A constructive model for coherent sheaves over a normal toric variety

Sebastian Gutsche

(j/w Sebastian Posur)

University of Siegen

Paderborn, March 6, 2018

Outline

Outline

1. Computable Categories

2. Model for coherent sheaves over normal toric varieties

Section 1

Computable Categories

Computable categories

Computable categories

A category becomes computable through

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

1 2 1

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

1 2 1

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

$$1 \xrightarrow[\begin{pmatrix} 1 & 2 \end{pmatrix}]{} 2 \xrightarrow[\begin{pmatrix} 3 \\ 4 \end{pmatrix}]{} 1$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

$$1 \xrightarrow[\begin{pmatrix} 1 & 2 \end{pmatrix}]{} 2 \xrightarrow[\begin{pmatrix} 3 \\ 4 \end{pmatrix}]{} 1$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$

$$\begin{tikzcd} 1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] & 1 \end{tikzcd}$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$

$$\begin{tikzcd} 1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] & 1 \end{tikzcd}$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$

$$\begin{tikzcd}
1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left] \arrow[loop below, "(1)"] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] \arrow[loop below, "\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}"] & 1 \arrow[loop below, "(1)"]
\end{tikzcd}$$

Computable categories

A category becomes computable through

- Data structures for *objects* and *morphisms*
- Algorithms to compute the *composition* of morphisms and *identity morphisms* of objects

Finitely generated $\mathbb{Q}$-vector spaces (skeletal)

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} = (11)$$

$$\begin{tikzcd}
1 \arrow[r, "\begin{pmatrix} 1 & 2 \end{pmatrix}"'] \arrow[rr, bend left] \arrow[loop below, "(1)"] & 2 \arrow[r, "\begin{pmatrix} 3 \\ 4 \end{pmatrix}"'] \arrow[loop below, "\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}"] & 1 \arrow[loop below, "(1)"]
\end{tikzcd}$$

Categorical operations

Some categorical operations in abelian categories

Categorical operations

Some categorical operations in abelian categories

- Zero morphisms

Categorical operations

Some categorical operations in abelian categories

- Zero morphisms
- Addition and subtraction of morphisms

Categorical operations

Some categorical operations in abelian categories

- Zero morphisms
- Addition and subtraction of morphisms
- Direct sums

Categorical operations

Some categorical operations in abelian categories

- Zero morphisms
- Addition and subtraction of morphisms
- Direct sums
- Kernels and Cokernels of morphisms

Categorical operations

Some categorical operations in abelian categories

- Zero morphisms
- Addition and subtraction of morphisms
- Direct sums
- Kernels and Cokernels of morphisms
- ...

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$.

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$.

$$A \xrightarrow{\varphi} B$$

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

$$A \xrightarrow{\varphi} B$$

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

... one needs an object $\ker \varphi$,

$$\begin{tikzcd} \ker \varphi & & \\ & A \arrow[r, "\varphi"] & B \end{tikzcd}$$

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ . . .

. . . one needs an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,

$$\begin{tikzcd} \ker\varphi \arrow[dr, hook, "\kappa"] & & \\ & A \arrow[r, "\varphi"] & B \end{tikzcd}$$

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ …

… one needs an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ

$$
\begin{tikzcd}
\ker\varphi \arrow[dr, hook, "\kappa"] \arrow[drr, bend left, "0"] & & \\
& A \arrow[r, "\varphi"] & B \\
T \arrow[ur, "\tau"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}
$$

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ ...

... one needs an object $\ker\varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ
a *unique* morphism $\lambda = \mathrm{KernelLift}(\varphi, \tau)$

$$
\begin{tikzcd}
\ker\varphi \arrow[dr, hook, "\kappa"] \arrow[drr, bend left, "0"] & & \\
& A \arrow[r, "\varphi"] & B \\
T \arrow[uu, dashed, "\lambda"] \arrow[ur, "\tau"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}
$$

Implementation of the kernel

Let $\varphi \in \mathrm{Hom}(A, B)$. To fully describe the kernel of φ . . .

