Actions, representations and various algebraic structures

Max Neunhöffer

University of St Andrews

12 November 2008
Actions and representations

An action of $G$ on $X$ is a map

$$A : X \times G \rightarrow X, \quad (x, g) \mapsto x \cdot g$$
Actions and representations

An action of $G$ on $X$ is a map

$$A : X \times G \rightarrow X, \quad (x, g) \mapsto x \cdot g$$

A representation of $G$ on $X$ is a map

$$R : G \rightarrow X^X = \{ f : X \rightarrow X \}$$
Actions and representations

An action of $G$ on $X$ is a map

$$A : X \times G \rightarrow X, \quad (x, g) \mapsto x \cdot g$$

A representation of $G$ on $X$ is a map

$$R : G \rightarrow X^X = \{ f : X \rightarrow X \}$$

The two concepts are the same:
Actions and representations

An action of $G$ on $X$ is a map

$$A : X \times G \to X, \quad (x, g) \mapsto x \cdot g$$

A representation of $G$ on $X$ is a map

$$R : G \to X^X = \{f : X \to X\}$$

The two concepts are the same:

given $A$, set

$$R(g) := (x \mapsto A(x, g)) = (x \mapsto x \cdot g)$$
Actions and representations

An action of \( G \) on \( X \) is a map

\[
A : X \times G \to X, \quad (x, g) \mapsto x \cdot g
\]

A representation of \( G \) on \( X \) is a map

\[
R : G \to X^X = \{f : X \to X\}
\]

The two concepts are the same:

given \( A \), set

\[
R(g) := (x \mapsto A(x, g)) = (x \mapsto x \cdot g)
\]

given \( R \), set

\[
A(x, g) := R(g)(x)
\]
Let $\mathbb{F}$ be a field and $G$ a finite group.
Group algebras — definition

Let $F$ be a field and $G$ a finite group.

$FG := \text{vector space with basis } G$, multiplication inherited from $G$ and distributive law:

$$
\left( \sum_{g \in G} \lambda_g \cdot g \right) \cdot \left( \sum_{\tilde{g} \in G} \mu_{\tilde{g}} \cdot \tilde{g} \right) = \sum_{g, \tilde{g} \in G} \lambda_g \cdot \mu_{\tilde{g}} \cdot (g\tilde{g})
$$

for $\lambda_g, \mu_{\tilde{g}} \in F$. 
Group algebras — definition

Let \( F \) be a field and \( G \) a finite group.

\[ FG := \text{vector space with basis } G, \text{ multiplication inherited from } G \text{ and distributive law:} \]

\[
\left( \sum_{g \in G} \lambda_g \cdot g \right) \cdot \left( \sum_{\tilde{g} \in G} \mu_{\tilde{g}} \cdot \tilde{g} \right) = \sum_{g, \tilde{g} \in G} \lambda_g \cdot \mu_{\tilde{g}} \cdot (g\tilde{g})
\]

for \( \lambda_g, \mu_{\tilde{g}} \in F \).

\[ FG := \{ f : G \to F \} \text{ with pointwise addition and convolution product:} \]

\[
(f \cdot h)(g) := \sum_{\tilde{g} \in G} f(g \cdot \tilde{g}^{-1}) \cdot h(\tilde{g})
\]

for \( f, h : G \to F \).
Group algebras — definition

Let \( \mathbb{F} \) be a field and \( G \) a finite group.

\[
\mathbb{F}G := \text{vector space with basis } G, \text{ multiplication inherited from } G \text{ and distributive law:}
\]

\[
\left( \sum_{g \in G} \lambda_g \cdot g \right) \cdot \left( \sum_{\tilde{g} \in G} \mu_{\tilde{g}} \cdot \tilde{g} \right) = \sum_{g, \tilde{g} \in G} \lambda_g \cdot \mu_{\tilde{g}} \cdot (g\tilde{g})
\]

for \( \lambda_g, \mu_{\tilde{g}} \in \mathbb{F} \).

\[
\mathbb{F}G := \{ f : G \to \mathbb{F} \} \text{ with pointwise addition and convolution product:}
\]

\[
(f \cdot h)(g) := \sum_{\tilde{g} \in G} f(g \cdot \tilde{g}^{-1}) \cdot h(\tilde{g})
\]

for \( f, h : G \to \mathbb{F} \).

\[
\mathbb{F}G := \text{associative } \mathbb{F}-\text{algebra with generators } G \text{ and relations } g \cdot \tilde{g} - (g\tilde{g}) = 0 \text{ for } g, \tilde{g} \in G.
\]
Group algebras — properties

\(\mathbb{F}:\) field, \(G:\) group, \(\mathbb{F}G:\) group algebra, \(V: \mathbb{F}\)-vector space.
Group algebras — properties

\( \mathbb{F} \): field, \( G \): group, \( \mathbb{F}G \): group algebra, \( V \): \( \mathbb{F} \)-vector space.

