Algorithms in invariant theory

Visu Makam
(joint work with Harm Derksen)

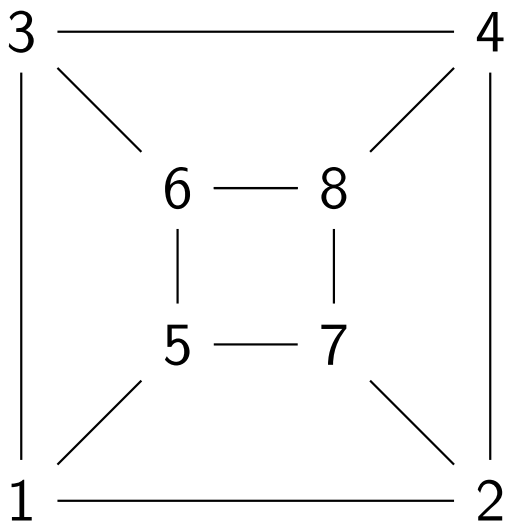
June 6, 2018

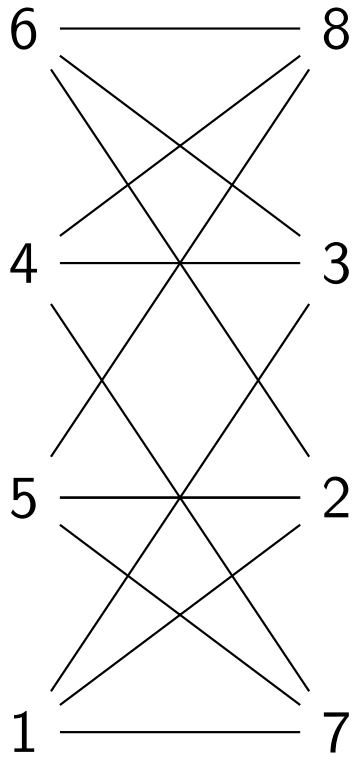
Table of contents

1. Graph isomorphism
2. Group actions and invariant rings
3. Null cone and orbit closure problems
4. Matrix semi-invariants
5. Matrix invariants
6. Orbit closure algorithms
7. Quivers
8. Positive characteristic

Graph isomorphism

- Are the two graphs isomorphic?





Graph isomorphism via group actions

- Fix a vertex set $\{1, 2, \ldots, n\}$.

$$\text{Graph } G \leftrightsquigarrow \text{adjacency matrix } A_G \in \mathrm{Mat}_{n,n}\,.$$

Graph isomorphism via group actions

- Fix a vertex set $\{1, 2, \ldots, n\}$.

$$\text{Graph } G \leftrightsquigarrow \text{adjacency matrix } A_G \in \mathrm{Mat}_{n,n}\,.$$

- $S_n \circlearrowleft \mathrm{Mat}_{n,n}$ by conjugation, i.e.,

$$\sigma \cdot A = M_\sigma A M_\sigma^{-1},$$

where M_σ is the permutation matrix associated to σ.

Graph isomorphism via group actions

- Fix a vertex set $\{1, 2, \ldots, n\}$.

$$\text{Graph } G \leftrightsquigarrow \text{adjacency matrix } A_G \in \mathrm{Mat}_{n,n}\,.$$

- $S_n \circlearrowleft \mathrm{Mat}_{n,n}$ by conjugation, i.e.,

$$\sigma \cdot A = M_\sigma A M_\sigma^{-1},$$

where M_σ is the permutation matrix associated to σ.

- $G_1 \cong G_2$ if and only if A_{G_1} and A_{G_2} are in the same orbit for the action of S_n.

Polynomial functions on vector spaces

- Let $V = K^n$ be a vector space.

Polynomial functions on vector spaces

- Let $V = K^n$ be a vector space.
- Let $e_1, \ldots, e_n$ denote the standard basis.

Polynomial functions on vector spaces

- Let $V = K^n$ be a vector space.
- Let $e_1, \ldots, e_n$ denote the standard basis.
- Let $x_1, \ldots, x_n$ be the corresponding coordinate functions.

Polynomial functions on vector spaces

- Let $V = K^n$ be a vector space.
- Let $e_1, \ldots, e_n$ denote the standard basis.
- Let $x_1, \ldots, x_n$ be the corresponding coordinate functions.
- The ring of polynomial functions is $K[x_1, \ldots, x_n]$.

Invariant polynomials

- Suppose we have a group G acting on V by linear transformations.

Invariant polynomials

- Suppose we have a group G acting on V by linear transformations.
- A function is called invariant if it is constant on orbits,

Invariant polynomials

- Suppose we have a group G acting on V by linear transformations.
- A function is called invariant if it is constant on orbits, i.e.,

$$f(g \cdot v) = f(v) \text{ for all } g \in G \text{ and } v \in V.$$

Invariant polynomials

- Suppose we have a group G acting on V by linear transformations.
- A function is called invariant if it is constant on orbits, i.e.,

$$f(g \cdot v) = f(v) \text{ for all } g \in G \text{ and } v \in V.$$

- The invariant ring $K[x_1, ..., x_n]^G$ is the subring consisting of invariant polynomials.

Invariant polynomials

- Suppose we have a group G acting on V by linear transformations.
- A function is called invariant if it is constant on orbits, i.e.,

$$f(g \cdot v) = f(v) \text{ for all } g \in G \text{ and } v \in V.$$

- The invariant ring $K[x_1, ..., x_n]^G$ is the subring consisting of invariant polynomials.
- This is a graded (by degree) subring of $K[x_1, ..., x_n]$.

Graph isomorphism revisited

- Recall $G_1 \cong G_2$ if and only if A_{G_1} and A_{G_2} are in the same orbit.

Graph isomorphism revisited

- Recall $G_1 \cong G_2$ if and only if A_{G_1} and A_{G_2} are in the same orbit.
- Suppose f is an invariant polynomial such that $f(A_{G_1}) \neq f(A_{G_2})$, then $G_1 \ncong G_2$.

