose $\varepsilon_i' \in \mathcal{C}_i$ non-nilpotent.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.
4. Compute $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathfrak{E}$ such that $\bar{\mathcal{C}}_i$ is a basis for S_i.
5. Choose $\varepsilon_i' \in \mathcal{C}_i$ non-nilpotent.
6. Find ε_i by powering up ε_i'.

LUX AND SZŐKE'S ALGORITHM: OUTLINE

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.
4. Compute $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathfrak{E}$ such that $\bar{\mathcal{C}}_i$ is a basis for S_i.
5. Choose $\varepsilon_i' \in \mathcal{C}_i$ non-nilpotent.
6. Find ε_i by powering up ε_i'.

Remarks: 1–4 can be achieved with the MeatAxe.

LUX AND SZŐKE'S ALGORITHM: OUTLINE

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.
4. Compute $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathfrak{E}$ such that $\bar{\mathcal{C}}_i$ is a basis for S_i.
5. Choose $\varepsilon'_i \in \mathcal{C}_i$ non-nilpotent.
6. Find ε_i by powering up ε'_i.

Remarks: 1–4 can be achieved with the MeatAxe.

$\mathcal{C}_i$ necessarily contains a non-nilpotent element.

Lux and Szőke's Algorithm: Outline

Here is an outline of the Lux-Szőke's algorithm:

1. Compute $\mathfrak{E}$ in its left regular representation.
2. Determine the composition factors of $\mathfrak{E}$.
3. Compute a basis for $J(\mathfrak{E})$.
4. Compute $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq \mathfrak{E}$ such that $\bar{\mathcal{C}}_i$ is a basis for S_i.
5. Choose $\varepsilon_i' \in \mathcal{C}_i$ non-nilpotent.
6. Find ε_i by powering up ε_i'.

Remarks: 1–4 can be achieved with the MeatAxe.

$\mathcal{C}_i$ necessarily contains a non-nilpotent element.

$\varepsilon_i = {\varepsilon_i'}^m$ if $\mathrm{Ker}({\varepsilon_i'}^m) = \mathrm{Ker}({\varepsilon_i'}^{2m})$.

Related Topics

More advanced topics, which I did not present in this series of lectures include:

1. Wedderburn decomposition of group algebras

Related Topics

More advanced topics, which I did not present in this series of lectures include:

1. Wedderburn decomposition of group algebras
2. Integral representations and lattices (representations of groups over the integers or rings of algebraic integers, lattices, . . .)

Related Topics

More advanced topics, which I did not present in this series of lectures include:

1. Wedderburn decomposition of group algebras
2. Integral representations and lattices (representations of groups over the integers or rings of algebraic integers, lattices, . . .)
3. Cohomology (low degree cohomology of groups, cohomology rings, module varieties, . . .)

Related Topics

More advanced topics, which I did not present in this series of lectures include:

1. Wedderburn decomposition of group algebras
2. Integral representations and lattices (representations of groups over the integers or rings of algebraic integers, lattices, ...)
3. Cohomology (low degree cohomology of groups, cohomology rings, module varieties, ...)
4. Representations of algebras given by quivers with relations

Related Topics

More advanced topics, which I did not present in this series of lectures include:

1. Wedderburn decomposition of group algebras
2. Integral representations and lattices (representations of groups over the integers or rings of algebraic integers, lattices, ...)
3. Cohomology (low degree cohomology of groups, cohomology rings, module varieties, ...)
4. Representations of algebras given by quivers with relations
5. Representations of Lie algebras

Related Topics

More advanced topics, which I did not present in this series of lectures include:

1. Wedderburn decomposition of group algebras
2. Integral representations and lattices (representations of groups over the integers or rings of algebraic integers, lattices, . . .)
3. Cohomology (low degree cohomology of groups, cohomology rings, module varieties, . . .)
4. Representations of algebras given by quivers with relations
5. Representations of Lie algebras
6. Invariant theory
7. . . .

References

1. J. F. Carlson and G. Matthews, Generators and relations for matrix algebras, *J. Algebra* **300** (2006), 134–159.
2. K. Lux and M. Szőke, Computing Homomorphism Spaces between Modules over Finite Dimensional Algebras, *Experim. Math.* **12** (2003), 91–98.
3. K. Lux and M. Szőke, Computing Decompositions of Modules over Finite-Dimensional Algebras, *Experim. Math.* **16** (2007), 1–6.

Thank you for your attention!