There is a bijection between

\[ \{ \varphi : G \to \text{GL}(V) \mid \varphi \text{ is a group homomorphism} \} \]

and

\[ \{ \psi : \mathbb{F}G \to \text{End}_\mathbb{F}(V) \mid \psi \text{ is an algebra homomorphism} \} \]
Group algebras — properties

\( F: \text{field}, \ G: \text{group}, \ FG: \text{group algebra}, \ V: F\text{-vector space}. \)

There is a bijection between

\[ \{ \varphi : G \to \text{GL}(V) \mid \varphi \text{ is a group homomorphism} \} \]

and

\[ \{ \psi : FG \to \text{End}_F(V) \mid \psi \text{ is an algebra homomorphism} \} \]

Given \( \varphi : G \to \text{GL}(V) \), define

\[ \psi \left( \sum_{g \in G} \lambda_g \cdot g \right) := \sum_{g \in G} \lambda_g \cdot \varphi(g) \]

(use finite presentation).
Group algebras — properties

\( F: \text{field}, \ G: \text{group}, \ F[G]: \text{group algebra}, \ V: F\text{-vector space.} \)

There is a bijection between

\[
\{ \varphi : G \rightarrow \text{GL}(V) \mid \varphi \text{ is a group homomorphism} \}
\]

and

\[
\{ \psi : F[G] \rightarrow \text{End}_F(V) \mid \psi \text{ is an algebra homomorphism} \}
\]

Given \( \varphi : G \rightarrow \text{GL}(V) \), define

\[
\psi \left( \sum_{g \in G} \lambda_g \cdot g \right) := \sum_{g \in G} \lambda_g \cdot \varphi(g)
\]

(use finite presentation).

Given \( \psi : F[G] \rightarrow \text{End}_F(V) \), simply restrict \( \varphi := \psi \mid_G \), since

\[
1_V = \psi(1_G) = \psi(g \cdot g^{-1}) = \psi(g) \cdot \psi(g^{-1}) \quad \text{for all } g \in G.
\]
Modules

**Definition (G-module or \( \mathbb{F}G \)-module)**

An \( \mathbb{F} \)-vector space \( V \) together with

- a group homomorphism \( \varphi : G \to \text{GL}(V) \),
- or an algebra homomorphism \( \psi : \mathbb{F}G \to \text{End}_\mathbb{F}(V) \)

is called a **\( G \)-module over \( \mathbb{F} \)** or an **\( \mathbb{F}G \)**-module.
Modules

Definition (G-module or $FG$-module)

An $F$-vector space $V$ together with

- a group homomorphism $\varphi : G \rightarrow GL(V)$,
- or an algebra homomorphism $\psi : FG \rightarrow \text{End}_F(V)$

is called a G-module over $F$ or an $FG$-module.

This is nothing but

an $F$-vector space with an $F$-linear action for $G$. 
Modules

Definition (G-module or FG-module)

An $\mathbb{F}$-vector space $V$ together with
- a group homomorphism $\varphi : G \rightarrow \text{GL}(V)$,
- or an algebra homomorphism $\psi : \mathbb{F}G \rightarrow \text{End}_\mathbb{F}(V)$
is called a $G$-module over $\mathbb{F}$ or an $\mathbb{F}G$-module.

This is nothing but

an $\mathbb{F}$-vector space with an $\mathbb{F}$-linear action for $G$.

This is nothing but

an $\mathbb{F}$-linear representation for $G$. 
Kernels and faithfulness

Let $A : X \times G \to X$ be an action,
Kernels and faithfulness

Let $A : X \times G \to X$ be an action, or equivalently, let $R : G \to X^X$ be a representation.
Kernels and faithfulness

Let $A : X \times G \to X$ be an action, or equivalently, let $R : G \to X^X$ be a representation.

Depending on the types of $G$ and $X$, it might make sense to speak of the kernel of the representation $R$ or not.
Kernels and faithfulness

Let $A : X \times G \to X$ be an action, or equivalently, let $R : G \to X^X$ be a representation.

Depending on the types of $G$ and $X$, it might make sense to speak of the kernel of the representation $R$ or not.

**Definition (Faithful representation/action)**

We call the representation $R$ (or the action $A$) **faithful**, if its kernel $\ker R$ is trivial.
Kernels and faithfulness

Let $A : X \times G \to X$ be an action, or equivalently, let $R : G \to X^X$ be a representation.

Depending on the types of $G$ and $X$, it might make sense to speak of the kernel of the representation $R$ or not.

**Definition (Faithful representation/action)**

We call the representation $R$ (or the action $A$) **faithful**, if its kernel $\ker R$ is trivial.

**Note:** If a $G$-module $V$ over $\mathbb{F}$ is faithful, it does not necessarily follow that the corresponding $\mathbb{F}G$-module $V$ is faithful!
Homomorphisms and isomorphisms

Let $A : X \times G \to X$ and $\tilde{A} : \tilde{X} \times G \to \tilde{X}$ be two actions.
Homomorphisms and isomorphisms

Let $A : X \times G \to X$ and $\tilde{A} : \tilde{X} \times G \to \tilde{X}$ be two actions.

Definition (G-homomorphism)

A homomorphism $\varphi : X \to \tilde{X}$ is called a $G$-homomorphism or $G$-equivariant, if

$$\varphi(x \cdot g) = \varphi(x) \cdot g$$

for all $x \in X$ and all $g \in G$. 