Graph isomorphism revisited

- Recall $G_1 \cong G_2$ if and only if A_{G_1} and A_{G_2} are in the same orbit.
- Suppose f is an invariant polynomial such that $f(A_{G_1}) \neq f(A_{G_2})$, then $G_1 \ncong G_2$.
- **Fact:** For finite group actions, orbits can be distinguished by invariant polynomials.

Graph isomorphism revisited

- Recall $G_1 \cong G_2$ if and only if A_{G_1} and A_{G_2} are in the same orbit.
- Suppose f is an invariant polynomial such that $f(A_{G_1}) \neq f(A_{G_2})$, then $G_1 \ncong G_2$.
- **Fact:** For finite group actions, orbits can be distinguished by invariant polynomials.
- **Recall:** Invariant rings are finitely generated.

Graph isomorphism revisited

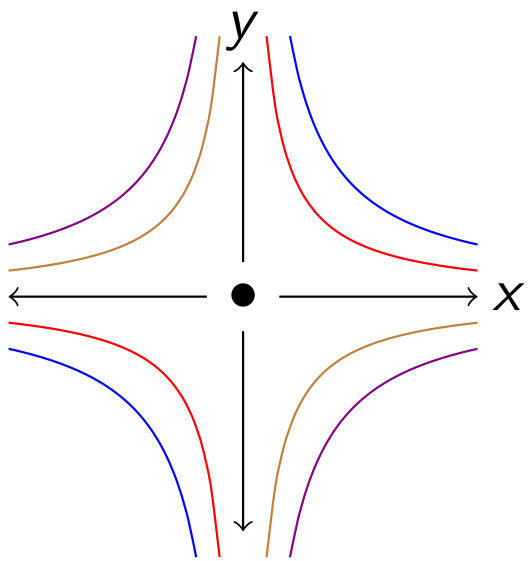
- Recall $G_1 \cong G_2$ if and only if A_{G_1} and A_{G_2} are in the same orbit.
- Suppose f is an invariant polynomial such that $f(A_{G_1}) \neq f(A_{G_2})$, then $G_1 \ncong G_2$.
- **Fact:** For finite group actions, orbits can be distinguished by invariant polynomials.
- **Recall:** Invariant rings are finitely generated.
- Algorithm for Graph isomorphism: Find a finite set of generators for $K[x_{i,j} | 1 \leq i, j \leq n]^{S_n}$, and test all.

An example

- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.

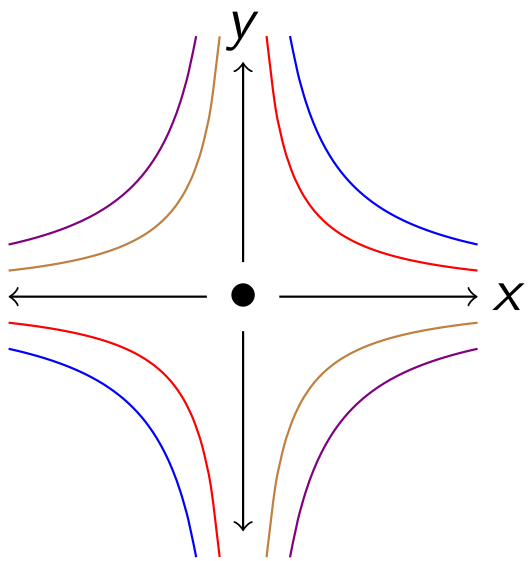
An example

- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



An example

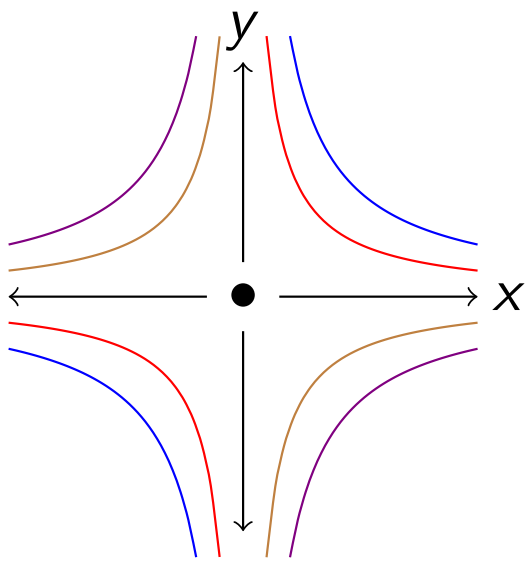
- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



- What are the orbits?

An example

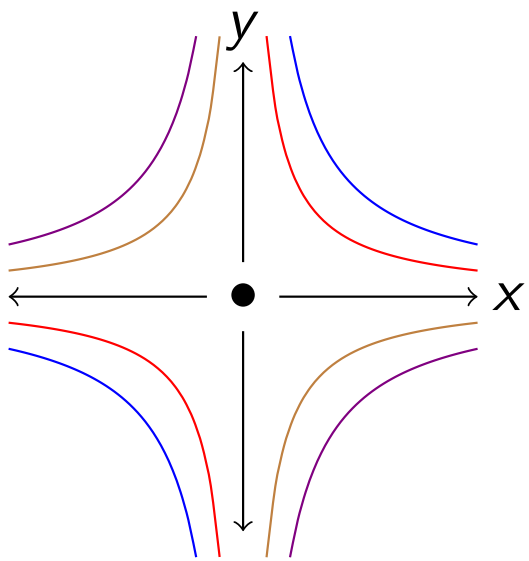
- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



- What are the orbits?
  1. $xy = k$ for $k \neq 0$;

An example

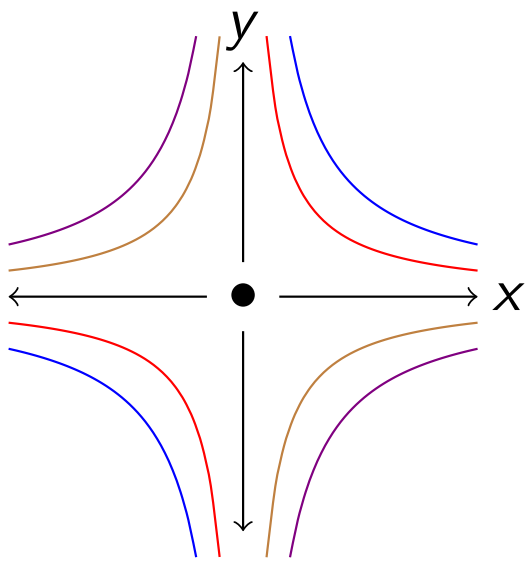
- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



