Computing GIT-Fans with Symmetry

Janko Boehm
joint with S. Keicher, Y. Ren

Technische Universität Kaiserslautern

29 January 2016

Quotients

Quotients of algebraic varieties by algebraic groups play an important role in constructing moduli spaces.

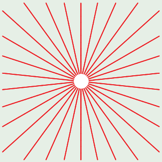
Quotients

Quotients of algebraic varieties by algebraic groups play an important role in constructing moduli spaces.

Example

The moduli space of 1-dimensional vector subspaces of $\mathbb{C}^2$ is

$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$



$$\mathbb{P}^1 = (\mathbb{C}^2 \backslash \{0\}) \ / \ \mathbb{C}^*$$

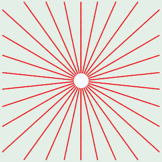
Quotients

Quotients of algebraic varieties by algebraic groups play an important role in constructing moduli spaces.

Example

The moduli space of 1-dimensional vector subspaces of $\mathbb{C}^2$ is

$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$



$$\mathbb{P}^1 = (\mathbb{C}^2 \backslash \{0\}) \;/\; \mathbb{C}^*$$

Construction depends on the choice of open subset

$$U = \mathbb{C}^2 \backslash \{0\} \subset \mathbb{C}^2$$

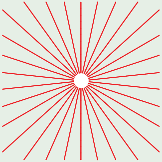
Quotients

Quotients of algebraic varieties by algebraic groups play an important role in constructing moduli spaces.

Example

The moduli space of 1-dimensional vector subspaces of $\mathbb{C}^2$ is

$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$



$$\mathbb{P}^1 = (\mathbb{C}^2 \backslash \{0\}) \ / \ \mathbb{C}^*$$

Construction depends on the choice of open subset

$$U = \mathbb{C}^2 \backslash \{0\} \subset \mathbb{C}^2$$

For torus actions on affine varieties, classify all possible quotients (choices of open sets) in terms of a polyhedral fan, the **GIT-fan**.

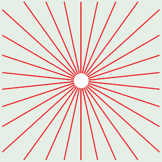
Quotients

Quotients of algebraic varieties by algebraic groups play an important role in constructing moduli spaces.

Example

The moduli space of 1-dimensional vector subspaces of $\mathbb{C}^2$ is

$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$



$$\mathbb{P}^1 = (\mathbb{C}^2 \backslash \{0\}) \ / \ \mathbb{C}^*$$

Construction depends on the choice of open subset

$$U = \mathbb{C}^2 \backslash \{0\} \subset \mathbb{C}^2$$

For torus actions on affine varieties, classify all possible quotients (choices of open sets) in terms of a polyhedral fan, the **GIT-fan**.
This setting plays an important role in constructing Mori Dream Spaces.
Some moduli spaces are of this type.

Outline

- Good quotients
- GIT-fan

Outline

- Good quotients
- GIT-fan

- Computing GIT-fans
- Fast monomial containment test

Outline

- Good quotients
- GIT-fan

- Computing GIT-fans
- Fast monomial containment test

- Symmetry groups of torus actions
- Representation of GIT-cones
- Symmetric GIT-fan algorithm

Outline

- Good quotients
- GIT-fan

- Computing GIT-fans
- Fast monomial containment test

- Symmetry groups of torus actions
- Representation of GIT-cones
- Symmetric GIT-fan algorithm

- Applications

Good Quotients

The goal of Geometric Invariant Theory (GIT) is to assign to a given algebraic variety X with action of an algebraic group G a reasonable quotient space $X /\!/ G$.

Good Quotients

The goal of Geometric Invariant Theory (GIT) is to assign to a given algebraic variety X with action of an algebraic group G a reasonable quotient space $X /\!/ G$.

Two main problems:

Good Quotients

The goal of Geometric Invariant Theory (GIT) is to assign to a given algebraic variety X with action of an algebraic group G a reasonable quotient space $X /\!/ G$.

Two main problems:

1. The orbit space X / G is not a good candidate for $X /\!/ G$ as it does not necessarily carry the structure of an algebraic variety (often not separated).

Good Quotients

The goal of Geometric Invariant Theory (GIT) is to assign to a given algebraic variety X with action of an algebraic group G a reasonable quotient space $X /\!/ G$.

Two main problems:

1. The orbit space X/G is not a good candidate for $X /\!/ G$ as it does not necessarily carry the structure of an algebraic variety (often not separated).

   Hence, for X affine define

   $$X /\!/ G = \operatorname{Spec} K[X]^G$$

   as the spectrum of the (finitely generated) invariant ring of the functions of X. For general X, glue the quotients of an affine covering.

Good Quotients

The goal of Geometric Invariant Theory (GIT) is to assign to a given algebraic variety X with action of an algebraic group G a reasonable quotient space $X /\!/ G$.

Two main problems:

1. The orbit space X / G is not a good candidate for $X /\!/ G$ as it does not necessarily carry the structure of an algebraic variety (often not separated).

   Hence, for X affine define

   $$X /\!/ G = \operatorname{Spec} K[X]^G$$

   as the spectrum of the (finitely generated) invariant ring of the functions of X. For general X, glue the quotients of an affine covering.

2. The quotient $X /\!/ G$ may not carry much information.

Good Quotients

The goal of Geometric Invariant Theory (GIT) is to assign to a given algebraic variety X with action of an algebraic group G a reasonable quotient space $X /\!/ G$.

Two main problems:

1. The orbit space X / G is not a good candidate for $X /\!/ G$ as it does not necessarily carry the structure of an algebraic variety (often not separated).

   Hence, for X affine define

   $$X /\!/ G = \operatorname{Spec} K[X]^G$$

   as the spectrum of the (finitely generated) invariant ring of the functions of X. For general X, glue the quotients of an affine covering.

2. The quotient $X /\!/ G$ may not carry much information.

   Hence pass to open subset $U \subset X$.

Good Quotients

Example

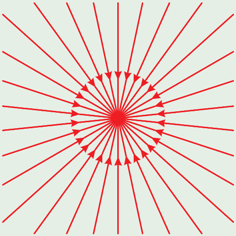
$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$

Good Quotients

Example

$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$

$$U = \mathbb{C}^2$$



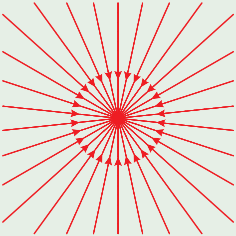
$$U // \mathbb{C}^* = \{pt\}$$

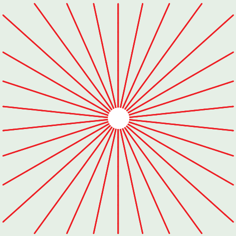
Good Quotients

Example

$$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \qquad t \cdot (x, y) = (tx, ty)$$

$$U = \mathbb{C}^2 \qquad\qquad U = \mathbb{C}^2 \backslash \{0\}$$





$$U /\!/ \mathbb{C}^* = \{pt\} \qquad\qquad U /\!/ \mathbb{C}^* = \mathbb{P}^1$$

GIT-Fan

In general, there are many choices for these open subsets $U \subset X$ leading to different quotients. To describe this behaviour, Dolgachev and Hu introduced the **GIT-fan**, a polyhedral fan describing the variation of GIT-quotients.

GIT-Fan

In general, there are many choices for these open subsets $U \subset X$ leading to different quotients. To describe this behaviour, Dolgachev and Hu introduced the **GIT-fan**, a polyhedral fan describing the variation of GIT-quotients. We focus on the action of an algebraic torus $G = (\mathbb{C}^*)^k$ on an affine variety $X \subset \mathbb{C}^r$.

GIT-Fan

In general, there are many choices for these open subsets $U \subset X$ leading to different quotients. To describe this behaviour, Dolgachev and Hu introduced the **GIT-fan**, a polyhedral fan describing the variation of GIT-quotients. We focus on the action of an algebraic torus $G = (\mathbb{C}^*)^k$ on an affine variety $X \subset \mathbb{C}^r$.

Setup:

- ideal $\mathfrak{a} \subset \mathbb{C}[T_1, \dots, T_r]$ defining X,
- matrix $Q = (q_1, \dots, q_r) \in \mathbb{Z}^{k \times r}$ such that $\mathfrak{a}$ is homogeneous w.r.t. grading $\deg(T_i) = q_i \in \mathbb{Z}^k$.