Equivalently, this means that this diagram commutes:

$$X \times G \xrightarrow{A} \xrightarrow{\varphi \times \text{id}_G} \tilde{X} \times G \xrightarrow{\tilde{A}} \tilde{X}$$
Homomorphisms and isomorphisms

Let \( A : X \times G \to X \) and \( \tilde{A} : \tilde{X} \times G \to \tilde{X} \) be two actions.

**Definition (G-homomorphism)**

A homomorphism \( \varphi : X \to \tilde{X} \) is called a \( G \)-homomorphism or \( G \)-equivariant, if

\[
\varphi(x \cdot g) = \varphi(x) \cdot g \quad \text{for all } x \in X \text{ and all } g \in G.
\]

Equivalently, this means

\[
\varphi(A(x, g)) = \tilde{A}(\varphi(x), g) \quad \text{for all } x \in X \text{ and all } g \in G.
\]
Homomorphisms and isomorphisms

Let $A : X \times G \to X$ and $\tilde{A} : \tilde{X} \times G \to \tilde{X}$ be two actions.

**Definition (G-homomorphism)**

A homomorphism $\varphi : X \to \tilde{X}$ is called a $G$-homomorphism or $G$-equivariant, if

$$\varphi(x \cdot g) = \varphi(x) \cdot g \quad \text{for all } x \in X \text{ and all } g \in G.$$  

Equivalently, this means

$$\varphi(A(x, g)) = \tilde{A}(\varphi(x), g) \quad \text{for all } x \in X \text{ and all } g \in G.$$  

Equivalently, this means that this diagram commutes:

$$
\begin{array}{ccc}
X \times G & \overset{A}{\longrightarrow} & X \\
\downarrow{\varphi \times \text{id}_G} & & \downarrow{\varphi} \\
\tilde{X} \times G & \overset{\tilde{A}}{\longrightarrow} & \tilde{X}
\end{array}
$$
Homomorphisms and isomorphisms

Let $A : X \times G \to X$ and $\tilde{A} : \tilde{X} \times G \to \tilde{X}$ be two actions.

**Definition (G-homomorphism)**

A homomorphism $\varphi : X \to \tilde{X}$ is called a G-homomorphism or G-equivariant, if

$$\varphi(x \cdot g) = \varphi(x) \cdot g \quad \text{for all } x \in X \text{ and all } g \in G.$$  

Equivalently, this means

$$\varphi(A(x, g)) = \tilde{A}(\varphi(x), g) \quad \text{for all } x \in X \text{ and all } g \in G.$$  

Equivalently, this means that this diagram commutes:

\[
\begin{array}{ccc}
X \times G & \xrightarrow{A} & X \\
\varphi \times \text{id}_G & \downarrow & \varphi \\
\tilde{X} \times G & \xrightarrow{\tilde{A}} & \tilde{X}
\end{array}
\]

If $\varphi$ has a G-equiv. inverse, it is called a G-isomorphism.
**Subacts**

Let $G$ act on $X$, i.e. $A : X \times G \rightarrow X$.

**Definition (G-invariant subset, Subact)**

A subset $Y \subseteq X$ is called $G$-invariant, if

$$y \cdot g \in Y \quad \text{for all } y \in Y \text{ and all } g \in G.$$  

The restriction $A|_{Y \times G}$ is then a map to $Y$ and $G$ acts on $Y$. 

---

**Actions and Reps**

Max Neunhöffer

**Group algebras**

**Modules**

**Faithfulness**

**Homomorphisms**

**Subacts**

**Factor acts**

**Extensions and direct sums**

**Indecomposability**

**Problems**

**Ordinary rep. theory**

**Modular rep. theory**

**Permutation groups**

**Matrix and projective groups**

**Orbits**
Subacts

Let $G$ act on $X$, i.e. $A : X \times G \rightarrow X$.

**Definition (G-invariant subset, Subact)**

A subset $Y \subseteq X$ is called $G$-invariant, if

$$y \cdot g \in Y \quad \text{for all } y \in Y \text{ and all } g \in G.$$ 

The restriction $A|_{Y \times G}$ is then a map to $Y$ and $G$ acts on $Y$. If $Y \subseteq X$ is also a substructure of $X$, we call $Y$ a subact (or submodule resp.).
Subacts

Let $G$ act on $X$, i.e. $A : X \times G \rightarrow X$.

Definition (G-invariant subset, Subact)

A subset $Y \subseteq X$ is called $G$-invariant, if

$$ y \cdot g \in Y \quad \text{for all } y \in Y \text{ and all } g \in G. $$

The restriction $A|_{Y \times G}$ is then a map to $Y$ and $G$ acts on $Y$. If $Y \subseteq X$ is also a substructure of $X$, we call $Y$ a subact (or submodule resp.).

Recall: A permutation representation was called transitive if it has no proper subacts.
Subacts

Let $G$ act on $X$, i.e. $A : X \times G \to X$.

**Definition (G-invariant subset, Subact)**

A subset $Y \subseteq X$ is called $G$-invariant, if

$$y \cdot g \in Y \quad \text{for all } y \in Y \text{ and all } g \in G.$$ 

The restriction $A|_{Y \times G}$ is then a map to $Y$ and $G$ acts on $Y$. If $Y \subseteq X$ is also a substructure of $X$, we call $Y$ a subact (or submodule resp.).