- What are the orbits?
  1. $xy = k$ for $k \neq 0$;
  2. x-axis minus the origin;

An example

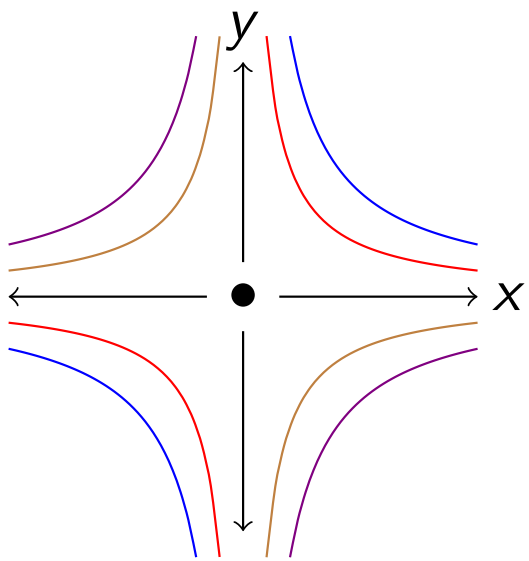
- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



- What are the orbits?
  1. $xy = k$ for $k \neq 0$;
  2. x-axis minus the origin;
  3. y-axis minus the origin;

An example

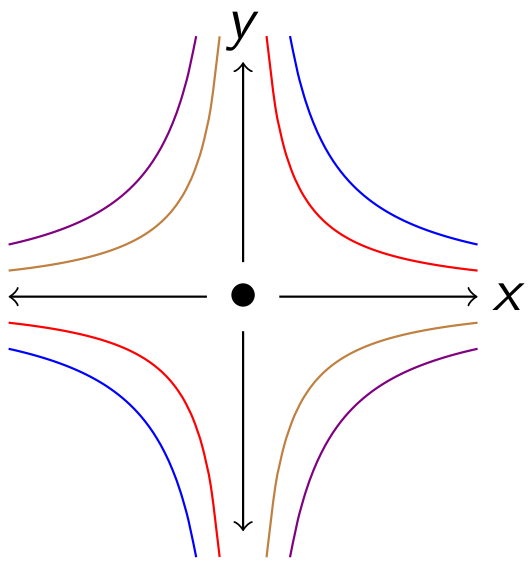
- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



- What are the orbits?
  1. $xy = k$ for $k \neq 0$;
  2. x-axis minus the origin;
  3. y-axis minus the origin;
  4. The origin.

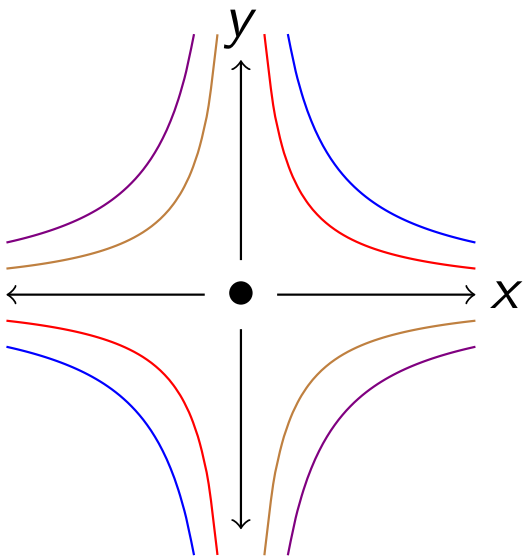
An example

- Consider $\mathbb{C}^*$ acting on $\mathbb{C}^2$ by scaling, i.e., $t \cdot (a, b) = (ta, t^{-1}b)$.



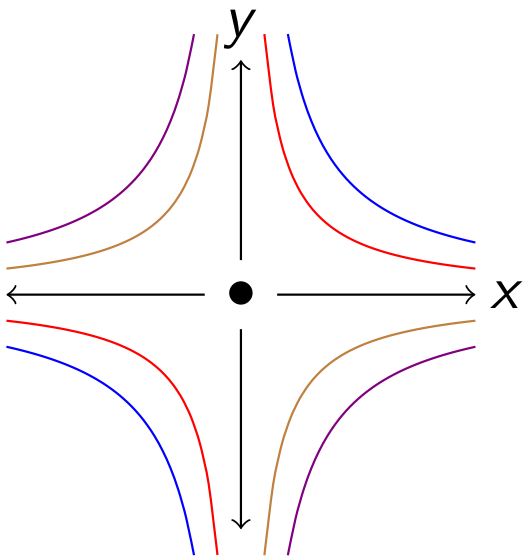
- What are the orbits?
  1. $xy = k$ for $k \neq 0$;
  2. x-axis minus the origin;
  3. y-axis minus the origin;
  4. The origin.
- Can invariant polynomials distinguish between orbits?

Example continued



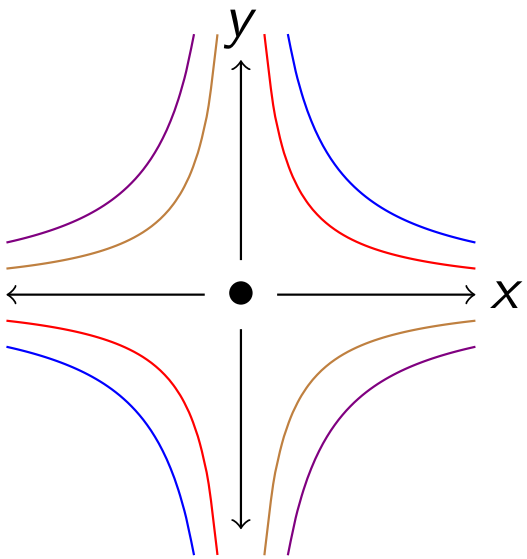
- The invariant ring is $K[xy]$.

