Computing resolutions of equivariant modules

Sebastian Posur

University of Siegen

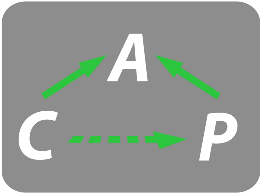
June 13, 2017

Equivariant resolutions

In this computation presentation, we will compute equivariant resolutions of $\mathbb{Z}$-graded modules over the exterior algebra in a **categorical** way.

Equivariant resolutions

In this computation presentation, we will compute equivariant resolutions of $\mathbb{Z}$-graded modules over the exterior algebra in a **categorical** way.



What is CAP?

CAP means **Categories, algorithms, and programming**

What is CAP?

CAP means **Categories, algorithms, and programming** and is a software project implemented in GAP.

What is CAP?

CAP means **Categories, algorithms, and programming** and is a software project implemented in GAP. It is joint work with Sebastian Gutsche, and based on ideas of the `homalg` project (Mohamed Barakat).

What is CAP?

CAP means **Categories, algorithms, and programming** and is a software project implemented in GAP. It is joint work with Sebastian Gutsche, and based on ideas of the `homalg` project (Mohamed Barakat).

- CAP serves as a categorical programming language with categorical operations as primitives.

What is CAP?

CAP means **Categories, algorithms, and programming** and is a software project implemented in GAP. It is joint work with Sebastian Gutsche, and based on ideas of the `homalg` project (Mohamed Barakat).

- CAP serves as a categorical programming language with categorical operations as primitives.
- CAP simplifies the implementation of specific instances of categories

Representations

Let

Representations

Let

- G be a finite group,

Representations

Let

- G be a finite group,
- k be a splitting field for G.

Representations

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Representations

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Maschke's theorem

$$\mathrm{Rep}_k(G) \xrightarrow{\quad\quad\sim\quad\quad} \bigoplus_{i\in\mathrm{Irr}(G)} k\text{-vec}$$

Representations

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Maschke's theorem

$$\mathrm{Rep}_k(G) \xrightarrow{\quad\sim\quad} \bigoplus_{i \in \mathrm{Irr}(G)} k\text{-vec}$$

$$V$$

Representations

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Maschke's theorem

$$\mathrm{Rep}_k(G) \xrightarrow{\quad\sim\quad} \bigoplus_{i \in \mathrm{Irr}(G)} k\text{-vec}$$

$$V \simeq \bigoplus_{i \in \mathrm{Irr}(G)} a_i V^i$$

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Maschke's theorem

$$\mathrm{Rep}_k(G) \xrightarrow{\quad\sim\quad} \bigoplus_{i \in \mathrm{Irr}(G)} k\text{-vec}$$

$$V \simeq \bigoplus_{i \in \mathrm{Irr}(G)} a_i V^i$$

Representations

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Maschke's theorem

$$\mathrm{Rep}_k(G) \xrightarrow{\sim} \bigoplus_{i \in \mathrm{Irr}(G)} k\text{-vec}$$

$$V \simeq \bigoplus_{i \in \mathrm{Irr}(G)} a_i V^i \longmapsto (a_i)_i$$

Representations

Let

- G be a finite group,
- k be a splitting field for G.

We denote by $\mathrm{Rep}_k(G)$ the category of representations of G.

Maschke's theorem

$$\begin{array}{ccc} \mathrm{Rep}_k(G) & \xrightarrow{\sim} & \bigoplus_{i \in \mathrm{Irr}(G)} k\text{-vec} \\ V \simeq \bigoplus_{i \in \mathrm{Irr}(G)} a_i V^i & \longmapsto & (a_i)_i \end{array}$$

We call $\bigoplus_{i \in \mathrm{Irr}(G)} k\text{-vec}$ a **skeletal model** for $\mathrm{Rep}_k(G)$.

Representations

Operations of abelian categories like

Representations

Operations of abelian categories like

- kernels, cokernels,

Representations

Operations of abelian categories like

- kernels, cokernels,
- pullbacks, pushouts,

Representations

Operations of abelian categories like

- kernels, cokernels,
- pullbacks, pushouts,
- diagram chases,

Representations

Operations of abelian categories like

- kernels, cokernels,
- pullbacks, pushouts,
- diagram chases,
- spectral sequences

Representations

Operations of abelian categories like

- kernels, cokernels,
- pullbacks, pushouts,
- diagram chases,
- spectral sequences

are easier to compute in the skeletal model since the matrices encoding morphisms are smaller.

Representations

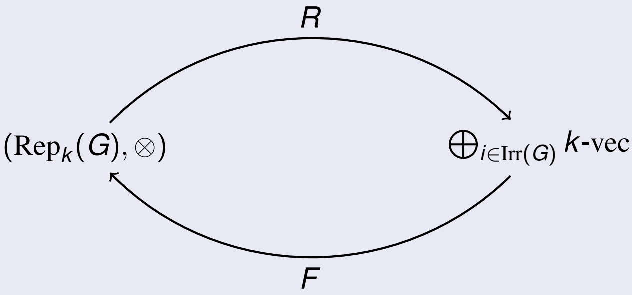
Operations of abelian categories like

- kernels, cokernels,
- pullbacks, pushouts,
- diagram chases,
- spectral sequences

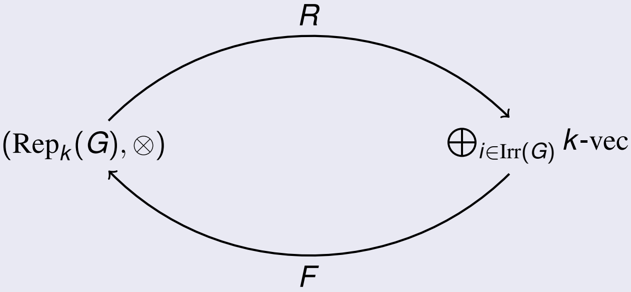
are easier to compute in the skeletal model since the matrices encoding morphisms are smaller.