GIT-Fan

In general, there are many choices for these open subsets $U \subset X$ leading to different quotients. To describe this behaviour, Dolgachev and Hu introduced the **GIT-fan**, a polyhedral fan describing the variation of GIT-quotients. We focus on the action of an algebraic torus $G = (\mathbb{C}^*)^k$ on an affine variety $X \subset \mathbb{C}^r$.

Setup:

- ideal $\mathfrak{a} \subset \mathbb{C}[T_1, \dots, T_r]$ defining X,
- matrix $Q = (q_1, \dots, q_r) \in \mathbb{Z}^{k \times r}$ such that $\mathfrak{a}$ is homogeneous w.r.t. grading $\deg(T_i) = q_i \in \mathbb{Z}^k$.

Remark

Q encodes an action $(\mathbb{C}^*)^k \times X \to X$.

GIT-Fan

In general, there are many choices for these open subsets $U \subset X$ leading to different quotients. To describe this behaviour, Dolgachev and Hu introduced the **GIT-fan**, a polyhedral fan describing the variation of GIT-quotients. We focus on the action of an algebraic torus $G = (\mathbb{C}^*)^k$ on an affine variety $X \subset \mathbb{C}^r$.

Setup:

- ideal $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ defining X,
- matrix $Q = (q_1, \ldots, q_r) \in \mathbb{Z}^{k \times r}$ such that $\mathfrak{a}$ is homogeneous w.r.t. grading $\deg(T_i) = q_i \in \mathbb{Z}^k$.

Remark

Q encodes an action $(\mathbb{C}^*)^k \times X \to X$.

Example

$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \quad t \cdot (x, y) = (tx, ty)$

GIT-Fan

In general, there are many choices for these open subsets $U \subset X$ leading to different quotients. To describe this behaviour, Dolgachev and Hu introduced the **GIT-fan**, a polyhedral fan describing the variation of GIT-quotients. We focus on the action of an algebraic torus $G = (\mathbb{C}^*)^k$ on an affine variety $X \subset \mathbb{C}^r$.

Setup:

- ideal $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ defining X,
- matrix $Q = (q_1, \ldots, q_r) \in \mathbb{Z}^{k \times r}$ such that $\mathfrak{a}$ is homogeneous w.r.t. grading $\deg(T_i) = q_i \in \mathbb{Z}^k$.

Remark

Q encodes an action $(\mathbb{C}^*)^k \times X \to X$.

Example

$\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \quad t \cdot (x, y) = (tx, ty)$ is encoded by $Q = (1, 1) \in \mathbb{Z}^{1 \times 2}$.

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \operatorname{cone}\{w \in \operatorname{im} Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \text{cone}\{w \in \text{im}\, Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

For $w \in \text{im}\, Q$ the **set of semistable points** is

$$X^{ss}(w) = \{x \in X \mid \exists n > 0,\ f \in A_{nw} \text{ with } f(x) \neq 0\} \subset X$$

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \text{cone}\{w \in \text{im } Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

For $w \in \text{im } Q$ the **set of semistable points** is

$$X^{ss}(w) = \{x \in X \mid \exists n > 0,\ f \in A_{nw} \text{ with } f(x) \neq 0\} \subset X$$

and the **GIT-cone** is

$$\lambda(w) = \bigcap_{w \in \omega(x)} \omega(x)$$

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \text{cone}\{w \in \text{im } Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

For $w \in \text{im } Q$ the **set of semistable points** is

$$X^{ss}(w) = \{x \in X \mid \exists n > 0,\ f \in A_{nw} \text{ with } f(x) \neq 0\} \subset X$$

and the **GIT-cone** is

$$\lambda(w) = \bigcap_{w \in \omega(x)} \omega(x)$$

Remark

$\Omega = \{\omega(x) \mid x \in X\}$ *is finite.*

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \text{cone}\{w \in \text{im } Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

For $w \in \text{im } Q$ the **set of semistable points** is

$$X^{ss}(w) = \{x \in X \mid \exists n > 0,\ f \in A_{nw} \text{ with } f(x) \neq 0\} \subset X$$

and the **GIT-cone** is

$$\lambda(w) = \bigcap_{w \in \omega(x)} \omega(x)$$

Remark

$\Omega = \{\omega(x) \mid x \in X\}$ *is finite.*
$X^{ss}(w) \subset X$ *is torus invariant open, non-empty if* $w \in \text{cone}(q_1, \dots, q_r)$

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \text{cone}\{w \in \text{im } Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

For $w \in \text{im } Q$ the **set of semistable points** is

$$X^{ss}(w) = \{x \in X \mid \exists n > 0,\ f \in A_{nw} \text{ with } f(x) \neq 0\} \subset X$$

and the **GIT-cone** is

$$\lambda(w) = \bigcap_{w \in \omega(x)} \omega(x)$$

Remark

$\Omega = \{\omega(x) \mid x \in X\}$ *is finite.*
$X^{ss}(w) \subset X$ *is torus invariant open, non-empty if* $w \in \text{cone}(q_1, \dots, q_r)$
$\lambda(w)$ *is a polyhedral cone.*

GIT-Fans

Write $A = \mathbb{C}[T_1, \dots, T_r]/\mathfrak{a}$.

Definition

For $x \in X$ the **orbit cone** is

$$\omega(x) = \text{cone}\{w \in \text{im } Q \mid \exists f \in A_w \text{ with } f(x) \neq 0\}$$

For $w \in \text{im } Q$ the **set of semistable points** is

$$X^{ss}(w) = \{x \in X \mid \exists n > 0,\ f \in A_{nw} \text{ with } f(x) \neq 0\} \subset X$$

and the **GIT-cone** is

$$\lambda(w) = \bigcap_{w \in \omega(x)} \omega(x)$$

Remark

$\Omega = \{\omega(x) \mid x \in X\}$ *is finite.*
$X^{ss}(w) \subset X$ *is torus invariant open, non-empty if* $w \in \text{cone}(q_1, \dots, q_r)$
$\lambda(w)$ *is a polyhedral cone.*
$w \sim w' \Leftrightarrow X^{ss}(w) = X^{ss}(w')$ *is equivalence relation.*

GIT-Fans

Theorem

For every $w \in \operatorname{im} Q$ *there is a good quotient* $X^{ss}(w) /\!\!/ G$.

GIT-Fans

Theorem

For every $w \in \operatorname{im} Q$ *there is a good quotient* $X^{ss}(w) /\!/ G$.

Theorem

$\lambda(w) \subset \lambda(w') \Leftrightarrow X^{ss}(w) \supset X^{ss}(w')$
$\lambda(w) = \lambda(w') \Leftrightarrow X^{ss}(w) = X^{ss}(w')$

GIT-Fans

Theorem

For every $w \in \operatorname{im} Q$ *there is a good quotient* $X^{ss}(w) /\!/ G$.

Theorem

$\lambda(w) \subset \lambda(w') \Leftrightarrow X^{ss}(w) \supset X^{ss}(w')$
$\lambda(w) = \lambda(w') \Leftrightarrow X^{ss}(w) = X^{ss}(w')$

Theorem

$\Lambda(\mathfrak{a}, Q) = \{\lambda(w) \mid w\}$ *is rational polyhedral fan, the* ***GIT-fan***.

GIT-Fans

Theorem

For every $w \in \operatorname{im} Q$ *there is a good quotient* $X^{ss}(w)/\!/G$.

Theorem

$$\lambda(w) \subset \lambda(w') \Leftrightarrow X^{ss}(w) \supset X^{ss}(w')$$
$$\lambda(w) = \lambda(w') \Leftrightarrow X^{ss}(w) = X^{ss}(w')$$

Theorem

$\Lambda(\mathfrak{a}, Q) = \{\lambda(w) \mid w\}$ *is rational polyhedral fan, the* **GIT-fan**.

Example

For $\mathbb{C}^* \times \mathbb{C}^2 \to \mathbb{C}^2, \quad t \cdot (x, y) = (tx, ty)$

$$X^{ss}(0) = \mathbb{C}^2$$
$$X^{ss}(1) = \mathbb{C}^2 \backslash \{0\}$$

$\Lambda(\langle 0 \rangle, (1, 1)) =$ 

Computing GIT-Fans

Computing GIT-Fans

In case of a torus acting on an affine variety, Berchthold/Hausen and Keicher have developed a method for computing the GIT-fan.

Computing GIT-Fans

In case of a torus acting on an affine variety, Berchthold/Hausen and Keicher have developed a method for computing the GIT-fan.