Example continued



- The invariant ring is $K[xy]$.
- Can't distinguish between the three orbits: x-axis minus origin, y-axis minus origin and the origin. Why?

Example continued



- The invariant ring is $K[xy]$.
- Can't distinguish between the three orbits: x-axis minus origin, y-axis minus origin and the origin. Why?
- Their orbit closures intersect!

Orbit closure problem

Problem (Orbit closure problem)

$G \curvearrowright V$. Decide if orbit closures of v and w intersect.

Orbit closure problem

Problem (Orbit closure problem)

$G \curvearrowright V$. Decide if orbit closures of v and w intersect.

Theorem

The orbit closures of two points intersect if and only if they cannot be distinguished by an invariant polynomial.

Orbit closure problem

Problem (Orbit closure problem)

$G \curvearrowright V$. Decide if orbit closures of v and w intersect.

Theorem

The orbit closures of two points intersect if and only if they cannot be distinguished by an invariant polynomial.

- Above theorem requires group to be reductive.

Orbit closure problem

Problem (Orbit closure problem)

$G \circlearrowright V$. Decide if orbit closures of v and w intersect.

Theorem

The orbit closures of two points intersect if and only if they cannot be distinguished by an invariant polynomial.

- Above theorem requires group to be reductive.
- All classical groups are reductive - GL_n, SL_n, Sp_n, O_n, tori, finite groups etc.

Null cone

Problem (Null cone membership problem)

Decide if a given point is in the null cone.

Null cone

Problem (Null cone membership problem)

Decide if a given point is in the null cone.

- Recall:

$$\text{null cone} = \{v \in V \mid 0 \in \overline{G \cdot v}\}.$$

Null cone

Problem (Null cone membership problem)

Decide if a given point is in the null cone.

- Recall:

$$\text{null cone} = \{v \in V \mid 0 \in \overline{G \cdot v}\}.$$

- Equivalently:

null cone = Zero set of homogeneous invariants of positive degree.

Null cone

Problem (Null cone membership problem)

Decide if a given point is in the null cone.

- Recall:

$$\text{null cone} = \{v \in V \mid 0 \in \overline{G \cdot v}\}.$$

- Equivalently:

null cone = Zero set of homogeneous invariants of positive degree.

- A special case of the orbit closure problem!

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.
- Algorithm for null cone: Check if $f_i(v) \neq 0$ for some i.

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.
- Algorithm for null cone: Check if $f_i(v) \neq 0$ for some i.
- Algorithm for orbit closure: Check if $f_i(v) \neq f_i(w)$ for some i.

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.
- Algorithm for null cone: Check if $f_i(v) \neq 0$ for some i.
- Algorithm for orbit closure: Check if $f_i(v) \neq f_i(w)$ for some i.
- How can we get algorithms if we are not in the best case scenario?

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.
- Algorithm for null cone: Check if $f_i(v) \neq 0$ for some i.
- Algorithm for orbit closure: Check if $f_i(v) \neq f_i(w)$ for some i.
- How can we get algorithms if we are not in the best case scenario?
- Start with degree bounds.

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.
- Algorithm for null cone: Check if $f_i(v) \neq 0$ for some i.
- Algorithm for orbit closure: Check if $f_i(v) \neq f_i(w)$ for some i.
- How can we get algorithms if we are not in the best case scenario?
- Start with degree bounds.
- **Main problems:** degree bounds, null cone and orbit closure.

If life was easy...

- **Best case scenario:** We have a small set of (homogeneous) generators $f_1, \ldots, f_r$ that are efficient to compute.
- Algorithm for null cone: Check if $f_i(v) \neq 0$ for some i.
- Algorithm for orbit closure: Check if $f_i(v) \neq f_i(w)$ for some i.
- How can we get algorithms if we are not in the best case scenario?
- Start with degree bounds.
- **Main problems:** degree bounds, null cone and orbit closure.
- **Main objects:** Matrix invariants and matrix semi-invariants.

Matrix semi-invariants

- Consider the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on $\mathrm{Mat}_{n,n}^m$ given by:

$$(P, Q) \cdot (X_1, ..., X_m) = (PX_1Q^{-1}, ..., PX_mQ^{-1}).$$

Matrix semi-invariants

- Consider the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on $\mathrm{Mat}_{n,n}^m$ given by:

$$(P, Q) \cdot (X_1, ..., X_m) = (PX_1Q^{-1}, ..., PX_mQ^{-1}).$$

- $\det(X_1)$ is an invariant polynomial.

Matrix semi-invariants

- Consider the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on $\mathrm{Mat}_{n,n}^m$ given by:

$$(P, Q) \cdot (X_1, ..., X_m) = (PX_1Q^{-1}, ..., PX_mQ^{-1}).$$

- $\det(X_1)$ is an invariant polynomial.
- $\det(\sum_i c_i X_i)$ is also an invariant polynomial.

Matrix semi-invariants

- Consider the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on $\mathrm{Mat}_{n,n}^m$ given by:

$$(P, Q) \cdot (X_1, ..., X_m) = (PX_1Q^{-1}, ..., PX_mQ^{-1}).$$

- $\det(X_1)$ is an invariant polynomial.
- $\det(\sum_i c_i X_i)$ is also an invariant polynomial.
- Are these all?

Matrix semi-invariants

- Consider the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on $\mathrm{Mat}_{n,n}^m$ given by:

$$(P, Q) \cdot (X_1, ..., X_m) = (PX_1Q^{-1}, ..., PX_mQ^{-1}).$$

- $\det(X_1)$ is an invariant polynomial.
- $\det(\sum_i c_i X_i)$ is also an invariant polynomial.
- Are these all?
- How about $\det \begin{pmatrix} X_1 & X_2 \\ X_2 & X_3 \end{pmatrix}$?

Description of invariants

- For $T = (T_1, ..., T_m) \in \mathrm{Mat}_{d,d}^m$, we define an invariant f_T by

Description of invariants

- For $T = (T_1, ..., T_m) \in \mathrm{Mat}_{d,d}^m$, we define an invariant f_T by

$$f_T(X_1, ..., X_m) = \det(X_1 \otimes T_1 + \cdots + X_m \otimes T_m).$$

Description of invariants

- For $T = (T_1, ..., T_m) \in \mathrm{Mat}_{d,d}^m$, we define an invariant f_T by

$$f_T(X_1, ..., X_m) = \det(X_1 \otimes T_1 + \cdots + X_m \otimes T_m).$$

- f_T is a homogeneous invariant of degree dn.