Recall: A permutation representation was called transitive if it has no proper subacts.

**Definition (Irreducible/simple module)**

An $\mathbb{F}G$-module $M$ is called irreducible or simple, if it has no submodules except 0 and $M$ itself.
**Factor acts**

Let $G$ act on $X$, i.e. $A : X \times G \to X$.

**Definition (G-invariant partition, factor act)**

Let $X = \bigcup_{i \in I} Y_i$ be partitioned such that

$$\forall i \in I \text{ and } g \in G, \text{ we have } Y_i \cdot g \subseteq Y_j \text{ for some } j \in I.$$

We say that the partition is **G-invariant** and get an action on the set of parts $Y := \{ Y_i \mid i \in I \}$:

$$Y_i \ast g := Y_j \quad \text{if} \quad Y_i \cdot g \subseteq Y_j.$$
**Factor acts**

Let $G$ act on $X$, i.e. $A : X \times G \rightarrow X$.

**Definition (G-invariant partition, factor act)**

Let $X = \bigcup_{i \in I} Y_i$ be partitioned such that

$$\forall i \in I \text{ and } g \in G, \text{ we have } Y_i \cdot g \subseteq Y_j \text{ for some } j \in I.$$ 

We say that the partition is **G-invariant** and get an action on the set of parts $Y := \{Y_i \mid i \in I\}$:

$$Y_i \ast g := Y_j \text{ if } Y_i \cdot g \subseteq Y_j.$$ 

**Recall:** We call a permutation action **primitive**, if it has no non-trivial factor acts.
Let $G$ act on $X$, i.e. $A: X \times G \rightarrow X$.

**Definition (G-invariant partition, factor act)**

Let $X = \bigcup_{i \in I} Y_i$ be partitioned such that

\[ \forall \ i \in I \text{ and } g \in G, \text{ we have } Y_i \cdot g \subseteq Y_j \text{ for some } j \in I. \]

We say that the partition is $G$-invariant and get an action on the set of parts $Y := \{ Y_i \mid i \in I \}$:

\[ Y_i \ast g := Y_j \text{ if } Y_i \cdot g \subseteq Y_j. \]

Recall: We call a permutation action primitive, if it has no non-trivial factor acts.

**Note:** We usually want extra conditions to ensure that $Y$ has the same algebraic structure as $X$ and the new action is a homomorphism of such structures for all $g$. 
Extensions and direct sums

This is only about modules!
Extensions and direct sums

This is only about modules!

Let

\[ 0 \rightarrow W \xrightarrow{i} V \xrightarrow{\pi} U \cong V/W \rightarrow 0 \]

be a module \( V \) with a non-trivial submodule.
Extensions and direct sums

This is only about modules!

Let

\[ 0 \to W \xrightarrow{i} V \xrightarrow{\pi} U \cong V/W \to 0 \]

be a module \( V \) with a non-trivial submodule.

This sequence may or may not be split:

\[ 0 \to W \xrightarrow{i} V \xrightarrow{\pi} U \xleftarrow{r} V \to W \to 0 , \]

i.e. there is \( r : U \to W \) with \( \pi \circ r = \text{id}_U \).
Extensions and direct sums

This is only about modules!

Let

\[ 0 \rightarrow W \xrightarrow{i} V \xrightarrow{\pi} U \cong V/W \rightarrow 0 \]

be a module \( V \) with a non-trivial submodule.

This sequence may or may not be split:

\[ 0 \rightarrow W \xrightarrow{i} V \xrightarrow{\pi} U \xleftarrow{r} U \rightarrow 0 , \]

i.e. there is \( r : U \rightarrow W \) with \( \pi \circ r = \text{id}_U \).

If and only if it is split, the module \( V \) is isomorphic to the direct sum

\[ V \cong W \oplus U. \]
Definition (Indecomposable module)

An $F^G$-module $V$ is called **indecomposable** if it is not isomorphic to a direct sum of two proper submodules. Otherwise it is called **decomposable**.
Indecomposability and semisimplicity

**Definition (Indecomposable module)**

An $\mathbb{F}G$-module $V$ is called **indecomposable** if it is not isomorphic to a direct sum of two proper submodules. Otherwise it is called **decomposable**.

**Lemma (Decomposable implies reducible)**

A **decomposable module is reducible**.
Indecomposability and semisimplicity

**Definition (Indecomposable module)**
An $\mathbb{F}G$-module $V$ is called **indecomposable** if it is not isomorphic to a direct sum of two proper submodules. Otherwise it is called **decomposable**.

**Lemma (Decomposable implies reducible)**
A **decomposable module** is reducible.

**Definition (Semisimple modules and algebras)**
A module is called **semisimple**, if it is isomorphic to a direct sum of simple modules.
Indecomposability and semisimplicity

**Definition (Indecomposable module)**

An $\mathbb{F}G$-module $V$ is called **indecomposable** if it is not isomorphic to a direct sum of two proper submodules. Otherwise it is called **decomposable**.

**Lemma (Decomposable implies reducible)**

A decomposable module is reducible.