Decomposition into torus orbits

$$\mathbb{C}^r = \bigcup_{\gamma} O(\gamma)$$

$$O(\gamma) = (\mathbb{C}^*)^r \cdot \sum_{e_i \in \gamma} e_i = \{(z_1, \ldots, z_r) \in \mathbb{C}^r \mid z_i \neq 0 \Leftrightarrow e_i \in \gamma\}$$

for faces $\gamma \prec \mathbb{Q}^r_{\geq 0}$.

Computing GIT-Fans

In case of a torus acting on an affine variety, Berchthold/Hausen and Keicher have developed a method for computing the GIT-fan.

Decomposition into torus orbits

$$\mathbb{C}^r = \bigcup_{\gamma} O(\gamma)$$

$$O(\gamma) = (\mathbb{C}^*)^r \cdot \sum_{e_i \in \gamma} e_i = \{(z_1, \ldots, z_r) \in \mathbb{C}^r \mid z_i \neq 0 \Leftrightarrow e_i \in \gamma\}$$

for faces $\gamma \prec \mathbb{Q}^r_{\geq 0}$.

Proposition

*Face $\gamma \prec \mathbb{Q}^r_{\geq 0}$ is called an $\mathfrak{a}$-**face** if the following equivalent conditions are satisfied:*

Computing GIT-Fans

In case of a torus acting on an affine variety, Berchthold/Hausen and Keicher have developed a method for computing the GIT-fan.

Decomposition into torus orbits

$$\mathbb{C}^r = \bigcup_{\gamma} O(\gamma)$$

$$O(\gamma) = (\mathbb{C}^*)^r \cdot \sum_{e_i \in \gamma} e_i = \{(z_1, \dots, z_r) \in \mathbb{C}^r \mid z_i \neq 0 \Leftrightarrow e_i \in \gamma\}$$

for faces $\gamma \prec \mathbb{Q}^r_{\geq 0}$.

Proposition

Face $\gamma \prec \mathbb{Q}^r_{\geq 0}$ is called an $\mathfrak{a}$-**face** *if the following equivalent conditions are satisfied:*

1. $X \cap O(\gamma) \neq \emptyset$

Computing GIT-Fans

In case of a torus acting on an affine variety, Berchthold/Hausen and Keicher have developed a method for computing the GIT-fan.

Decomposition into torus orbits

$$\mathbb{C}^r = \bigcup_{\gamma} O(\gamma)$$

$$O(\gamma) = (\mathbb{C}^*)^r \cdot \sum_{e_i \in \gamma} e_i = \{(z_1, \ldots, z_r) \in \mathbb{C}^r \mid z_i \neq 0 \Leftrightarrow e_i \in \gamma\}$$

for faces $\gamma \prec \mathbb{Q}^r_{\geq 0}$.

Proposition

Face $\gamma \prec \mathbb{Q}^r_{\geq 0}$ is called an $\mathfrak{a}$-**face** *if the following equivalent conditions are satisfied:*

1. $X \cap O(\gamma) \neq \emptyset$
2. $\left(\mathfrak{a}|_{T_i=0 \text{ for } e_i \notin \gamma}\right) : \langle T_1 \cdots T_r \rangle^\infty \neq \langle 1 \rangle$

Computing GIT-Fans

In case of a torus acting on an affine variety, Berchthold/Hausen and Keicher have developed a method for computing the GIT-fan.

Decomposition into torus orbits

$$\mathbb{C}^r = \bigcup_{\gamma} O(\gamma)$$

$$O(\gamma) = (\mathbb{C}^*)^r \cdot \sum_{e_i \in \gamma} e_i = \{(z_1, \ldots, z_r) \in \mathbb{C}^r \mid z_i \neq 0 \Leftrightarrow e_i \in \gamma\}$$

for faces $\gamma \prec \mathbb{Q}^r_{\geq 0}$.

Proposition

Face $\gamma \prec \mathbb{Q}^r_{\geq 0}$ is called an $\mathfrak{a}$**-face** *if the following equivalent conditions are satisfied:*

1. $X \cap O(\gamma) \neq \emptyset$
2. $\left(\mathfrak{a}|_{T_i=0 \text{ for } e_i \notin \gamma}\right) : \langle T_1 \cdots T_r \rangle^\infty \neq \langle 1 \rangle$
3. *There is* $x \in X$ *with* $x_i \neq 0 \Leftrightarrow e_i \in \gamma$

Computing GIT-Fans

1. Determine $\mathfrak{a}$-faces.

Computing GIT-Fans

1. Determine $\mathfrak{a}$-faces.
2. Compute set of orbit cones

$$\Omega = \{Q(\gamma) \mid \gamma \text{ an } \mathfrak{a}\text{-face}\}$$

where

$$Q(\gamma) = \text{cone}(q_i \mid e_i \in \gamma) \subset \Gamma = Q(\mathbb{Q}^r_{\geq 0}) = \text{cone}(q_1, \ldots, q_r) \subset \mathbb{Q}^k$$

is projection of γ with respect to Q.

1. Determine $\mathfrak{a}$-faces.
2. Compute set of orbit cones

$$\Omega = \{Q(\gamma) \mid \gamma \text{ an } \mathfrak{a}\text{-face}\}$$

   where

$$Q(\gamma) = \operatorname{cone}(q_i \mid e_i \in \gamma) \subset \Gamma = Q(\mathbb{Q}^r_{\geq 0}) = \operatorname{cone}(q_1, \ldots, q_r) \subset \mathbb{Q}^k$$

   is projection of γ with respect to Q.
3. Determine GIT-fan:

$$\Lambda(\mathfrak{a}, Q) = \{\lambda_\Omega(w) \mid w \in \Gamma\} \quad \text{where} \quad \lambda_\Omega(w) = \bigcap_{w \in \eta \in \Omega} \eta$$

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma \text{ is an } \mathfrak{a}\text{-face}\}$

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma \text{ is an } \mathfrak{a}\text{-face}\}$
2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma \text{ is an } \mathfrak{a}\text{-face}\}$

2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$

3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \dots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma \text{ is an } \mathfrak{a}\text{-face}\}$

2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$

3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.

4: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma$ *is an* $\mathfrak{a}$*-face*$\}$
2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$
3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
4: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$
5: $\mathcal{F} := \{(\eta, \lambda_\Omega(w_0)) \mid \eta \prec \lambda_\Omega(w_0)$ *interior facet*$\}$.

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma$ *is an* $\mathfrak{a}$*-face*$\}$

2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$

3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.

4: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$

5: $\mathcal{F} := \{(\eta, \lambda_\Omega(w_0)) \mid \eta \prec \lambda_\Omega(w_0)$ *interior facet*$\}$.

6: **while** *there is* $(\eta, \lambda) \in \mathcal{F}$ **do**

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \dots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma$ *is an* $\mathfrak{a}$*-face*$\}$

2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$

3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.

4: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$

5: $\mathcal{F} := \{(\eta, \lambda_\Omega(w_0)) \mid \eta \prec \lambda_\Omega(w_0)$ *interior facet*$\}$.

6: **while** *there is* $(\eta, \lambda) \in \mathcal{F}$ **do**

7: *Find* $w \in Q(\gamma)$ *such that* $w \notin \lambda$ *and* $\lambda_\Omega(w) \cap \lambda = \eta$.

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma$ *is an* $\mathfrak{a}$*-face*$\}$
2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$
3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
4: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$
5: $\mathcal{F} := \{(\eta, \lambda_\Omega(w_0)) \mid \eta \prec \lambda_\Omega(w_0)$ *interior facet*$\}$.
6: **while** *there is* $(\eta, \lambda) \in \mathcal{F}$ **do**
7:     *Find* $w \in Q(\gamma)$ *such that* $w \notin \lambda$ *and* $\lambda_\Omega(w) \cap \lambda = \eta$.
8:     $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}$

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \dots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q*.*

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$*.*

1: $\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma \text{ is an } \mathfrak{a}\text{-face}\}$

2: $\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$

3: *Choose a vector* $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.

4: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$

5: $\mathcal{F} := \{(\eta, \lambda_\Omega(w_0)) \mid \eta \prec \lambda_\Omega(w_0) \text{ interior facet}\}$.

6: **while** *there is* $(\eta, \lambda) \in \mathcal{F}$ **do**

7: *Find* $w \in Q(\gamma)$ *such that* $w \notin \lambda$ *and* $\lambda_\Omega(w) \cap \lambda = \eta$.