Description of invariants

- For $T = (T_1, ..., T_m) \in \mathrm{Mat}_{d,d}^m$, we define an invariant f_T by

$$f_T(X_1, ..., X_m) = \det(X_1 \otimes T_1 + \cdots + X_m \otimes T_m).$$

- f_T is a homogeneous invariant of degree dn.

Theorem (Derksen-Weyman, Domokos-Zubkov, Schofield-Van den Bergh)

The linear span of $\{f_T \mid T \in \mathrm{Mat}_{d,d}^m\}$ *gives all degree dn invariants.*

Description of invariants

- For $T = (T_1, ..., T_m) \in \mathrm{Mat}_{d,d}^m$, we define an invariant f_T by

$$f_T(X_1, ..., X_m) = \det(X_1 \otimes T_1 + \cdots + X_m \otimes T_m).$$

- f_T is a homogeneous invariant of degree dn.

Theorem (Derksen-Weyman, Domokos-Zubkov, Schofield-Van den Bergh)

The linear span of $\{f_T \mid T \in \mathrm{Mat}_{d,d}^m\}$ *gives all degree* dn *invariants.*

- No homogeneous invariants of degree k unless $n|k$.

Characterizations of the null cone

The following are equivalent (Recall from K.V.'s talk):

Characterizations of the null cone

The following are equivalent (Recall from K.V.'s talk):

- $(A_1, \ldots, A_m)$ not in null cone.

Characterizations of the null cone

The following are equivalent (Recall from K.V.'s talk):

- $(A_1, \ldots, A_m)$ not in null cone.
- $f_T(A_1, \ldots, A_m) = \det(A_1 \otimes T_1 + \cdots + A_m \otimes T_m) \neq 0$ for some T.

Characterizations of the null cone

The following are equivalent (Recall from K.V.'s talk):

- $(A_1, \ldots, A_m)$ not in null cone.
- $f_T(A_1, \ldots, A_m) = \det(A_1 \otimes T_1 + \cdots + A_m \otimes T_m) \neq 0$ for some T.
- Some blow-up contains an invertible matrix.

Characterizations of the null cone

The following are equivalent (Recall from K.V.'s talk):

- $(A_1, \ldots, A_m)$ not in null cone.
- $f_T(A_1, \ldots, A_m) = \det(A_1 \otimes T_1 + \cdots + A_m \otimes T_m) \neq 0$ for some T.
- Some blow-up contains an invertible matrix.
- $\operatorname{ncrk}(t_1 A_1 + \cdots + t_m A_m)$ is full.

Characterizations of the null cone

The following are equivalent (Recall from K.V.'s talk):

- $(A_1, \ldots, A_m)$ not in null cone.
- $f_T(A_1, \ldots, A_m) = \det(A_1 \otimes T_1 + \cdots + A_m \otimes T_m) \neq 0$ for some T.
- Some blow-up contains an invertible matrix.
- $\operatorname{ncrk}(t_1 A_1 + \cdots + t_m A_m)$ is full.
- There is no shrunk subspace.

Results on matrix semi-invariants

Theorem (Derksen, Makam)

For every $d \geq n-1$, $(A_1, \ldots, A_m)$ is not in null cone if and only if there exists $T \in \mathrm{Mat}_{d,d}^m$ such that $f_T(A) \neq 0$.

Results on matrix semi-invariants

Theorem (Derksen, Makam)

For every $d \geq n-1$, $(A_1, \ldots, A_m)$ is not in null cone if and only if there exists $T \in \mathrm{Mat}_{d,d}^m$ such that $f_T(A) \neq 0$.

- Equivalently, there is an invertible matrix in the d^{th} blow-up.

Results on matrix semi-invariants

Theorem (Derksen, Makam)

For every $d \geq n-1$, $(A_1, \ldots, A_m)$ is not in null cone if and only if there exists $T \in \mathrm{Mat}_{d,d}^m$ such that $f_T(A) \neq 0$.

- Equivalently, there is an invertible matrix in the d^{th} blow-up.
- IQS algorithm = constructive version of above result.

Results on matrix semi-invariants

Theorem (Derksen, Makam)

For every $d \geq n-1$, $(A_1, \ldots, A_m)$ is not in null cone if and only if there exists $T \in \mathrm{Mat}_{d,d}^m$ such that $f_T(A) \neq 0$.

- Equivalently, there is an invertible matrix in the d^{th} blow-up.
- IQS algorithm = constructive version of above result.
- Invariants of polynomial degree suffice to define the null cone.

Results on matrix semi-invariants

Theorem (Derksen, Makam)

For every $d \geq n-1$, $(A_1, \ldots, A_m)$ is not in null cone if and only if there exists $T \in \mathrm{Mat}_{d,d}^m$ such that $f_T(A) \neq 0$.

- Equivalently, there is an invertible matrix in the d^{th} blow-up.
- IQS algorithm = constructive version of above result.
- Invariants of polynomial degree suffice to define the null cone.

Corollary (Derksen, Makam)

We have an upper bound of n^6 for the degree of generators for matrix semi-invariants.

Results on matrix semi-invariants

Theorem (Derksen, Makam)

For every $d \geq n-1$, $(A_1, \ldots, A_m)$ is not in null cone if and only if there exists $T \in \mathrm{Mat}_{d,d}^m$ such that $f_T(A) \neq 0$.

- Equivalently, there is an invertible matrix in the d^{th} blow-up.
- IQS algorithm = constructive version of above result.
- Invariants of polynomial degree suffice to define the null cone.

Corollary (Derksen, Makam)

We have an upper bound of n^6 for the degree of generators for matrix semi-invariants.

- Randomized algorithm for null cone and orbit closure problems!

Derksen's bound in perspective

- Bound for null cone $\approx$ Randomized algorithm for null cone.