What about tensor products?

Structure transport

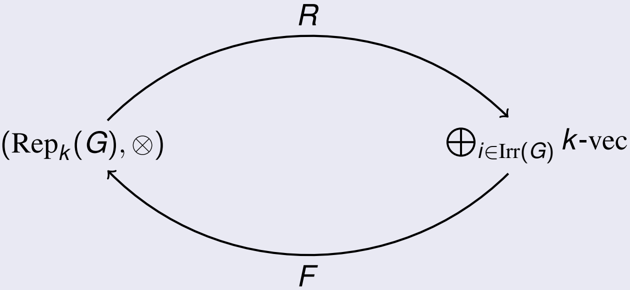


Structure transport



$$\otimes_{\text{skeletal}} := R \circ \otimes \circ (F \times F)$$

Structure transport



$$\otimes_{\text{skeletal}} := R \circ \otimes \circ (F \times F)$$

Also works for associators $A \otimes (B \otimes C) \to (A \otimes B) \otimes C$

Internal algebra

Data structure of an algebra internal to a tensor category

Internal algebra

Data structure of an algebra internal to a tensor category

- $\mu_E : E \otimes E \to E$

Internal algebra

Data structure of an algebra internal to a tensor category

- $\mu_E : E \otimes E \to E$
- Associativity constraint

Internal algebra

Data structure of an algebra internal to a tensor category

- $\mu_E : E \otimes E \to E$
- Associativity constraint

Data structure of a module internal to a tensor category

Internal algebra

Data structure of an algebra internal to a tensor category

- $\mu_E : E \otimes E \to E$
- Associativity constraint

Data structure of a module internal to a tensor category

- $\mu_M : M \otimes E \to M$

Internal algebra

Data structure of an algebra internal to a tensor category

- $\mu_E : E \otimes E \to E$
- Associativity constraint

Data structure of a module internal to a tensor category

- $\mu_M : M \otimes E \to M$
- Associativity constraint

How to work with internal modules?

Free modules

B

How to work with internal modules?

Free modules

$$B \otimes E$$

How to work with internal modules?

Free modules

$$(B \otimes E) \otimes E$$

$$B \otimes E$$

How to work with internal modules?

Free modules

$$
\begin{array}{c}
(B \otimes E) \otimes E \\
\\
\\
B \otimes (E \otimes E) \\
\big\downarrow B \otimes \mu_E \\
B \otimes E
\end{array}
$$

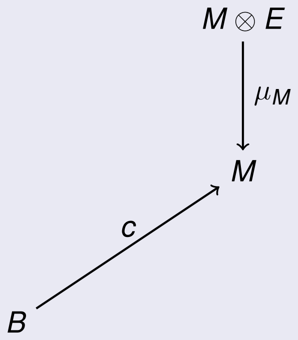
How to work with internal modules?

Free modules

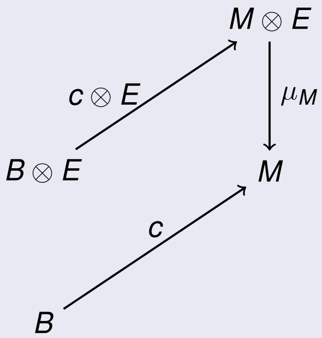
$$\begin{array}{c} (B \otimes E) \otimes E \\ \wr \big\downarrow \text{AssociatorLeftToRight}(B, E, E) \\ B \otimes (E \otimes E) \\ \big\downarrow B \otimes \mu_E \\ B \otimes E \end{array}$$

How to work with internal modules?

Universal property of free modules

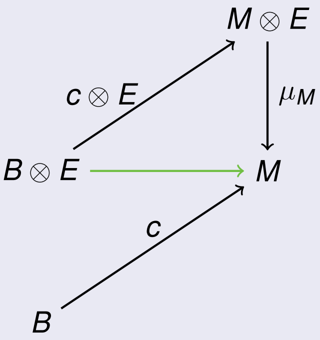


Universal property of free modules



How to work with internal modules?

Universal property of free modules



Equivariant cohomology table

H^2 :	$\chi_4 + 2\chi_2$	$\chi_1 + \chi_3$	.	.	.	.	.	.
H^1 :	.	.	χ_1	χ_2	χ_4	χ_5	.	.
H^0 :	.	.	.	.	.	.	$\chi_3 + \chi_5$	$2\chi_4 + \chi_2$
p	-4	-3	-2	-1	0	1	2	3

The equivariant cohomology table of a rank 2 vector bundle.

Download CAP

Download CAP

http://homalg-project.github.io/CAP_project/