8: $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}$

9: $\mathcal{F} := \mathcal{F} \ominus \{(\tau, \lambda_\Omega(w)) \mid \tau \subset \lambda_\Omega(w) \text{ interior facet}\}$

GIT-Algorithm

Algorithm

Input: *Ideal* $\mathfrak{a} \subset \mathbb{C}[T_1, \ldots, T_r]$ *and matrix* $Q \in \mathbb{Z}^{k \times r}$ *of full rank such that* $\mathfrak{a}$ *is homogeneous w.r.t. multigrading by* Q.

Output: *The set of maximal cones of* $\Lambda(\mathfrak{a}, Q)$.

$\mathcal{A} := \{\gamma \prec \mathbb{Q}^r_{\geq 0} \mid \gamma \text{ is an } \mathfrak{a}\text{-face}\}$
$\Omega := \{Q(\gamma) \mid \gamma \in \mathcal{A}\}$
Choose a vector $w_0 \in Q(\mathbb{Q}^r_{\geq 0})$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
$\mathcal{C} := \{\lambda_\Omega(w_0)\}$
$\mathcal{F} := \{(\eta, \lambda_\Omega(w_0)) \mid \eta \prec \lambda_\Omega(w_0) \text{ interior facet}\}$.
while *there is* $(\eta, \lambda) \in \mathcal{F}$ **do**
    Find $w \in Q(\gamma)$ *such that* $w \notin \lambda$ *and* $\lambda_\Omega(w) \cap \lambda = \eta$.
    $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}$
    $\mathcal{F} := \mathcal{F} \ominus \{(\tau, \lambda_\Omega(w)) \mid \tau \subset \lambda_\Omega(w) \text{ interior facet}\}$
return $\mathcal{C}$

Singular

GIT-algorithm is implemented in SINGULAR library `gitfan.lib` using Gröbner bases for determining $\mathfrak{a}$-faces via saturation.

Singular

GIT-algorithm is implemented in SINGULAR library `gitfan.lib` using Gröbner bases for determining $\mathfrak{a}$-faces via saturation.

- Computer algebra system for polynomial computations, over 30 development teams worldwide, over 130 libraries for advanced topics.

Singular

GIT-algorithm is implemented in SINGULAR library `gitfan.lib` using Gröbner bases for determining $\mathfrak{a}$-faces via saturation.

- Computer algebra system for polynomial computations, over 30 development teams worldwide, over 130 libraries for advanced topics.

- Special emphasis on algebraic geometry, commutative and non-commutative algebra, singularity theory.

Singular

GIT-algorithm is implemented in SINGULAR library `gitfan.lib` using Gröbner bases for determining $\mathfrak{a}$-faces via saturation.

- Computer algebra system for polynomial computations, over 30 development teams worldwide, over 130 libraries for advanced topics.

- Special emphasis on algebraic geometry, commutative and non-commutative algebra, singularity theory.
- Packages for convex and tropical geometry.

Singular-Polymake Interface

Example (`polymake.so`)

Compute the normal fan F of the Newton polytope P of $f = x^3 + y^3 + 1$.

Singular-Polymake Interface

Example (polymake.so)

Compute the normal fan F of the Newton polytope P of $f = x^3 + y^3 + 1$.

```
> LIB "polymake.so";
  Welcome to polymake version 2.14
  Copyright (c) 1997-2015
  Ewgenij Gawrilow, Michael Joswig (TU Berlin)
  http://www.polymake.org
  // ** loaded polymake.so
```

Singular-Polymake Interface

Example (polymake.so)

Compute the normal fan F of the Newton polytope P of $f = x^3 + y^3 + 1$.

```
> LIB "polymake.so";
  Welcome to polymake version 2.14
  Copyright (c) 1997-2015
  Ewgenij Gawrilow, Michael Joswig (TU Berlin)
  http://www.polymake.org
  // ** loaded polymake.so
> ring R = 0,(x,y),dp; poly f = x3+y3+1;
```

Singular-Polymake Interface

Example (polymake.so)

Compute the normal fan F of the Newton polytope P of $f = x^3 + y^3 + 1$.

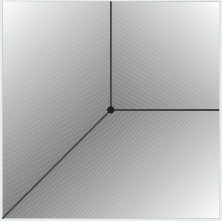
```
> LIB "polymake.so";
  Welcome to polymake version 2.14
  Copyright (c) 1997-2015
  Ewgenij Gawrilow, Michael Joswig (TU Berlin)
  http://www.polymake.org
  // ** loaded polymake.so
> ring R = 0,(x,y),dp; poly f = x3+y3+1;
> polytope P = newtonPolytope(f);
```

Singular-Polymake Interface

Example (polymake.so)

Compute the normal fan F of the Newton polytope P of $f = x^3 + y^3 + 1$.

```
> LIB "polymake.so";
  Welcome to polymake version 2.14
  Copyright (c) 1997-2015
  Ewgenij Gawrilow, Michael Joswig (TU Berlin)
  http://www.polymake.org
  // ** loaded polymake.so
> ring R = 0,(x,y),dp; poly f = x3+y3+1;
> polytope P = newtonPolytope(f);
> fan F = normalFan(P); F;
  RAYS:
  -1 -1 #0
   0  1 #1
   1  0 #2
  MAXIMAL_CONES:
  {0 1} #Dimension 2
  {0 2}
  {1 2}
```



Parallel computations in Singular

Computation of $\mathfrak{a}$-faces is trivially parallel.

Parallel computations in Singular

Computation of $\mathfrak{a}$-faces is trivially parallel.

Example (parallel.lib)

```
> LIB "parallel.lib"; LIB "random.lib";
> ring R = 0,x(1..4),dp;
> ideal I = randomid(maxideal(3),3,100);
```

Parallel computations in Singular

Computation of $\mathfrak{a}$-faces is trivially parallel.

Example (parallel.lib)

```
> LIB "parallel.lib"; LIB "random.lib";
> ring R = 0,x(1..4),dp;
> ideal I = randomid(maxideal(3),3,100);
> proc sizeStd(ideal I, string monord){
      def R = basering; list RL = ringlist(R);
      RL[3][1][1] = monord; def S = ring(RL); setring(S);
      return(size(std(imap(R,I))));}
```

Parallel computations in Singular

Computation of $\mathfrak{a}$-faces is trivially parallel.

Example (parallel.lib)

```
> LIB "parallel.lib"; LIB "random.lib";
> ring R = 0,x(1..4),dp;
> ideal I = randomid(maxideal(3),3,100);
> proc sizeStd(ideal I, string monord){
      def R = basering; list RL = ringlist(R);
      RL[3][1][1] = monord; def S = ring(RL); setring(S);
      return(size(std(imap(R,I))));}
> list commands = "sizeStd","sizeStd";
> list args = list(I,"lp"),list(I,"dp");
```

Parallel computations in Singular

Computation of $\mathfrak{a}$-faces is trivially parallel.

Example (parallel.lib)

```
> LIB "parallel.lib"; LIB "random.lib";
> ring R = 0,x(1..4),dp;
> ideal I = randomid(maxideal(3),3,100);
> proc sizeStd(ideal I, string monord){
      def R = basering; list RL = ringlist(R);
      RL[3][1][1] = monord; def S = ring(RL); setring(S);
      return(size(std(imap(R,I))));}
> list commands = "sizeStd","sizeStd";
> list args = list(I,"lp"),list(I,"dp");
> parallelWaitFirst(commands, args);
  [1] empty list
  [2] 11
```

Parallel computations in Singular

Computation of a-faces is trivially parallel.

Example (parallel.lib)

```
> LIB "parallel.lib"; LIB "random.lib";
> ring R = 0,x(1..4),dp;
> ideal I = randomid(maxideal(3),3,100);
> proc sizeStd(ideal I, string monord){
      def R = basering; list RL = ringlist(R);
      RL[3][1][1] = monord; def S = ring(RL); setring(S);
      return(size(std(imap(R,I))));}
> list commands = "sizeStd","sizeStd";
> list args = list(I,"lp"),list(I,"dp");
> parallelWaitFirst(commands, args);
  [1] empty list
  [2] 11
> parallelWaitAll(commands, args);
  [1] 55
  [2] 11
```

Fast Monomial Containment Test

Generalization of (Sturmfels, 1996), where degree reverse lex (dp) is used:

Fast Monomial Containment Test

Generalization of (Sturmfels, 1996), where degree reverse lex (dp) is used:

Proposition

Let $>$ be a monomial ordering on $R = K[Y_1, \ldots, Y_n]$ and $\mathcal{G}$ a Gröbner basis of I. Suppose that for all $f \in \mathcal{G}$

$$Y_n \mid f \quad \Longleftrightarrow \quad Y_n \mid \mathrm{LM}_>(f).$$

Fast Monomial Containment Test

Generalization of (Sturmfels, 1996), where degree reverse lex (dp) is used:

Proposition

Let $>$ be a monomial ordering on $R = K[Y_1, \dots, Y_n]$ and $\mathcal{G}$ a Gröbner basis of I. Suppose that for all $f \in \mathcal{G}$

$$Y_n \mid f \quad \Longleftrightarrow \quad Y_n \mid \mathrm{LM}_>(f).$$

Then

$$\left\{ \frac{f}{Y_n^i} \;\middle|\; f \in \mathcal{G} \text{ and } i \geq 0 \text{ maximal such that } Y_n^i \mid f \right\}$$

is a Gröbner basis for $I : Y_n^\infty$.