Derksen's bound in perspective

- Bound for null cone $\approx$ Randomized algorithm for null cone.
- Bound for generators $\approx$ Randomized algorithm for orbit closure.

Derksen's bound in perspective

- Bound for null cone $\approx$ Randomized algorithm for null cone.
- Bound for generators $\approx$ Randomized algorithm for orbit closure.

Theorem (Derksen)

Degree bound for generators = Poly(bound for null cone).

Derksen's bound in perspective

- Bound for null cone $\approx$ Randomized algorithm for null cone.
- Bound for generators $\approx$ Randomized algorithm for orbit closure.

Theorem (Derksen)

Degree bound for generators = Poly(bound for null cone).

- Randomized algorithm for null cone $\approx$ Randomized algorithm for orbit closure

How about deterministic algorithms?

- Deterministic algorithm for null cone $\overset{?}{\approx}$ Deterministic algorithm for orbit closure

How about deterministic algorithms?

- Deterministic algorithm for null cone $\overset{?}{\approx}$ Deterministic algorithm for orbit closure

- GGOW, IQS give deterministic algorithm for null cone for matrix semi-invariants.

How about deterministic algorithms?

- Deterministic algorithm for null cone $\overset{?}{\approx}$ Deterministic algorithm for orbit closure

- GGOW, IQS give deterministic algorithm for null cone for matrix semi-invariants.
- Is there a deterministic algorithm for the orbit closure problem for matrix semi-invariants?

How about deterministic algorithms?

- Deterministic algorithm for null cone $\overset{?}{\approx}$ Deterministic algorithm for orbit closure

- GGOW, IQS give deterministic algorithm for null cone for matrix semi-invariants.
- Is there a deterministic algorithm for the orbit closure problem for matrix semi-invariants?
- We will need to look at matrix invariants first.

Matrix invariants

- Consider the simultaneous conjugation action of GL_n on $\mathrm{Mat}_{n,n}^m$ given by

$$g \cdot (X_1, \ldots, X_m) = (gX_1g^{-1}, \ldots, gX_mg^{-1}).$$

Matrix invariants

- Consider the simultaneous conjugation action of GL_n on $\mathrm{Mat}_{n,n}^m$ given by
$$g \cdot (X_1, \ldots, X_m) = (gX_1g^{-1}, \ldots, gX_mg^{-1}).$$

- $\mathrm{Tr}(X_1), \mathrm{Tr}(X_1X_2X_3), \mathrm{Tr}(X_1X_3X_2)$ are all invariant polynomials.

Matrix invariants

- Consider the simultaneous conjugation action of GL_n on $\mathrm{Mat}_{n,n}^m$ given by
$$g \cdot (X_1, \ldots, X_m) = (gX_1g^{-1}, \ldots, gX_mg^{-1}).$$

- $\mathrm{Tr}(X_1), \mathrm{Tr}(X_1X_2X_3), \mathrm{Tr}(X_1X_3X_2)$ are all invariant polynomials.
- For a word $w = i_1 i_2 \ldots i_k$ with $i_j \in \{1, 2, \ldots, m\}$, define $X_w = X_{i_1} X_{i_2} \ldots X_{i_k}$.

Matrix invariants

- Consider the simultaneous conjugation action of GL_n on $\mathrm{Mat}_{n,n}^m$ given by
$$g \cdot (X_1, \ldots, X_m) = (gX_1g^{-1}, \ldots, gX_mg^{-1}).$$

- $\mathrm{Tr}(X_1), \mathrm{Tr}(X_1X_2X_3), \mathrm{Tr}(X_1X_3X_2)$ are all invariant polynomials.
- For a word $w = i_1 i_2 \ldots i_k$ with $i_j \in \{1, 2, \ldots, m\}$, define $X_w = X_{i_1} X_{i_2} \ldots X_{i_k}$.
- Observe that $\mathrm{Tr}(X_w)$ is an invariant polynomial (of deg k).

Matrix invariants

Theorem (Procesi/Sibirskii)

Invariants of the form $Tr(X_w)$ form an (infinite) generating set (in $\mathrm{char}\,0$*).*

Matrix invariants

Theorem (Procesi/Sibirskii)

Invariants of the form $Tr(X_w)$ form an (infinite) generating set (in $\mathrm{char}\,0$*).*

Theorem (Razmyslov)

Traces of words of length $\leq n^2$ already form a generating set.

Matrix invariants

Theorem (Procesi/Sibirskii)

Invariants of the form $Tr(X_w)$ form an (infinite) generating set (in $\mathrm{char}\,0$*).*

Theorem (Razmyslov)

Traces of words of length $\leq n^2$ already form a generating set.

- In other words, a degree bound of n^2.

Matrix invariants

Theorem (Procesi/Sibirskii)

Invariants of the form $Tr(X_w)$ form an (infinite) generating set (in char 0*).*

Theorem (Razmyslov)

Traces of words of length $\leq n^2$ already form a generating set.

- In other words, a degree bound of n^2.
- However, this is still an exponentially large generating set!

Orbit closure problem for matrix invariants

- Two algorithms.

Orbit closure problem for matrix invariants

- Two algorithms.
- **Algorithm 1:** (Forbes, Shpilka)

Orbit closure problem for matrix invariants

- Two algorithms.
- **Algorithm 1:** (Forbes, Shpilka)
- (Mulmuley's construction:) Given $A = (A_1, \ldots, A_m) \in \mathrm{Mat}_{n,n}^m$, construct a (non-commutative) polynomial $P_d(A)$.

Orbit closure problem for matrix invariants

- Two algorithms.
- **Algorithm 1:** (Forbes, Shpilka)
- (Mulmuley's construction:) Given $A = (A_1, \ldots, A_m) \in \mathrm{Mat}_{n,n}^m$, construct a (non-commutative) polynomial $P_d(A)$.
- Coefficient of any monomial in $P_d(A)$ is of the form $Tr(A_{i_1} A_{i_2} \ldots A_{i_d})$ (and all words of length d appear).