. . . one needs an object $\ker \varphi$,
its embedding $\kappa = \mathrm{KernelEmbedding}(\varphi)$,
and for every test morphism τ
a *unique* morphism $\lambda = \mathrm{KernelLift}(\varphi, \tau)$, such that

$$
\begin{tikzcd}
\ker\varphi \arrow[dr, hook, "\kappa"] \arrow[drr, bend left, "0"] & & \\
\circlearrowleft & A \arrow[r, "\varphi"] & B \\
T \arrow[uu, dashed, "\lambda"] \arrow[ur, "\tau"] \arrow[urr, bend right, "0"'] & &
\end{tikzcd}
$$

The CAP - project

CAP - Categories, Algorithms, Programming

The CAP - project

CAP - Categories, Algorithms, Programming

CAP is a framework written in GAP

The CAP - project

CAP - Categories, Algorithms, Programming

CAP is a framework written in GAP to implement computable categories and provides

The CAP - project

CAP - Categories, Algorithms, Programming

CAP is a framework written in GAP to implement computable categories and provides

- specifications of categorical operations,

The CAP - project

CAP - Categories, Algorithms, Programming

CAP is a framework written in GAP to implement computable categories and provides

- specifications of categorical operations,
- generic algorithms based on basic categorical operations,

The CAP - project

CAP - Categories, Algorithms, Programming

CAP is a framework written in GAP to implement computable categories and provides

- specifications of categorical operations,
- generic algorithms based on basic categorical operations,
- a categorical programming language having categorical operations as syntax elements.

The CAP - project

CAP - Categories, Algorithms, Programming

CAP is a framework written in GAP to implement computable categories and provides

- specifications of categorical operations,
- generic algorithms based on basic categorical operations,
- a categorical programming language having categorical operations as syntax elements.

Computing the intersection

Let $M_1 \subseteq N$ and $M_2 \subseteq N$ subobjects.

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
M_1 \arrow[dr, hook, "\iota_1"] & \\
& N \\
M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 & & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$
\begin{tikzcd}
 & & M_1 \arrow[rd, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[rru, two heads, "\pi_1"] & & & N \\
 & & M_2 \arrow[ru, hook, "\iota_2"'] &
\end{tikzcd}
$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] & & N \\
 & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$
\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}
$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$
- $\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$
\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] \arrow[dr, two heads, "\pi_2"'] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}
$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$
- $\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

Computing the intersection

Let $M_1 \hookrightarrow N$ and $M_2 \hookrightarrow N$ subobjects.
Compute their intersection $\gamma : M_1 \cap M_2 \hookrightarrow N$.

$$
\begin{tikzcd}
 & & M_1 \arrow[dr, hook, "\iota_1"] & \\
M_1 \cap M_2 \arrow[r, hook, "\kappa"] & M_1 \oplus M_2 \arrow[ur, two heads, "\pi_1"] \arrow[dr, two heads, "\pi_2"'] \arrow[rr, "\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2"] & & N \\
 & & M_2 \arrow[ur, hook, "\iota_2"'] &
\end{tikzcd}
$$

- $\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left((M_1, M_2), i\right)$, $i = 1, 2$
- $\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$
- $\kappa := \mathrm{KernelEmbedding}(\varphi)$
- $\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$$\pi_i := \text{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right),\ i = 1, 2$$

$$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$$

$$\kappa := \text{KernelEmbedding}\left(\varphi\right)$$

$$\gamma := \iota_1 \circ \pi_1 \circ \kappa$$

Translation to CAP

$\pi_i := \text{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

$\kappa := \text{KernelEmbedding}\left(\varphi\right)$

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$\pi_i := \text{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

```
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
```

$\kappa := \text{KernelEmbedding}\left(\varphi\right)$

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$\pi_i := \text{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right)$, $i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

```
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
```

$\kappa := \text{KernelEmbedding}\left(\varphi\right)$

```
kappa := KernelEmbedding( phi );
```

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

Translation to CAP

$\pi_i := \mathrm{ProjectionInFactorOfDirectSum}\left(\left(M_1, M_2\right), i\right),\ i = 1, 2$