Fast Monomial Containment Test

Generalization of (Sturmfels, 1996), where degree reverse lex (dp) is used:

Proposition

Let $>$ be a monomial ordering on $R = K[Y_1, \ldots, Y_n]$ and $\mathcal{G}$ a Gröbner basis of I. Suppose that for all $f \in \mathcal{G}$

$$Y_n \mid f \quad \Longleftrightarrow \quad Y_n \mid \mathrm{LM}_>(f).$$

Then

$$\left\{ \frac{f}{Y_n^i} \;\middle|\; f \in \mathcal{G} \text{ and } i \geq 0 \text{ maximal such that } Y_n^i \mid f \right\}$$

is a Gröbner basis for $I : Y_n^\infty$.

Algorithm

To compute $I : (Y_1 \cdot \ldots \cdot Y_n)^\infty$,

Fast Monomial Containment Test

Generalization of (Sturmfels, 1996), where degree reverse lex (dp) is used:

Proposition

Let $>$ be a monomial ordering on $R = K[Y_1, \ldots, Y_n]$ and $\mathcal{G}$ a Gröbner basis of I. Suppose that for all $f \in \mathcal{G}$

$$Y_n \mid f \quad \Longleftrightarrow \quad Y_n \mid \mathrm{LM}_>(f).$$

Then

$$\left\{ \frac{f}{Y_n^i} \;\middle|\; f \in \mathcal{G} \text{ and } i \geq 0 \text{ maximal such that } Y_n^i \mid f \right\}$$

is a Gröbner basis for $I : Y_n^\infty$.

Algorithm

To compute $I : (Y_1 \cdot \ldots \cdot Y_n)^\infty$, replace any remainder $r \neq 0$ in Buchberger's algorithm by

$$\frac{r}{Y_1^{a_1} \cdot \ldots \cdot Y_n^{a_n}} \quad \text{where } a_j \text{ is maximal s.t. } Y_j^{a_j} \mid r.$$

Timings

Saturation in product of variables for ideal $\mathfrak{a}$ with 225 generators in 40 variables with variables not in J equal to 0:

Timings

Saturation in product of variables for ideal $\mathfrak{a}$ with 225 generators in 40 variables with variables not in J equal to 0:

$\{1,\ldots,40\}\setminus J$	$40-\lvert J\rvert$	$\mathfrak{a}$-face	divgbsat	gbsat	sat	rabinowitsch
$\{3,4,5,7,\ldots,15\}$	28	no	1	761	517	342
$\{9,11,12,13,15\}$	35	no	1	57200	$*$	$*$
$\{11,12,13,15\}$	36	no	1	44100	$*$	$*$
$\{9,11,14,15\}$	36	yes	64	121000	$*$	$*$
$\{9,11,15\}$	37	yes	1170	114000	$*$	$*$
$\{9,11,13\}$	37	no	1	31400	$*$	$*$

(in seconds, * did not finish in $>$ 2 days)

Symmetry Groups of Torus Actions

Definition

A **symmetry group** of the action of $\mathbb{C}^r$ on X is a subgroup $G \subset S_r$ of the symmetric group such that there are group actions

Symmetry Groups of Torus Actions

Definition

A **symmetry group** of the action of $\mathbb{C}^r$ on X is a subgroup $G \subset S_r$ of the symmetric group such that there are group actions

$$\begin{array}{lllllll} G & \times & \mathbb{K}[T_1,\ldots,T_r] & \to & \mathbb{K}[T_1,\ldots,T_r], & (\sigma, T_j) & \mapsto & \sigma(T_j) = c_{\sigma,j} \cdot T_{\sigma(j)} \\ G & \times & \mathbb{Q}^r & \to & \mathbb{Q}^r, & (\sigma, e_j) & \mapsto & \sigma(e_j) = e_{\sigma(j)} \\ G & \times & \mathbb{Q}^k & \to & \mathbb{Q}^k, & (\sigma, v) & \mapsto & A_\sigma \cdot v \end{array}$$

Symmetry Groups of Torus Actions

Definition

A **symmetry group** of the action of $\mathbb{C}^r$ on X is a subgroup $G \subset S_r$ of the symmetric group such that there are group actions

$$\begin{array}{lllll} G \times \mathbb{K}[T_1,\ldots,T_r] & \to & \mathbb{K}[T_1,\ldots,T_r], & (\sigma, T_j) & \mapsto \sigma(T_j) = c_{\sigma,j} \cdot T_{\sigma(j)} \\ G \times \mathbb{Q}^r & \to & \mathbb{Q}^r, & (\sigma, e_j) & \mapsto \sigma(e_j) = e_{\sigma(j)} \\ G \times \mathbb{Q}^k & \to & \mathbb{Q}^k, & (\sigma, v) & \mapsto A_\sigma \cdot v \end{array}$$

with $A_\sigma \in \mathrm{GL}(k,\mathbb{Q})$ and $c_\sigma \in \mathbb{T}^r$

Symmetry Groups of Torus Actions

Definition

A **symmetry group** of the action of $\mathbb{C}^r$ on X is a subgroup $G \subset S_r$ of the symmetric group such that there are group actions

$$\begin{array}{llllll} G & \times \mathbb{K}[T_1,\ldots,T_r] & \to & \mathbb{K}[T_1,\ldots,T_r], & (\sigma, T_j) & \mapsto & \sigma(T_j) = c_{\sigma,j} \cdot T_{\sigma(j)} \\ G & \times \mathbb{Q}^r & \to & \mathbb{Q}^r, & (\sigma, e_j) & \mapsto & \sigma(e_j) = e_{\sigma(j)} \\ G & \times \mathbb{Q}^k & \to & \mathbb{Q}^k, & (\sigma, v) & \mapsto & A_\sigma \cdot v \end{array}$$

with $A_\sigma \in \mathrm{GL}(k, \mathbb{Q})$ and $c_\sigma \in \mathbb{T}^r$ such that $G \cdot \mathfrak{a} = \mathfrak{a}$ and that for each $\sigma \in G$ the following diagram is commutative:

Symmetry Groups of Torus Actions

Definition

A **symmetry group** of the action of $\mathbb{C}^r$ on X is a subgroup $G \subset S_r$ of the symmetric group such that there are group actions

$$\begin{array}{lllll} G \times \mathbb{K}[T_1,\ldots,T_r] & \to & \mathbb{K}[T_1,\ldots,T_r], & (\sigma, T_j) & \mapsto & \sigma(T_j) = c_{\sigma,j} \cdot T_{\sigma(j)} \\ G \times \mathbb{Q}^r & \to & \mathbb{Q}^r, & (\sigma, e_j) & \mapsto & \sigma(e_j) = e_{\sigma(j)} \\ G \times \mathbb{Q}^k & \to & \mathbb{Q}^k, & (\sigma, v) & \mapsto & A_\sigma \cdot v \end{array}$$

with $A_\sigma \in \mathrm{GL}(k,\mathbb{Q})$ and $c_\sigma \in \mathbb{T}^r$ such that $G \cdot \mathfrak{a} = \mathfrak{a}$ and that for each $\sigma \in G$ the following diagram is commutative:

$$\begin{array}{ccc} e_j & \longmapsto & e_{\sigma(j)} \\ \mathbb{Q}^r & \longrightarrow & \mathbb{Q}^r \\ Q \downarrow & & \downarrow Q \\ \mathbb{Q}^k & \underset{A_\sigma}{\longrightarrow} & \mathbb{Q}^k \end{array}$$

Perfect hash function for cones with compatible group action

Representation of GIT-Cones

Perfect hash function for cones with compatible group action

$$h_\Omega \colon \Lambda(\mathfrak{a}, Q) \to \{0,1\}^\Omega, \qquad \lambda \mapsto \left[\begin{array}{l} \Omega \to \{0,1\} \\ \vartheta \mapsto \begin{cases} 1 & \lambda \subset \vartheta \\ 0 & \lambda \not\subset \vartheta \end{cases} \end{array} \right]$$