Orbit closure problem for matrix invariants

- Two algorithms.
- **Algorithm 1:** (Forbes, Shpilka)
- (Mulmuley's construction:) Given $A = (A_1, \ldots, A_m) \in \mathrm{Mat}_{n,n}^m$, construct a (non-commutative) polynomial $P_d(A)$.
- Coefficient of any monomial in $P_d(A)$ is of the form $Tr(A_{i_1} A_{i_2} \ldots A_{i_d})$ (and all words of length d appear).
- Check if $P_d(A) - P_d(B) = 0$ for $1 \leq d \leq n^2$.

Orbit closure problem for matrix invariants

- Two algorithms.
- **Algorithm 1:** (Forbes, Shpilka)
- (Mulmuley's construction:) Given $A = (A_1, \ldots, A_m) \in \mathrm{Mat}_{n,n}^m$, construct a (non-commutative) polynomial $P_d(A)$.
- Coefficient of any monomial in $P_d(A)$ is of the form $Tr(A_{i_1} A_{i_2} \ldots A_{i_d})$ (and all words of length d appear).
- Check if $P_d(A) - P_d(B) = 0$ for $1 \leq d \leq n^2$.
- **Raz-Shpilka:** PIT for non-commutative polynomials (ROABPs) is in polynomial time.

Our algorithm

- **Algorithm 2:** (Derksen, Makam).

Our algorithm

- **Algorithm 2:** (Derksen, Makam).
- Given $(A_1, \ldots, A_m)$ and $(B_1, \ldots, B_m)$, construct for all i

$$C_i = \begin{pmatrix} A_i & 0 \\ 0 & B_i \end{pmatrix}.$$

Our algorithm

- **Algorithm 2:** (Derksen, Makam).
- Given $(A_1, \ldots, A_m)$ and $(B_1, \ldots, B_m)$, construct for all i

$$C_i = \begin{pmatrix} A_i & 0 \\ 0 & B_i \end{pmatrix}.$$

- Consider the algebra $\mathcal{C} \subset \mathrm{Mat}_{2n,2n}$ generated by $C_1, \ldots, C_m$.

Our algorithm

- **Algorithm 2:** (Derksen, Makam).
- Given $(A_1, \ldots, A_m)$ and $(B_1, \ldots, B_m)$, construct for all i

$$C_i = \begin{pmatrix} A_i & 0 \\ 0 & B_i \end{pmatrix}.$$

- Consider the algebra $\mathcal{C} \subset \mathrm{Mat}_{2n,2n}$ generated by $C_1, \ldots, C_m$.
- $\mathcal{C}$ is spanned by words

$$C_w = C_{i_1} C_{i_2} \ldots C_{i_k} = \begin{pmatrix} A_{i_1} \ldots A_{i_k} & 0 \\ 0 & B_{i_1} \ldots B_{i_k} \end{pmatrix}.$$

Our algorithm

- **Algorithm 2:** (Derksen, Makam).
- Given $(A_1, \ldots, A_m)$ and $(B_1, \ldots, B_m)$, construct for all i

$$C_i = \begin{pmatrix} A_i & 0 \\ 0 & B_i \end{pmatrix}.$$

- Consider the algebra $\mathcal{C} \subset \mathrm{Mat}_{2n,2n}$ generated by $C_1, \ldots, C_m$.
- $\mathcal{C}$ is spanned by words

$$C_w = C_{i_1} C_{i_2} \ldots C_{i_k} = \begin{pmatrix} A_{i_1} \ldots A_{i_k} & 0 \\ 0 & B_{i_1} \ldots B_{i_k} \end{pmatrix}.$$

- **Key idea:** Can extract in polynomial time a subset of this spanning set which forms a basis.

Our algorithm continued..

- If $C_w = C_u + C_v$, i.e.,

$$\begin{pmatrix} A_w & 0 \\ 0 & B_w \end{pmatrix} = \begin{pmatrix} A_u & 0 \\ 0 & B_u \end{pmatrix} + \begin{pmatrix} A_v & 0 \\ 0 & B_v \end{pmatrix}$$

Our algorithm continued..

- If $C_w = C_u + C_v$, i.e.,

$$\begin{pmatrix} A_w & 0 \\ 0 & B_w \end{pmatrix} = \begin{pmatrix} A_u & 0 \\ 0 & B_u \end{pmatrix} + \begin{pmatrix} A_v & 0 \\ 0 & B_v \end{pmatrix}$$

- We have $\mathrm{Tr}(A_w) = \mathrm{Tr}(A_u) + \mathrm{Tr}(A_v)$ and $\mathrm{Tr}(B_w) = \mathrm{Tr}(B_u) + \mathrm{Tr}(B_v)$.

Our algorithm continued..

- If $C_w = C_u + C_v$, i.e.,

$$\begin{pmatrix} A_w & 0 \\ 0 & B_w \end{pmatrix} = \begin{pmatrix} A_u & 0 \\ 0 & B_u \end{pmatrix} + \begin{pmatrix} A_v & 0 \\ 0 & B_v \end{pmatrix}$$

- We have $\mathrm{Tr}(A_w) = \mathrm{Tr}(A_u) + \mathrm{Tr}(A_v)$ and $\mathrm{Tr}(B_w) = \mathrm{Tr}(B_u) + \mathrm{Tr}(B_v)$.
- So, checking $\mathrm{Tr}(A_w) = \mathrm{Tr}(B_w)$ is redundant!

Our algorithm continued..

- If $C_w = C_u + C_v$, i.e.,

$$\begin{pmatrix} A_w & 0 \\ 0 & B_w \end{pmatrix} = \begin{pmatrix} A_u & 0 \\ 0 & B_u \end{pmatrix} + \begin{pmatrix} A_v & 0 \\ 0 & B_v \end{pmatrix}$$

- We have $\mathrm{Tr}(A_w) = \mathrm{Tr}(A_u) + \mathrm{Tr}(A_v)$ and $\mathrm{Tr}(B_w) = \mathrm{Tr}(B_u) + \mathrm{Tr}(B_v)$.
- So, checking $\mathrm{Tr}(A_w) = \mathrm{Tr}(B_w)$ is redundant!

Algorithm:

Our algorithm continued..