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
```

$\varphi := \iota_1 \circ \pi_1 - \iota_2 \circ \pi_2$

```
lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );
```

$\kappa := \mathrm{KernelEmbedding}\left(\varphi\right)$

```
kappa := KernelEmbedding( phi );
```

$\gamma := \iota_1 \circ \pi_1 \circ \kappa$

```
gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );

kappa := KernelEmbedding( phi );

gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

lambda := PostCompose( iota1, pi1 );
phi := lambda - PostCompose( iota2, pi2 );

kappa := KernelEmbedding( phi );

gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
Schnitt := function( iota1, iota2 )

  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );

  kappa := KernelEmbedding( phi );

  gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
Schnitt := function( iota1, iota2 )

  M1 := Source( iota1 );
  M2 := Source( iota2 );

  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );

  kappa := KernelEmbedding( phi );

  gamma := PostCompose( lambda, kappa );
```

Translation to CAP

```
Schnitt := function( iota1, iota2 )

  M1 := Source( iota1 );
  M2 := Source( iota2 );
  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );
  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );
  kappa := KernelEmbedding( phi );
  gamma := PostCompose( lambda, kappa );
  return gamma;
end;
```

Translation to CAP

```
Schnitt := function( iota1, iota2 )

  local M1, M2, pi1, pi2, lambda, phi, kappa, gamma;

  M1 := Source( iota1 );
  M2 := Source( iota2 );

  pi1 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 1 );
  pi2 := ProjectionInFactorOfDirectSum( [ M1, M2 ], 2 );

  lambda := PostCompose( iota1, pi1 );
  phi := lambda - PostCompose( iota2, pi2 );

  kappa := KernelEmbedding( phi );

  gamma := PostCompose( lambda, kappa );