**Definition (Semisimple modules and algebras)**

A module is called **semisimple**, if it is isomorphic to a direct sum of simple modules.

An $\mathbb{F}$-algebra $\mathcal{A}$ is called **semisimple**, if every $\mathcal{A}$-module is semisimple.
Ordinary representation theory of groups

For a finite group, the group algebra $\mathbb{C}G$ is semisimple.
Ordinary representation theory of groups

For a finite group, the group algebra $\mathbb{C}G$ is semisimple.

The ordinary representation theory of groups solves:

Problem (Classification of simple modules)

*Classify the isomorphism types of simple $\mathbb{C}G$-modules,*
Ordinary representation theory of groups

For a finite group, the group algebra \( \mathbb{C}G \) is semisimple.

The ordinary representation theory of groups solves:

**Problem (Classification of simple modules)**

*Classify the isomorphism types of simple \( \mathbb{C}G \)-modules, i.e. classify irreducible \( \mathbb{C}G \)-modules up to isomorphism.*
Ordinary representation theory of groups

For a finite group, the group algebra $\mathbb{C}G$ is semisimple.

The ordinary representation theory of groups solves:

**Problem (Classification of simple modules)**

Classify the isomorphism types of simple $\mathbb{C}G$-modules, i.e. classify irreducible $\mathbb{C}G$-modules up to isomorphism.

**Lemma (Characters)**

Two representations

$$R_1 : G \to \text{GL}(V) \quad \text{and} \quad R_2 : G \to \text{GL}(W)$$

afforded by two $\mathbb{C}G$-modules $V$ and $W$ are isomorphic, if and only if their characters $\chi_1 = \text{Tr} \circ R_1$ and $\chi_2 = \text{Tr} \circ R_2$ are equal.
Ordinary representation theory of groups

For a finite group, the group algebra $\mathbb{C}G$ is semisimple.

The ordinary representation theory of groups solves:

Problem (Classification of simple modules)

Classify the isomorphism types of simple $\mathbb{C}G$-modules, i.e. classify irreducible $\mathbb{C}G$-modules up to isomorphism.

Lemma (Characters)

Two representations

$$R_1 : G \rightarrow \text{GL}(V) \quad \text{and} \quad R_2 : G \rightarrow \text{GL}(W)$$

afforded by two $\mathbb{C}G$-modules $V$ and $W$ are isomorphic, if and only if their characters $\chi_1 = \text{Tr} \circ R_1$ and $\chi_2 = \text{Tr} \circ R_2$ are equal.

The two characters $\chi_i : G \rightarrow \mathbb{C}$ are class functions.
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
- Character tables of alternating groups.
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
- Character tables of alternating groups.
- The ATLAS (character tables of simple groups).
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
- Character tables of alternating groups.
- The ATLAS (character tables of simple groups).
- Some generic character tables.
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
- Character tables of alternating groups.
- The ATLAS (character tables of simple groups).
- Some generic character tables.

Still to do:

- Determine character tables for more groups.
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
- Character tables of alternating groups.
- The ATLAS (character tables of simple groups).
- Some generic character tables.

Still to do:

- Determine character tables for more groups.
- Determine more generic tables for whole families of groups.
Research problems in ordinary rep. theory

Already done:

- Character tables of symmetric groups.
- Character tables of alternating groups.
- The ATLAS (character tables of simple groups).
- Some generic character tables.

Still to do:

- Determine character tables for more groups.
- Determine more generic tables for whole families of groups.
- Devise better algorithms to compute tables.
Actions and Reps
Max Neunhöffer

Modular representation theory of groups

F: field with char(F) | |G|, then FG is not semisimple.
Modular representation theory of groups

\( \mathbb{F} \): field with \( \text{char}(\mathbb{F}) \mid |G| \), then \( \mathbb{F}G \) is not semisimple.

The modular rep. theory of groups strives to solve:

**Problem (Classification of simple modules)**

Classify the *isomorphism types* of simple \( \mathbb{F}G \)-modules,
Modular representation theory of groups

\( \mathbb{F} \): field with \( \text{char}(\mathbb{F}) \mid |G| \), then \( \mathbb{F}G \) is not semisimple.

The modular rep. theory of groups strives to solve:

**Problem (Classification of simple modules)**

*Classify the isomorphism types of simple \( \mathbb{F}G \)-modules, i.e. classify irreducible \( \mathbb{F}G \)-modules up to isomorphism.*
Modular representation theory of groups

$F$: field with $\text{char}(F) | |G|$, then $F[G]$ is not semisimple.

The modular rep. theory of groups strives to solve:

### Problem (Classification of simple modules)

*Classify the isomorphism types of simple $F[G]$-modules, i.e. classify irreducible $F[G]$-modules up to isomorphism.*

### Problem (Classification of indecomposable modules)

*Classify the isomorphism types of indecomposable $F[G]$-modules.*
Modular representation theory of groups

\( F \): field with \( \text{char}(F) \mid |G| \), then \( FG \) is not semisimple.

The modular rep. theory of groups strives to solve:

**Problem (Classification of simple modules)**

Classify the isomorphism types of simple \( FG \)-modules, i.e. classify irreducible \( FG \)-modules up to isomorphism.