Representation of GIT-Cones

Perfect hash function for cones with compatible group action

$$h_\Omega \colon \Lambda(\mathfrak{a}, Q) \to \{0,1\}^\Omega, \qquad \lambda \mapsto \left[\begin{array}{l} \Omega \to \{0,1\} \\ \vartheta \mapsto \begin{cases} 1 & \lambda \subset \vartheta \\ 0 & \lambda \not\subset \vartheta \end{cases} \end{array} \right]$$

$$G \times \{0,1\}^\Omega \to \{0,1\}^\Omega, \qquad (g, b) \mapsto \left[\begin{array}{l} \Omega \to \{0,1\} \\ \vartheta \mapsto b(g^{-1} \cdot \vartheta) \end{array} \right]$$

Representation of GIT-Cones

Perfect hash function for cones with compatible group action

$$h_\Omega \colon \Lambda(\mathfrak{a}, Q) \to \{0,1\}^\Omega, \qquad \lambda \mapsto \left[\begin{array}{l} \Omega \to \{0,1\} \\ \vartheta \mapsto \begin{cases} 1 & \lambda \subset \vartheta \\ 0 & \lambda \not\subset \vartheta \end{cases} \end{array} \right]$$

$$G \times \{0,1\}^\Omega \to \{0,1\}^\Omega, \qquad (g, b) \mapsto \left[\begin{array}{l} \Omega \to \{0,1\} \\ \vartheta \mapsto b(g^{-1} \cdot \vartheta) \end{array} \right]$$

such that

$$g \cdot h_\Omega(\lambda) = h_\Omega(g \cdot \lambda).$$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of G-orbits of* $\text{faces}(\mathbb{Q}^r_{\geq 0})$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of G-orbits of* $\text{faces}(\mathbb{Q}^r_{\geq 0})$
2: $\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma$ *is* $\mathfrak{a}$*-face*$\}$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
2: $\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma \text{ is } \mathfrak{a}\text{-face}\}$
3: $\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
2: $\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma \textit{ is } \mathfrak{a}\textit{-face}\}$
3: $\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
4: $\Omega :=$ *set of minimal elements of* $\Omega(k)$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of* *G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
2: $\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma$ *is* $\mathfrak{a}$*-face*$\}$
3: $\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
4: $\Omega :=$ *set of minimal elements of* $\Omega(k)$
5: *Choose* $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
2: $\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma$ *is* $\mathfrak{a}$*-face*$\}$
3: $\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
4: $\Omega :=$ *set of minimal elements of* $\Omega(k)$
5: *Choose* $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
6: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

1: $\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
2: $\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma \text{ is } \mathfrak{a}\text{-face}\}$
3: $\Omega := \bigcup_{\gamma\in\mathcal{A}} G \cdot Q(\gamma)$
4: $\Omega :=$ *set of minimal elements of* $\Omega(k)$
5: *Choose* $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
6: $\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$
7: $\mathcal{F} := \{(\eta, v) \mid \eta \prec \lambda_\Omega(w_0) \text{ interior facet with inner normal } v\}$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

$\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
$\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma \textit{ is } \mathfrak{a}\textit{-face}\}$
$\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
$\Omega :=$ *set of minimal elements of* $\Omega(k)$
Choose $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$*.*
$\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$
$\mathcal{F} := \{(\eta, v) \mid \eta \prec \lambda_\Omega(w_0) \textit{ interior facet with inner normal } v\}$
while *there is* $(\eta, v) \in \mathcal{F}$ **do**
    Find $w \in Q(\Gamma)$ *such that* $\eta \prec \lambda_\Omega(w)$ *is a facet and* $-v \in \lambda_\Omega(w)^\vee$*.*

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

$\mathcal{S} :=$ *system of representatives of* *G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
$\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma \text{ is } \mathfrak{a}\text{-face}\}$
$\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
$\Omega :=$ *set of minimal elements of* $\Omega(k)$
Choose $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
$\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$
$\mathcal{F} := \{(\eta, v) \mid \eta \prec \lambda_\Omega(w_0) \text{ interior facet with inner normal } v\}$
while *there is* $(\eta, v) \in \mathcal{F}$ **do**
    Find $w \in Q(\Gamma)$ *such that* $\eta \prec \lambda_\Omega(w)$ *is a facet and* $-v \in \lambda_\Omega(w)^\vee$.
    if $G \cdot h_\Omega(\lambda_\Omega(w)) \cap \mathcal{H} = \emptyset$ **then**
        $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}, \mathcal{H} := \mathcal{H} \cup \{h_\Omega(\lambda_\Omega(w))\}$

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

$\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
$\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma$ *is* $\mathfrak{a}$*-face*$\}$
$\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
$\Omega :=$ *set of minimal elements of* $\Omega(k)$
Choose $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
$\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$
$\mathcal{F} := \{(\eta, v) \mid \eta \prec \lambda_\Omega(w_0)$ *interior facet with inner normal* $v\}$
while *there is* $(\eta, v) \in \mathcal{F}$ **do**
    Find $w \in Q(\Gamma)$ *such that* $\eta \prec \lambda_\Omega(w)$ *is a facet and* $-v \in \lambda_\Omega(w)^\vee$.
    if $G \cdot h_\Omega(\lambda_\Omega(w)) \cap \mathcal{H} = \emptyset$ **then**
        $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}, \mathcal{H} := \mathcal{H} \cup \{h_\Omega(\lambda_\Omega(w))\}$
        $\mathcal{F} := \mathcal{F} \ominus \{(\tilde{\eta}, \tilde{v}) \mid \tilde{\eta} \prec \lambda_\Omega(w)$ *interior facet w. inner normal* $\tilde{v}\}$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

$\mathcal{S} :=$ *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
$\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma \text{ is } \mathfrak{a}\text{-face}\}$
$\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
$\Omega :=$ *set of minimal elements of* $\Omega(k)$
Choose $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
$\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$
$\mathcal{F} := \{(\eta, v) \mid \eta \prec \lambda_\Omega(w_0) \text{ interior facet with inner normal } v\}$
while *there is* $(\eta, v) \in \mathcal{F}$ **do**
    Find $w \in Q(\Gamma)$ *such that* $\eta \prec \lambda_\Omega(w)$ *is a facet and* $-v \in \lambda_\Omega(w)^\vee$.
    if $G \cdot h_\Omega(\lambda_\Omega(w)) \cap \mathcal{H} = \emptyset$ **then**
        $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}, \mathcal{H} := \mathcal{H} \cup \{h_\Omega(\lambda_\Omega(w))\}$
        $\mathcal{F} := \mathcal{F} \ominus \{(\tilde{\eta}, \tilde{v}) \mid \tilde{\eta} \prec \lambda_\Omega(w) \text{ interior facet w. inner normal } \tilde{v}\}$
    else
        $\mathcal{F} := \mathcal{F} \setminus \{(\eta, v)\}$

Symmetric GIT-Algorithm

Algorithm (System of representatives of the G-orbits on $\Lambda(\mathfrak{a}, Q)(k)$)

$\mathcal{S}$:= *system of representatives of G-orbits of* $\mathrm{faces}(\mathbb{Q}^r_{\geq 0})$
$\mathcal{A} := \{\gamma \in \mathcal{S} \mid \gamma$ *is* $\mathfrak{a}$*-face*$\}$
$\Omega := \bigcup_{\gamma \in \mathcal{A}} G \cdot Q(\gamma)$
Ω := *set of minimal elements of* $\Omega(k)$
Choose $w_0 \in Q(\Gamma)$ *such that* $\dim(\lambda_\Omega(w_0)) = k$.
$\mathcal{C} := \{\lambda_\Omega(w_0)\}$, $\mathcal{H} := \{h_\Omega(\lambda_\Omega(w_0))\}$
$\mathcal{F} := \{(\eta, v) \mid \eta \prec \lambda_\Omega(w_0)$ *interior facet with inner normal* $v\}$
while *there is* $(\eta, v) \in \mathcal{F}$ **do**
    Find $w \in Q(\Gamma)$ *such that* $\eta \prec \lambda_\Omega(w)$ *is a facet and* $-v \in \lambda_\Omega(w)^\vee$.
    if $G \cdot h_\Omega(\lambda_\Omega(w)) \cap \mathcal{H} = \emptyset$ **then**
        $\mathcal{C} := \mathcal{C} \cup \{\lambda_\Omega(w)\}, \mathcal{H} := \mathcal{H} \cup \{h_\Omega(\lambda_\Omega(w))\}$
        $\mathcal{F} := \mathcal{F} \ominus \{(\tilde{\eta}, \tilde{v}) \mid \tilde{\eta} \prec \lambda_\Omega(w)$ *interior facet w. inner normal* $\tilde{v}\}$
    else
        $\mathcal{F} := \mathcal{F} \setminus \{(\eta, v)\}$
return $\mathcal{C}$