  return gamma;
end;
```

Section 2

Model for coherent sheaves over normal toric varieties

Coherent sheaves

Coherent sheaves

Projective space

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$,

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$,

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$,
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$,
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$

In the language of category theory:

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = \left(K^n / K^*\right) - \overline{\{0\}}$,
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$

In the language of category theory:

$$\mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$,
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$

In the language of category theory:

S-mod $\qquad\qquad\qquad \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$,
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$

In the language of category theory:

S-mod $\qquad\qquad \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$,
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading

In the language of category theory:

$S\text{-mod} \qquad \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}\,(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading

In the language of category theory:

S-mod $\qquad\qquad \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}\,(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K\,[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading

In the language of category theory:

$S\text{-}\mathrm{grmod}_{\mathbb{Z}}$ $\qquad\qquad$ $\mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading

In the language of category theory:

$S\text{-}\mathrm{grmod}_{\mathbb{Z}}$ $\qquad\qquad$ $\mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n / K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, \underline{x_n]}$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:

$S\text{-grmod}_{\mathbb{Z}}$ $\qquad\qquad$ $\mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}_{\mathbb{Z}}^{0} \qquad\qquad \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}\,(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K\,[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}^0_{\mathbb{Z}} \qquad\qquad \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}\,(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}^0_{\mathbb{Z}} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Projective space

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$, $K^* \cong \operatorname{Hom}(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-}\mathrm{grmod}_{\mathbb{Z}} / S\text{-}\mathrm{grmod}^0_{\mathbb{Z}} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Projective space $\mathbb{P}^{n-1} = (K^n/K^*) - \overline{\{0\}}$, $K^* \cong \mathrm{Hom}\,(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K\,[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}_{\mathbb{Z}}^{0} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n / K^*) - \overline{\{0\}}$, $K^* \cong \operatorname{Hom}(\mathbb{Z}, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}_{\mathbb{Z}}^{0} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - \overline{\{0\}}$, $G' \cong \mathrm{Hom}\,(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K\,[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}}/S\text{-grmod}_{\mathbb{Z}}^{0} \xrightarrow{\ \sim\ } \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a $\mathbb{Z}$-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}^0_{\mathbb{Z}} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a G-grading modulo modules that are only supported on $\overline{\{0\}}$.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}_{\mathbb{Z}}^{0} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}\,(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a G-grading modulo modules that are only supported on Z.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}^0_{\mathbb{Z}} \xrightarrow{\sim} \mathfrak{Coh}\left(\mathbb{P}^{n-1}\right)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}\,(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K\,[x_1, \ldots, x_n]$ with a G-grading modulo modules that are only supported on Z.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_{\mathbb{Z}} / S\text{-grmod}^0_{\mathbb{Z}} \xrightarrow{\sim} \mathfrak{Coh}\,(X)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a G-grading modulo modules that are only supported on Z.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \xrightarrow{\sim} \mathfrak{Coh}(X)$$