**Problem (Classification of indecomposable modules)**

Classify the isomorphism types of indecomposable \( FG \)-modules.

**Lemma (Brauer characters)**

Two irreducible representations \( R_1 : G \to \text{GL}(V) \) and \( R_2 : G \to \text{GL}(W) \) afforded by two \( FG \)-modules \( V \) and \( W \) are isomorphic, if and only if their Brauer characters \( \psi_1 \) and \( \psi_2 \) are equal.
Modular representation theory of groups

\( \mathbb{F} \): field with \( \text{char}(\mathbb{F}) \mid |G| \), then \( \mathbb{F}G \) is not semisimple.

The modular rep. theory of groups strives to solve:

**Problem (Classification of simple modules)**

Classify the isomorphism types of simple \( \mathbb{F}G \)-modules, i.e. classify irreducible \( \mathbb{F}G \)-modules up to isomorphism.

**Problem (Classification of indecomposable modules)**

Classify the isomorphism types of indecomposable \( \mathbb{F}G \)-modules.

**Lemma (Brauer characters)**

Two irreducible representations \( R_1 : G \to \text{GL}(V) \) and \( R_2 : G \to \text{GL}(W) \) afforded by two \( \mathbb{F}G \)-modules \( V \) and \( W \) are isomorphic, if and only if their Brauer characters \( \psi_1 \) and \( \psi_2 \) are equal.

The two Brauer characters \( \psi_i \) take values in \( \mathbb{C} \)!
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups $(n \leq 18)$.
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups \((n \leq 18)\).
- Brauer tables of some small alternating groups.
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups \((n \leq 18)\).
- Brauer tables of some small alternating groups.
- Modular ATLAS (Brauer tables of simple groups).
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups \((n \leq 18)\).
- Brauer tables of some small alternating groups.
- Modular ATLAS (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.
Research problems in modular rep. theory

Already done:

- Brauer tables of **some small symmetric groups** \((n \leq 18)\).
- Brauer tables of **some small alternating groups**.
- **Modular ATLAS** (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.

Still to do:

- Determine Brauer tables for **more groups**.
Research problems in modular rep. theory

Already done:

- Brauer tables of **some small** symmetric groups $(n \leq 18)$.
- Brauer tables of **some small** alternating groups.
- **Modular ATLAS** (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.

Still to do:

- Determine Brauer tables for **more groups**.
- Complete the **Modular ATLAS**.
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups \((n \leq 18)\).
- Brauer tables of some small alternating groups.
- Modular ATLAS (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.

Still to do:

- Determine Brauer tables for more groups.
- Complete the Modular ATLAS.
- Classify simple modules of \(\mathbb{F}S_n\).
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups $(n \leq 18)$.
- Brauer tables of some small alternating groups.
- Modular ATLAS (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.

Still to do:

- Determine Brauer tables for more groups.
- Complete the Modular ATLAS.
- Classify simple modules of $\mathbb{F}S_n$.
- Compute the 2-modular Brauer table of the Monster.
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups $(n \leq 18)$.
- Brauer tables of some small alternating groups.
- Modular ATLAS (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.

Still to do:

- Determine Brauer tables for more groups.
- Complete the Modular ATLAS.
- Classify simple modules of $\mathbb{F}S_n$.
- Compute the 2-modular Brauer table of the Monster.
- Find an algorithm to compute a Brauer table???
Research problems in modular rep. theory

Already done:

- Brauer tables of some small symmetric groups ($n \leq 18$).
- Brauer tables of some small alternating groups.
- Modular ATLAS (Brauer tables of simple groups). 1992 by Hiß, Jansen, Lux and Parker: groups up to page 100 in the ATLAS, now some more.

Still to do:

- Determine Brauer tables for more groups.
- Complete the Modular ATLAS.
- Classify simple modules of $\mathbb{F}S_n$.
- Compute the 2-modular Brauer table of the Monster.
- Find an algorithm to compute a Brauer table???
- Classify indecomposable $\mathbb{F}G$-modules???
Problem (Permutation group algorithms)

Given $G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n$ on a computer. Find efficient algorithms to compute with and in $G$: 

Test membership of $\pi \in S_n$ in $G$. 
Find the group order $|G|$. 
Decide whether $G = A_n$ or $G = S_n$ or none. 
Find orbits and blocks of primitivity. 
Find a presentation. 
Find the centre of $G$. 

All of this is done and works well in nearly linear time: runtime is bounded by $C \cdot n \cdot k \cdot \log D(|G|)$. 