An Example with D_4-Symmetry

Example

$$\mathfrak{a} = \langle T_1 T_3 - T_2 T_4 \rangle \subset \mathbb{K}[T_1, \ldots, T_4] \qquad \deg(T_j) = q_j$$

$$Q = (q_1, \ldots, q_4) = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix}$$

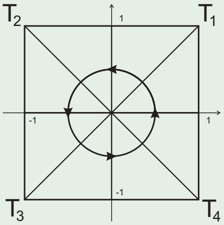
An Example with D_4-Symmetry

Example

$$\mathfrak{a} = \langle T_1 T_3 - T_2 T_4 \rangle \subset \mathbb{K}[T_1, \dots, T_4] \qquad \deg(T_j) = q_j$$

$$Q = (q_1, \dots, q_4) = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix}$$

$$G = D_4 = \langle (1,2)(3,4), (1,2,3,4) \rangle \subset S_4$$



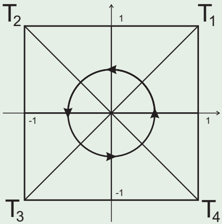
An Example with D_4-Symmetry

Example

$$\mathfrak{a} = \langle T_1 T_3 - T_2 T_4 \rangle \subset \mathbb{K}[T_1, \ldots, T_4] \qquad \deg(T_j) = q_j$$

$$Q = (q_1, \ldots, q_4) = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{pmatrix}$$

$$G = D_4 = \langle (1,2)(3,4), (1,2,3,4) \rangle \subset S_4$$



$$A_{(1,2)(3,4)} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \qquad A_{(1,2,3,4)} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Example with D_4-symmetry

Example

γ	$\lvert G \cdot \gamma \rvert$	$\mathfrak{a}\big\rvert_{T_i=0 \text{ for } e_i \notin \gamma}$	$\mathfrak{a}$-face
$\gamma_0 = \mathrm{cone}(0)$	1	0	`true`
$\gamma_1 = \mathrm{cone}(e_1)$	4	0	`true`
$\gamma_2 = \mathrm{cone}(e_1, e_2)$	4	0	`true`
$\gamma_2' = \mathrm{cone}(e_1, e_3)$	2	$\langle T_1 T_3 \rangle$	`false`
$\gamma_3 = \mathrm{cone}(e_1, e_2, e_3)$	4	$\langle T_1 T_3 \rangle$	`false`
$\gamma_4 = \mathrm{cone}(e_1, e_2, e_3, e_4)$	1	$\langle T_1 T_3 - T_2 T_4 \rangle$	`true`

Example with D_4-symmetry

Example

γ	$\lvert G\cdot\gamma\rvert$	$\mathfrak{a}\big\rvert_{T_i=0 \text{ for } e_i\notin\gamma}$	$\mathfrak{a}$-face
$\gamma_0 = \mathrm{cone}(0)$	1	0	`true`
$\gamma_1 = \mathrm{cone}(e_1)$	4	0	`true`
$\gamma_2 = \mathrm{cone}(e_1, e_2)$	4	0	`true`
$\gamma_2' = \mathrm{cone}(e_1, e_3)$	2	$\langle T_1 T_3\rangle$	`false`
$\gamma_3 = \mathrm{cone}(e_1, e_2, e_3)$	4	$\langle T_1 T_3\rangle$	`false`
$\gamma_4 = \mathrm{cone}(e_1, e_2, e_3, e_4)$	1	$\langle T_1 T_3 - T_2 T_4\rangle$	`true`

$$Q(\gamma_0) = \mathrm{cone}(0), \quad Q(\gamma_1) = \mathrm{cone}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad Q(\gamma_2) = \mathrm{cone}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right), \quad Q(\gamma_4) = \mathbb{Q}^2$$

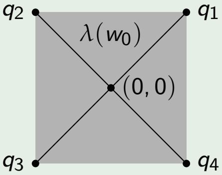
Example with D_4-symmetry

Example

γ	$\lvert G\cdot\gamma\rvert$	$\mathfrak{a}\vert_{T_i=0 \text{ for } e_i\notin\gamma}$	$\mathfrak{a}$-face
$\gamma_0 = \text{cone}(0)$	1	0	`true`
$\gamma_1 = \text{cone}(e_1)$	4	0	`true`
$\gamma_2 = \text{cone}(e_1, e_2)$	4	0	`true`
$\gamma_2' = \text{cone}(e_1, e_3)$	2	$\langle T_1 T_3\rangle$	`false`
$\gamma_3 = \text{cone}(e_1, e_2, e_3)$	4	$\langle T_1 T_3\rangle$	`false`
$\gamma_4 = \text{cone}(e_1, e_2, e_3, e_4)$	1	$\langle T_1 T_3 - T_2 T_4\rangle$	`true`

$$Q(\gamma_0) = \text{cone}(0), \quad Q(\gamma_1) = \text{cone}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad Q(\gamma_2) = \text{cone}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right), \quad Q(\gamma_4) = \mathbb{Q}^2$$

$$w_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



Mori dream spaces have a finitely generated Cox ring. They are toric iff the Cox ring is a polynomial ring.

Mori dream spaces have a finitely generated Cox ring. They are toric iff the Cox ring is a polynomial ring. Similar to toric varieties, they admit a construction as a GIT-quotient (Hu, Keel, 2000).

Mori dream spaces have a finitely generated Cox ring. They are toric iff the Cox ring is a polynomial ring. Similar to toric varieties, they admit a construction as a GIT-quotient (Hu, Keel, 2000). The GIT-fan yields the Mori chamber decomposition, which describes all birational modifications.

Mori dream spaces have a finitely generated Cox ring. They are toric iff the Cox ring is a polynomial ring. Similar to toric varieties, they admit a construction as a GIT-quotient (Hu, Keel, 2000). The GIT-fan yields the Mori chamber decomposition, which describes all birational modifications.

For the Deligne-Mumford compactification **moduli space of stable curves** of genus 0 with n marked points $\overline{M}_{0,n}$ (only double points, on each component ≥ 3 marked or double points) we have:

Moduli Spaces of Stable Maps and Mori Dream Spaces

Mori dream spaces have a finitely generated Cox ring. They are toric iff the Cox ring is a polynomial ring. Similar to toric varieties, they admit a construction as a GIT-quotient (Hu, Keel, 2000). The GIT-fan yields the Mori chamber decomposition, which describes all birational modifications.

For the Deligne-Mumford compactification **moduli space of stable curves** of genus 0 with n marked points $\overline{M}_{0,n}$ (only double points, on each component ≥ 3 marked or double points) we have:

1. $\overline{M}_{0,n}$ for $n \leq 6$ is a Mori dream space (Castravet, 2009 for $n = 6$)

Moduli Spaces of Stable Maps and Mori Dream Spaces

Mori dream spaces have a finitely generated Cox ring. They are toric iff the Cox ring is a polynomial ring. Similar to toric varieties, they admit a construction as a GIT-quotient (Hu, Keel, 2000). The GIT-fan yields the Mori chamber decomposition, which describes all birational modifications.

For the Deligne-Mumford compactification **moduli space of stable curves** of genus 0 with n marked points $\overline{M}_{0,n}$ (only double points, on each component ≥ 3 marked or double points) we have:

1. $\overline{M}_{0,n}$ for $n \leq 6$ is a Mori dream space (Castravet, 2009 for $n = 6$)
2. $\overline{M}_{0,n}$ for $n > 133$ is not a Mori dream space (Castravet, Tevelev, 2013)

Example: GIT-fan for $\mathbb{G}(2,5)$

Example

Cox ring of $\overline{M}_{0,5}$ is isomorphic to $\mathbb{Z}^5$-graded coordinate ring $R = \mathbb{K}[T_1, \ldots, T_{10}]/\mathfrak{a}$ of affine cone over $\mathbb{G}(2,5)$.