Coherent sheaves

Normal toric variety (smooth)

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a G-grading modulo modules that are only supported on Z.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \xrightarrow{\sim} \mathfrak{Coh}(X)$$

Coherent sheaves

Normal toric variety

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a G-grading modulo modules that are only supported on Z.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \xrightarrow{\sim} \mathfrak{Coh}(X)$$

Coherent sheaves

Normal toric variety

- Toric variety $X = (K^n / G') - Z$, $G' \cong \mathrm{Hom}\,(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K\,[x_1, \ldots, x_n]$ with a G-grading modulo modules that sheafify to zero.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \xrightarrow{\sim} \mathfrak{Coh}\,(X)$$

Coherent sheaves

Normal toric variety

- Toric variety $X = (K^n/G') - Z$, $G' \cong \mathrm{Hom}(G, K^*)$
- Coherent sheaves correspond to f. g. modules over $S := K[x_1, \ldots, x_n]$ with a G-grading modulo modules that sheafify to zero.

In the language of category theory:
Equivalence of categories

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \xrightarrow{\sim} \mathfrak{Coh}(X)$$

Computability of $S\text{-grmod}_G / S\text{-grmod}_G^0$?

Serre quotients

Serre quotient

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory.

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory. The **Serre quotient** $\mathcal{A}/\mathcal{C}$ is an abelian category with

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory. The **Serre quotient** $\mathcal{A}/\mathcal{C}$ is an abelian category with

- $\mathrm{Obj}_{\mathcal{A}/\mathcal{C}} := \mathrm{Obj}_{\mathcal{A}}$

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory. The **Serre quotient** $\mathcal{A}/\mathcal{C}$ is an abelian category with

- $\mathrm{Obj}_{\mathcal{A}/\mathcal{C}} := \mathrm{Obj}_{\mathcal{A}}$
- $\mathrm{Hom}_{\mathcal{A}/\mathcal{C}}(A, B) :=$

$$\left\{ \begin{tikzcd} A \arrow[rr, dashed] & & B \\ & X \arrow[ul, "\psi"] \arrow[ur, "\varphi"'] & \end{tikzcd} \;\middle|\; \qquad \right\}$$

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory. The **Serre quotient** $\mathcal{A}/\mathcal{C}$ is an abelian category with

- $\mathrm{Obj}_{\mathcal{A}/\mathcal{C}} := \mathrm{Obj}_{\mathcal{A}}$
- $\mathrm{Hom}_{\mathcal{A}/\mathcal{C}}(A, B) :=$

$$\left\{ \begin{tikzcd} A \arrow[rr, dashed] & & B \\ & X \arrow[ul, "\psi"] \arrow[ur, "\varphi"'] & \end{tikzcd} \;\middle|\; \operatorname{coker}(\psi) \in \mathcal{C} \right\}$$

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory. The **Serre quotient** $\mathcal{A}/\mathcal{C}$ is an abelian category with

- $\mathrm{Obj}_{\mathcal{A}/\mathcal{C}} := \mathrm{Obj}_{\mathcal{A}}$
- $\mathrm{Hom}_{\mathcal{A}/\mathcal{C}}(A, B) :=$

$$\left\{ \begin{tikzcd} A \arrow[rr, dashed] & & B \\ & X \arrow[ul, "\psi"] \arrow[ur, "\varphi"'] & \end{tikzcd} \;\middle|\; \begin{array}{l} \mathrm{coker}(\psi) \in \mathcal{C} \\ \varphi(\ker(\psi)) \in \mathcal{C} \end{array} \right\}$$

Serre quotients

Serre quotient

Let $\mathcal{A}$ be an abelian category and $\mathcal{C}$ a thick subcategory. The **Serre quotient** $\mathcal{A}/\mathcal{C}$ is an abelian category with

- $\mathrm{Obj}_{\mathcal{A}/\mathcal{C}} := \mathrm{Obj}_{\mathcal{A}}$
- $\mathrm{Hom}_{\mathcal{A}/\mathcal{C}}(A, B) :=$

$$\left\{ \begin{tikzcd} A \arrow[rr, dashed] & & B \\ & X \arrow[ul, "\psi"] \arrow[ur, "\varphi"'] & \end{tikzcd} \;\middle|\; \begin{array}{l} \mathrm{coker}(\psi) \in \mathcal{C} \\ \varphi(\ker(\psi)) \in \mathcal{C} \end{array} \right\} \Big/ \sim$$

Composition in the Serre quotient

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$
\begin{tikzcd}
A \arrow[rr, dashed] & & B \arrow[rr, dashed] & & C \\
& X \arrow[ul] \arrow[ur] & & Y \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$\begin{tikzcd}
A \arrow[rr, dashed] & & B \arrow[rr, dashed] & & C \\
& X \arrow[ul] \arrow[ur] & & Y \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$\begin{tikzcd}
A \arrow[rr, dashed] & & B \arrow[rr, dashed] & & C \\
& X \arrow[ul] \arrow[ur] & & Y \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$\begin{tikzcd}
A \arrow[rr, dashed, tail] & & B \arrow[rr, dashed, tail] & & C \\
& X \arrow[ul, tail] \arrow[ur, tail] & & Y \arrow[ul, tail] \arrow[ur, tail] & \\