Permutation groups
Permutation groups

Problem (Permutation group algorithms)

Given \( G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n \) on a computer. Find efficient algorithms to compute with and in \( G \):

- Test membership of \( \pi \in S_n \) in \( G \).
Permutation groups

Problem (Permutation group algorithms)

Given \( G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n \) on a computer.
Find efficient algorithms to compute with and in \( G \):

- Test membership of \( \pi \in S_n \) in \( G \).
- Find the group order \(|G|\).
Permutation groups

Problem (Permutation group algorithms)

Given $G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n$ on a computer. Find efficient algorithms to compute with and in $G$:

- Test membership of $\pi \in S_n$ in $G$.
- Find the group order $|G|$.
- Decide whether $G = A_n$ or $G = S_n$ or none.
**Permutation groups**

**Problem (Permutation group algorithms)**

Given \( G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n \) on a computer. Find efficient algorithms to compute with and in \( G \):

- **Test membership of** \( \pi \in S_n \) **in** \( G \).
- **Find the group order** \( |G| \).
- **Decide whether** \( G = A_n \) **or** \( G = S_n \) **or none.**
- **Find orbits and blocks of primitivity.**
Permutation groups

Problem (Permutation group algorithms)

Given $G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n$ on a computer. Find efficient algorithms to compute with and in $G$:

- Test membership of $\pi \in S_n$ in $G$.
- Find the group order $|G|$.
- Decide whether $G = A_n$ or $G = S_n$ or none.
- Find orbits and blocks of primitivity.
- Find a presentation.
Permutation groups

Problem (Permutation group algorithms)

Given \( G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n \) on a computer.
Find efficient algorithms to compute with and in \( G \):

- Test membership of \( \pi \in S_n \) in \( G \).
- Find the group order \( |G| \).
- Decide whether \( G = A_n \) or \( G = S_n \) or none.
- Find orbits and blocks of primitivity.
- Find a presentation.
- Find the centre of \( G \).

...All of this is done and works well in nearly linear time: runtime is bounded by \( C \cdot n \cdot k \cdot \log D(|G|) \).
Problem (Permutation group algorithms)

Given $G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n$ on a computer. Find efficient algorithms to compute with and in $G$:

- Test membership of $\pi \in S_n$ in $G$.
- Find the group order $|G|$.
- Decide whether $G = A_n$ or $G = S_n$ or none.
- Find orbits and blocks of primitivity.
- Find a presentation.
- Find the centre of $G$.
- \ldots
Permutation groups

Problem (Permutation group algorithms)

Given $G := \langle g_1, \ldots, g_k \in S_n \rangle \leq S_n$ on a computer. Find efficient algorithms to compute with and in $G$:

- Test membership of $\pi \in S_n$ in $G$.
- Find the group order $|G|$.
- Decide whether $G = A_n$ or $G = S_n$ or none.
- Find orbits and blocks of primitivity.
- Find a presentation.
- Find the centre of $G$.
- ... 

All of this is done and works well in nearly linear time:

runtime is bounded by $C \cdot n \cdot k \cdot \log^D(|G|)$. 

---

*Actions and Reps*
- Max Neunhöffer

*Group algebras*
- Algebras
- Modules

*Faithfulness*

*Homomorphisms*

*Subacts*

*Factor acts*

*Extensions and direct sums*

*Indecomposability*

*Problems*
- Ordinary rep. theory
- Modular rep. theory
- Permutation groups
- Matrix and projective groups
- Orbits
Open questions for permutation groups

Still to do (in nearly linear time):

- Compute the centraliser $C_G(H)$ for some $H < S_n$. 
Open questions for permutation groups

Still to do (in nearly linear time):

- Compute the centraliser $C_G(H)$ for some $H < S_n$.
- Compute the derived subgroup $G'$.
Open questions for permutation groups

Still to do (in nearly linear time):

- Compute the centraliser $C_G(H)$ for some $H < S_n$.
- Compute the derived subgroup $G'$.
- Compute intersections of $G, H < S_n$. 
Open questions for permutation groups

Still to do (in nearly linear time):

- Compute the centraliser $C_G(H)$ for some $H < S_n$.
- Compute the derived subgroup $G'$.
- Compute intersections of $G, H < S_n$.
- Compute conjugacy classes of permutation groups.
Open questions for permutation groups

Still to do (in nearly linear time):

- Compute the centraliser $C_G(H)$ for some $H < S_n$.
- Compute the derived subgroup $G'$.
- Compute intersections of $G, H < S_n$.
- Compute conjugacy classes of permutation groups.
- Test $G, H < S_n$ for conjugacy.
Matrix and projective groups

**Problem (Matrix group algorithms)**

*Given* \( G := \langle M_1, \ldots, M_k \in \text{GL}(\mathbb{F}_q^n) \rangle \leq \text{GL}(\mathbb{F}_q^n) \) *on a computer*.  

**Ultimate goal:** Answer similar questions as for permutation groups.
Matrix and projective groups

Problem (Matrix group algorithms)

Given $G := \langle M_1, \ldots, M_k \in \text{GL}(\mathbb{F}_q^n) \rangle \leq \text{GL}(\mathbb{F}_q^n)$ on a computer.

Ultimate goal: Answer similar questions as for permutation groups.

This is largely unsolved!
Matrix and projective groups

Problem (Matrix group algorithms)

Given $G := \langle M_1, \ldots, M_k \in \text{GL}(F_q^n) \rangle \leq \text{GL}(F_q^n)$ on a computer.

Ultimate goal: Answer similar questions as for permutation groups.

This is largely unsolved!

Problem (Projective group algorithms)

Given $G := \langle \bar{M}_1, \ldots, \bar{M}_k \in \text{PGL}(n, q) \rangle \leq \text{PGL}(n, q)$ on a computer.