Example: GIT-fan for $\mathbb{G}(2,5)$

Example

Cox ring of $\overline{M}_{0,5}$ is isomorphic to $\mathbb{Z}^5$-graded coordinate ring $R = \mathbb{K}[T_1,\ldots,T_{10}]/\mathfrak{a}$ of affine cone over $\mathbb{G}(2,5)$. Symmetry group action of

$$S_5 \cong G = \langle (2,3)(5,6)(9,10),\ (1,5,9,10,3)(2,7,8,4,6) \rangle \subset S_{10}$$

Example: GIT-fan for $\mathbb{G}(2,5)$

Example

Cox ring of $\overline{M}_{0,5}$ is isomorphic to $\mathbb{Z}^5$-graded coordinate ring $R = \mathbb{K}[T_1, \ldots, T_{10}]/\mathfrak{a}$ of affine cone over $\mathbb{G}(2,5)$. Symmetry group action of

$$S_5 \cong G = \langle (2,3)(5,6)(9,10),\ (1,5,9,10,3)(2,7,8,4,6) \rangle \subset S_{10}$$

	number	number of orbits
monomial containment tests	$2^{10} = 1024$	34
$\mathfrak{a}$-faces	172	14

Example: GIT-fan for $\mathbb{G}(2,5)$

Example

Cox ring of $\overline{M}_{0,5}$ is isomorphic to $\mathbb{Z}^5$-graded coordinate ring $R = \mathbb{K}[T_1, \ldots, T_{10}]/\mathfrak{a}$ of affine cone over $\mathbb{G}(2,5)$. Symmetry group action of

$$S_5 \cong G = \langle (2,3)(5,6)(9,10),\ (1,5,9,10,3)(2,7,8,4,6) \rangle \subset S_{10}$$

	number	number of orbits
monomial containment tests	$2^{10} = 1024$	34
$\mathfrak{a}$-faces	172	14

$172 = (1+1) + (5+5) + (10+10+10+10+10) + (15+15) + 20 + (30+30)$

Example: GIT-fan for $\mathbb{G}(2,5)$

Example

Cox ring of $\overline{M}_{0,5}$ is isomorphic to $\mathbb{Z}^5$-graded coordinate ring $R = \mathbb{K}[T_1, \ldots, T_{10}]/\mathfrak{a}$ of affine cone over $\mathbb{G}(2,5)$. Symmetry group action of

$$S_5 \cong G = \langle (2,3)(5,6)(9,10),\ (1,5,9,10,3)(2,7,8,4,6) \rangle \subset S_{10}$$

	number	number of orbits
monomial containment tests	$2^{10} = 1024$	34
$\mathfrak{a}$-faces	172	14

$172 = (1+1) + (5+5) + (10+10+10+10+10) + (15+15) + 20 + (30+30)$

$$|\Omega(5)/G| = 4$$

Example: GIT-fan for $\mathbb{G}(2,5)$

Example

Cox ring of $\overline{M}_{0,5}$ is isomorphic to $\mathbb{Z}^5$-graded coordinate ring $R = \mathbb{K}[T_1, \ldots, T_{10}]/\mathfrak{a}$ of affine cone over $\mathbb{G}(2,5)$. Symmetry group action of

$$S_5 \cong G = \langle (2,3)(5,6)(9,10),\ (1,5,9,10,3)(2,7,8,4,6) \rangle \subset S_{10}$$

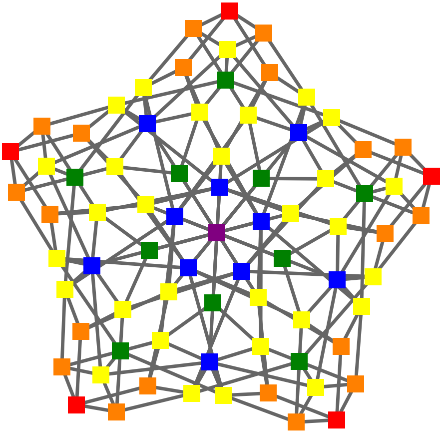
	number	number of orbits
monomial containment tests	$2^{10} = 1024$	34
$\mathfrak{a}$-faces	172	14

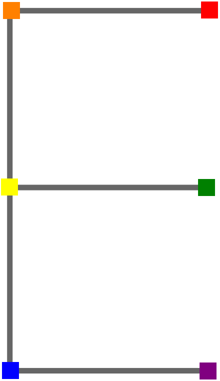
$172 = (1+1) + (5+5) + (10+10+10+10+10) + (15+15) + 20 + (30+30)$

$$|\Omega(5)/G| = 4$$

$$|\Lambda(5)| = 76 = 1 + 10 + 30 + 10 + 20 + 5$$

Adjacency graph of the maximal-dimensional GIT-cones and their orbits:





Application: Mori Chamber Decomposition of Mov($\overline{M}_{0,6}$)

Moving cone Mov($\overline{M}_{0,6}$) classifies all small modifications (rational maps which are isomorphisms on open subsets which have a complement of codimension ≥ 2).

Example

Cox ring is $\mathbb{Z}^{16}$-graded, has 40 generators (Castravet, 2009),

Application: Mori Chamber Decomposition of $\text{Mov}(\overline{M}_{0,6})$

Moving cone $\text{Mov}(\overline{M}_{0,6})$ classifies all small modifications (rational maps which are isomorphisms on open subsets which have a complement of codimension ≥ 2).

Example

Cox ring is $\mathbb{Z}^{16}$-graded, has 40 generators (Castravet, 2009), and 225 relations (Bernal Guillen, 2012),

Application: Mori Chamber Decomposition of $\mathrm{Mov}(\overline{M}_{0,6})$

Moving cone $\mathrm{Mov}(\overline{M}_{0,6})$ classifies all small modifications (rational maps which are isomorphisms on open subsets which have a complement of codimension ≥ 2).

Example

Cox ring is $\mathbb{Z}^{16}$-graded, has 40 generators (Castravet, 2009), and 225 relations (Bernal Guillen, 2012), and natural $G = S_6$–action.

Application: Mori Chamber Decomposition of Mov($\overline{M}_{0,6}$)

Moving cone Mov($\overline{M}_{0,6}$) classifies all small modifications (rational maps which are isomorphisms on open subsets which have a complement of codimension ≥ 2).

Example

Cox ring is $\mathbb{Z}^{16}$-graded, has 40 generators (Castravet, 2009), and 225 relations (Bernal Guillen, 2012), and natural $G = S_6$–action.
The moving cone Mov($\overline{M}_{0,6}$) has

$$176\,512\,225$$

GIT-cones of maximal dimension 16, which decompose into

$$249\,605$$

orbits under the S_6-action:

Application: Mori Chamber Decomposition of Mov($\overline{M}_{0,6}$)

Moving cone Mov($\overline{M}_{0,6}$) classifies all small modifications (rational maps which are isomorphisms on open subsets which have a complement of codimension ≥ 2).

Example

Cox ring is $\mathbb{Z}^{16}$-graded, has 40 generators (Castravet, 2009), and 225 relations (Bernal Guillen, 2012), and natural $G = S_6$–action.
The moving cone Mov($\overline{M}_{0,6}$) has

176 512 225

GIT-cones of maximal dimension 16, which decompose into

249 605

orbits under the S_6-action:

cardinality	1	6	10	15	20	30	45	60
no. of orbits	1	1	1	4	1	1	10	27

cardinality	72	90	120	180	240	360	720
no. of orbits	4	46	32	488	4	7934	241051

References

- J. Boehm, S. Keicher, Y. Ren. *Computing GIT-fans with symmetry and the Mori chamber decomposition of $\overline{M}_{0,6}$*, arXiv:1603.09241 (2016).
- J. Boehm, S. Keicher, Y. Ren. *gitfan.lib. A Singular library for computing the GIT fan*, SINGULAR distribution.
- S. Keicher. *Computing the GIT-fan*, Internat. J. Algebra Comput. (2012).

- D. Mumford, J. Fogarty, F. Kirwan. *Geometric invariant theory*. Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer (1994).
- I. V. Dolgachev and Y. Hu. *Variation of geometric invariant theory quotients*. Publ. Math., Inst. Hautes Etud. Sci. (1998).
- F. Berchtold, J. Hausen. *GIT equivalence beyond the ample cone*. Michigan Math. J. (2006).
- I. Arzhantsev, U. Derenthal, J. Hausen, A. Laface. *Cox Rings*, Cambridge studies in advanced mathematics (2014).

- A.-M. Castravet. *The Cox ring of $\overline{M}_{0,6}$*. Trans. Amer. Math. Soc. (2009).
- M. M. Bernal Guillen. *Relations in the Cox Ring of $\overline{M}_{0,6}$*. PhD (2012).