& & Z \arrow[ul, tail] \arrow[ur, tail] & &
\end{tikzcd}$$

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$
\begin{tikzcd}
A \arrow[rr, dashed] & & B \arrow[rr, dashed] & & C \\
& X \arrow[ul] \arrow[ur] & & Y \arrow[ul] \arrow[ur] & \\
& & Z \arrow[ul] \arrow[ur] & &
\end{tikzcd}
$$

`FiberProduct`: Algorithm for intersection

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$\begin{tikzcd}
A \arrow[rr, dashed] & & B \arrow[rr, dashed] & & C \\
& X \arrow[ul] \arrow[ur] & & Y \arrow[ul] \arrow[ur] & \\
& & Z \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Composition in the Serre quotient

Composition in the Serre quotient $\mathcal{A}/\mathcal{C}$

$$\begin{tikzcd}
A \arrow[rr, dashed] & & B \arrow[rr, dashed] & & C \\
& X \arrow[ul] \arrow[ur] & & Y \arrow[ul] \arrow[ur] & \\
& & Z \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Composition only by computations in $\mathcal{A}$!

Computability of toric coherent sheaves

Computability of toric coherent sheaves

Theorem (Barakat, Lange-Hegermann)

Is $\mathcal{A}$ *computable* abelian and $\mathcal{C}$ *decidable*,

Computability of toric coherent sheaves

Theorem (Barakat, Lange-Hegermann)

Is $\mathcal{A}$ *computable* abelian and $\mathcal{C}$ *decidable*, then $\mathcal{A}/\mathcal{C}$ is computable abelian.

Computability of toric coherent sheaves

Theorem (Barakat, Lange-Hegermann)

Is $\mathcal{A}$ *computable* abelian and $\mathcal{C}$ *decidable*, then $\mathcal{A}/\mathcal{C}$ is computable abelian.

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \cong \mathfrak{Coh}\,(X)\ ?$$

Computability of toric coherent sheaves

Theorem (Barakat, Lange-Hegermann)

Is $\mathcal{A}$ *computable* abelian and $\mathcal{C}$ *decidable*, then $\mathcal{A}/\mathcal{C}$ is computable abelian.

$$S\text{-grmod}_G / S\text{-grmod}_G^0 \cong \mathfrak{Coh}(X) \text{ ?}$$

Decidability of $S\text{-}\mathrm{grmod}_G^0$

Every toric variety X with Cox ring S has a finite affine cover $\{U_\tau\}$, defined by the orbits of the torus acting on X, and naturally indexed by monomials $\tau \in \mathrm{Mon}\,(S)$.

Decidability of $S\text{-}\mathrm{grmod}_G^0$

Every toric variety X with Cox ring S has a finite affine cover $\{U_\tau\}$, defined by the orbits of the torus acting on X, and naturally indexed by monomials $\tau \in \mathrm{Mon}\,(S)$.

Let $M \in S\text{-}\mathrm{grmod}_G$.

Decidability of $S\text{-}\mathrm{grmod}^0_G$

Every toric variety X with Cox ring S has a finite affine cover $\{U_\tau\}$, defined by the orbits of the torus acting on X, and naturally indexed by monomials $\tau \in \mathrm{Mon}(S)$.

Let $M \in S\text{-}\mathrm{grmod}_G$. We have $M \in S\text{-}\mathrm{grmod}^0_G$ iff

$$\Gamma\left(\widetilde{M}, U_\tau\right)$$

Decidability of $S\text{-}\mathrm{grmod}_G^0$

Every toric variety X with Cox ring S has a finite affine cover $\{U_\tau\}$, defined by the orbits of the torus acting on X, and naturally indexed by monomials $\tau \in \mathrm{Mon}(S)$.

Let $M \in S\text{-}\mathrm{grmod}_G$. We have $M \in S\text{-}\mathrm{grmod}_G^0$ iff

$$\Gamma\left(\widetilde{M}, U_\tau\right) = (M_\tau)_0$$

Decidability of $S\text{-grmod}_G^0$

Every toric variety X with Cox ring S has a finite affine cover $\{U_\tau\}$, defined by the orbits of the torus acting on X, and naturally indexed by monomials $\tau \in \mathrm{Mon}(S)$.

Let $M \in S\text{-grmod}_G$. We have $M \in S\text{-grmod}_G^0$ iff

$$\Gamma\left(\widetilde{M}, U_\tau\right) = (M_\tau)_0 = 0 \text{ for all } U_\tau.$$

Decidability of $S\text{-}\mathrm{grmod}_G^0$

Every toric variety X with Cox ring S has a finite affine cover $\{U_\tau\}$, defined by the orbits of the torus acting on X, and naturally indexed by monomials $\tau \in \mathrm{Mon}(S)$.

Let $M \in S\text{-}\mathrm{grmod}_G$. We have $M \in S\text{-}\mathrm{grmod}_G^0$ iff

$$\Gamma\left(\widetilde{M}, U_\tau\right) = (M_\tau)_0 = 0 \text{ for all } U_\tau.$$

Given a presentation of M, how to compute a presentation of $(M_\tau)_0$ for a given τ?

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
   1. Compute inequalities for the cone $\mathrm{Mon}\left((S_\tau)_0\right)$.

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
   1. Compute inequalities for the cone $\mathrm{Mon}\left((S_\tau)_0\right)$.
   2. Compute the generators as HILBERT basis of $\mathrm{Mon}\left((S_\tau)_0\right)$.

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