Ultimate goal: Answer similar questions as for permutation groups.
Constructive recognition

**Problem (Constructive recognition)**

Let $\mathbb{F}_q$ be the field with $q$ elements and

$$M_1, \ldots, M_k \in \text{GL}(\mathbb{F}_q^n).$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- an algorithm that, given $M \in \text{GL}(\mathbb{F}_q^n)$,
  - decides, whether or not $M \in G$, and,
  - if so, expresses $M$ as word in the $M_i$. 

The runtime should be bounded from above by a polynomial in $n$, $k$, and $\log q$. A Monte Carlo algorithm is enough. (Verification!)
Constructive recognition

Problem (Constructive recognition)

Let $\mathbb{F}_q$ be the field with $q$ elements and

$$M_1, \ldots, M_k \in \text{GL}(\mathbb{F}_q^n).$$

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- an algorithm that, given $M \in \text{GL}(\mathbb{F}_q^n)$,
  - decides, whether or not $M \in G$, and,
  - if so, expresses $M$ as word in the $M_i$.

- The runtime should be bounded from above by a polynomial in $n$, $k$ and $\log q$. 
Constructive recognition

Problem (Constructive recognition)

Let $\mathbb{F}_q$ be the field with $q$ elements and $M_1, \ldots, M_k \in \text{GL}(\mathbb{F}_q^n)$.

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- an algorithm that, given $M \in \text{GL}(\mathbb{F}_q^n)$,
  - decides, whether or not $M \in G$, and,
  - if so, expresses $M$ as word in the $M_i$.

The runtime should be bounded from above by a polynomial in $n$, $k$ and $\log q$.

A Monte Carlo Algorithmus is enough.
### Constructive recognition

#### Problem (Constructive recognition)

Let $\mathbb{F}_q$ be the field with $q$ elements and\n
\[ M_1, \ldots, M_k \in \text{GL}(\mathbb{F}_q^n). \]

Find for $G := \langle M_1, \ldots, M_k \rangle$:

- The group order $|G|$ and
- an algorithm that, given $M \in \text{GL}(\mathbb{F}_q^n)$,\n  - decides, whether or not $M \in G$, and,\n  - if so, expresses $M$ as word in the $M_i$.

The runtime should be bounded from above by a polynomial in $n$, $k$ and $\log q$.

A Monte Carlo Algorithmus is enough. (Verification!)
Recursion: composition trees

We get a tree:

```
    G
   /|
  /  \
N-----H
```

Up arrows: inclusions
Down arrows: homomorphisms
Recursion: composition trees

We get a tree:

\[ \begin{array}{c}
  \text{G} \\
  \downarrow \quad \downarrow \\
  \text{N} \\
  \downarrow \quad \downarrow \\
  \text{N}_1 \quad \text{H}_1 \\
  \downarrow \quad \downarrow \\
  \text{N}_3 \quad \text{H}_3 \\
  \downarrow \quad \downarrow \\
  \text{N}_2 \quad \text{H}_2 \\
  \downarrow \quad \downarrow \\
  \emptyset \quad \emptyset \\
  \downarrow \quad \downarrow \\
  \emptyset \\
  \end{array} \]

Up arrows: inclusions
Down arrows: homomorphisms
Recursion: composition trees

We get a tree:

Up arrows: inclusions
Down arrows: homomorphisms

Old idea, improvements are still being made
Enumerating large orbits

Orbit enumerations play an important role in
Enumerating large orbits

Orbit enumerations play an important role in

- modular representation theory,
Enumerating large orbits

**Orbit enumerations** play an important role in
- modular representation theory,
- permutation group algorithms,
Enumerating large orbits

Orbit enumerations play an important role in

- modular representation theory,
- permutation group algorithms,
- matrix and projective group algorithms,
Enumerating large orbits

Orbit enumerations play an important role in

- modular representation theory,
- permutation group algorithms,
- matrix and projective group algorithms,
- combinatorics,
Enumerating large orbits

**Orbit enumerations** play an important role in

- modular representation theory,
- permutation group algorithms,
- matrix and projective group algorithms,
- combinatorics,
- finite geometry.
Enumerating large orbits

**Orbit enumerations** play an important role in
- modular representation theory,
- permutation group algorithms,
- matrix and projective group algorithms,
- combinatorics,
- finite geometry.

To get a feeling:
- To enumerate an orbit of 1140000 vectors in $\mathbb{F}_2^{760}$ needs around 90 seconds.
Enumerating large orbits

** Orbit enumerations play an important role in **
- modular representation theory,
- permutation group algorithms,
- matrix and projective group algorithms,
- combinatorics,
- finite geometry.

To get a feeling:
- To enumerate an orbit of 1140000 vectors in $\mathbb{F}_2^{760}$ needs around 90 seconds.
- To enumerate 95% of the same orbit with better tricks takes 1.1 seconds.
Enumerating large orbits

Orbit enumerations play an important role in

- modular representation theory,
- permutation group algorithms,
- matrix and projective group algorithms,
- combinatorics,
- finite geometry.

To get a feeling:

- To enumerate an orbit of 1140000 vectors in $\mathbb{F}_2^{760}$ needs around 90 seconds.
- To enumerate 95% of the same orbit with better tricks takes 1.1 seconds.

Finding better ways to enumerate orbits is a current research topic.