2. Compute $(S_\tau)_0$-generators of $(S(\gamma)_\tau)_0$, $\gamma \in \{\alpha_i, \beta_j\}$

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i\in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j\in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
2. Compute $(S_\tau)_0$-generators of $(S(\gamma)_\tau)_0\,,\ \gamma \in \{\alpha_i,\ \beta_j\}$

$$\mathrm{Mon}\left((S(\gamma)_\tau)_0\right) = \underbrace{P'}_{\text{finite}} + \mathrm{Mon}\left((S_\tau)_0\right)$$

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j \in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
2. Compute $(S_\tau)_0$-generators of $(S(\gamma)_\tau)_0$, $\gamma \in \{\alpha_i, \beta_j\}$

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i\in I} S(\alpha_i) \xrightarrow{\varphi} \prod_{j\in J} S(\beta_j) \longrightarrow M \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
2. Compute $(S_\tau)_0$-generators of $(S(\gamma)_\tau)_0$, $\gamma \in \{\alpha_i, \beta_j\}$
3. Compute $(\varphi_\tau)_0$ using the embedding $P' \subseteq \mathrm{Mon}(S_\tau)$ by computing representations for $\varphi_\tau(p')$ for all $p' \in P'$.

Presentation of $(M_\tau)_0$

Strategy for computing a presentation of $(M_\tau)_0$ from a free graded presentation of M:

$$\prod_{i \in I} \left(S\left(\alpha_i\right)_\tau \right) \xrightarrow{(\varphi_\tau)_0} \prod_{j \in J} \left(S\left(\beta_j\right)_\tau \right)_0 \longrightarrow (M_\tau)_0 \longrightarrow 0.$$

Algorithm

1. Compute algebra generators for $(S_\tau)_0$
2. Compute $(S_\tau)_0$-generators of $\left(S\left(\gamma\right)_\tau\right)_0$, $\gamma \in \left\{\alpha_i, \beta_j\right\}$
3. Compute $(\varphi_\tau)_0$ using the embedding $P' \subseteq \mathrm{Mon}\left(S_\tau\right)$

Coherent sheaves

So $S\text{-}\mathrm{grmod}^0_G$ is decidable

Coherent sheaves

So $S\text{-}\mathrm{grmod}_G^0$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}_G^0 \cong \mathfrak{Coh}\,(X)$$

computable abelian!

Coherent sheaves

So $S\text{-}\mathrm{grmod}_G^0$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}_G^0 \cong \mathfrak{Coh}\,(X)$$

computable abelian!

We can apply algorithms for abelian categories to coherent sheaves over normal toric varieties:

Coherent sheaves

So $S\text{-}\mathrm{grmod}_G^0$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}_G^0 \cong \mathfrak{Coh}\,(X)$$

computable abelian!

We can apply algorithms for abelian categories to coherent sheaves over normal toric varieties:

- Homology

Coherent sheaves

So $S\text{-}\mathrm{grmod}^0_G$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}^0_G \cong \mathfrak{Coh}\,(X)$$

computable abelian!

We can apply algorithms for abelian categories to coherent sheaves over normal toric varieties:

- Homology
- Diagram chases

Coherent sheaves

So $S\text{-}\mathrm{grmod}_G^0$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}_G^0 \cong \mathfrak{Coh}\,(X)$$

computable abelian!

We can apply algorithms for abelian categories to coherent sheaves over normal toric varieties:

- Homology
- Diagram chases
- Spectral sequences

Coherent sheaves

So $S\text{-}\mathrm{grmod}_G^0$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}_G^0 \cong \mathfrak{Coh}\,(X)$$

computable abelian!

We can apply algorithms for abelian categories to coherent sheaves over normal toric varieties:

- Homology
- Diagram chases
- Spectral sequences
- Purity filtration

Coherent sheaves

So $S\text{-}\mathrm{grmod}_G^0$ is decidable and therefore

$$S\text{-}\mathrm{grmod}_G / S\text{-}\mathrm{grmod}_G^0 \cong \mathfrak{Coh}\,(X)$$

computable abelian!

We can apply algorithms for abelian categories to coherent sheaves over normal toric varieties:

- Homology
- Diagram chases
- Spectral sequences
- Purity filtration
- ...

Workshop: CAP Days 2018

Workshop: CAP Days 2018

August 28 - 31, 2018 at University of Siegen, featuring

Workshop: CAP Days 2018

August 28 - 31, 2018 at University of Siegen, featuring

- Talks on ongoing research

Workshop: CAP Days 2018

August 28 - 31, 2018 at University of Siegen, featuring

- Talks on ongoing research
- Tutorials

Workshop: CAP Days 2018

August 28 - 31, 2018 at University of Siegen, featuring

- Talks on ongoing research
- Tutorials
- Coding Sprint

Workshop: CAP Days 2018

August 28 - 31, 2018 at University of Siegen, featuring

- Talks on ongoing research
- Tutorials
- Coding Sprint

More information and registration:

`https://homalg-project.github.io/capdays-2018/`