- If $C_w = C_u + C_v$, i.e.,

$$\begin{pmatrix} A_w & 0 \\ 0 & B_w \end{pmatrix} = \begin{pmatrix} A_u & 0 \\ 0 & B_u \end{pmatrix} + \begin{pmatrix} A_v & 0 \\ 0 & B_v \end{pmatrix}$$

- We have $\mathrm{Tr}(A_w) = \mathrm{Tr}(A_u) + \mathrm{Tr}(A_v)$ and $\mathrm{Tr}(B_w) = \mathrm{Tr}(B_u) + \mathrm{Tr}(B_v)$.
- So, checking $\mathrm{Tr}(A_w) = \mathrm{Tr}(B_w)$ is redundant!

Algorithm:

1. Extract a basis $C_{w_1}, C_{w_2}, \ldots, C_{w_r}$ of $\mathcal{C}$.

Our algorithm continued..

- If $C_w = C_u + C_v$, i.e.,

$$\begin{pmatrix} A_w & 0 \\ 0 & B_w \end{pmatrix} = \begin{pmatrix} A_u & 0 \\ 0 & B_u \end{pmatrix} + \begin{pmatrix} A_v & 0 \\ 0 & B_v \end{pmatrix}$$

- We have $\mathrm{Tr}(A_w) = \mathrm{Tr}(A_u) + \mathrm{Tr}(A_v)$ and $\mathrm{Tr}(B_w) = \mathrm{Tr}(B_u) + \mathrm{Tr}(B_v)$.
- So, checking $\mathrm{Tr}(A_w) = \mathrm{Tr}(B_w)$ is redundant!

Algorithm:

1. Extract a basis $C_{w_1}, C_{w_2}, \ldots, C_{w_r}$ of $\mathcal{C}$.
2. Check if $\mathrm{Tr}(A_{w_i}) = \mathrm{Tr}(B_{w_i})$ for $i = 1, 2, \ldots, r$.

Some observations

- The size of the basis is at most $2n^2$. And we don't need the degree bound!

Some observations

- The size of the basis is at most $2n^2$. And we don't need the degree bound!
- Analyzing combinatorially, we can show that invariants of degree $2n^{1.5}$ already suffice to separate orbit closures!

Some observations

- The size of the basis is at most $2n^2$. And we don't need the degree bound!
- Analyzing combinatorially, we can show that invariants of degree $2n^{1.5}$ already suffice to separate orbit closures!
- For generating invariants, lower bounds of degree $n(n+1)/2$ are known!

Some observations

- The size of the basis is at most $2n^2$. And we don't need the degree bound!
- Analyzing combinatorially, we can show that invariants of degree $2n^{1.5}$ already suffice to separate orbit closures!
- For generating invariants, lower bounds of degree $n(n+1)/2$ are known!
- Algorithm giving an algebraic result!

Matrix semi-invariants revisited

- Recall the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on m-tuples of $n \times n$ matrices

$$(P, Q) \cdot (X_1, \ldots, X_m) = (PX_1Q^{-1}, \ldots, PX_mQ^{-1}).$$

Matrix semi-invariants revisited

- Recall the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on m-tuples of $n \times n$ matrices

$$(P, Q) \cdot (X_1, \ldots, X_m) = (PX_1Q^{-1}, \ldots, PX_mQ^{-1}).$$

- For $T = (T_1, \ldots, T_m)$, we defined $f_T(X) = \det(\sum_i X_i \otimes T_i)$

Matrix semi-invariants revisited

- Recall the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on m-tuples of $n \times n$ matrices

$$(P, Q) \cdot (X_1, \ldots, X_m) = (PX_1Q^{-1}, \ldots, PX_mQ^{-1}).$$

- For $T = (T_1, \ldots, T_m)$, we defined $f_T(X) = \det(\sum_i X_i \otimes T_i)$
- f_Ts are a spanning set of invariants.

Matrix semi-invariants revisited

- Recall the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on m-tuples of $n \times n$ matrices

$$(P, Q) \cdot (X_1, \ldots, X_m) = (PX_1Q^{-1}, \ldots, PX_mQ^{-1}).$$

- For $T = (T_1, \ldots, T_m)$, we defined $f_T(X) = \det(\sum_i X_i \otimes T_i)$
- f_Ts are a spanning set of invariants.
- Want to decide if the orbit closures of $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$ intersect.

Matrix semi-invariants revisited

- Recall the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on m-tuples of $n \times n$ matrices

$$(P, Q) \cdot (X_1, \ldots, X_m) = (PX_1Q^{-1}, \ldots, PX_mQ^{-1}).$$

- For $T = (T_1, \ldots, T_m)$, we defined $f_T(X) = \det(\sum_i X_i \otimes T_i)$
- f_Ts are a spanning set of invariants.
- Want to decide if the orbit closures of $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$ intersect.
- Equivalently, want to know if $f_T(A) \neq f_T(B)$ for some T.

Orbit closure for matrix semi-invariants

- Do the orbit closures for $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$ intersect for the left-right action?

Orbit closure for matrix semi-invariants

- Do the orbit closures for $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$ intersect for the left-right action?
- **Case 1**: A or B is in the null cone.

Orbit closure for matrix semi-invariants

- Do the orbit closures for $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$ intersect for the left-right action?
- **Case 1**: A or B is in the null cone.

  IQS algorithm suffices!

Orbit closure for matrix semi-invariants

- Do the orbit closures for $A = (A_1, \ldots, A_m)$ and $B = (B_1, \ldots, B_m)$ intersect for the left-right action?
- **Case 1**: A or B is in the null cone.

  IQS algorithm suffices!
- From now on, assume w.l.o.g that A and B are not in the null cone.

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.
- Special case: $A = (I, A_2, \ldots, A_m)$

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.
- Special case: $A = (I, A_2, \ldots, A_m)$
- Either $\det(B_1) = 1$ or their orbit closures do not intersect.

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.
- Special case: $A = (I, A_2, \ldots, A_m)$
- Either $\det(B_1) = 1$ or their orbit closures do not intersect.
- If $\det(B_1) = 1$, then can find $(P, Q) \in \mathrm{SL}_n \times \mathrm{SL}_n$ s.t $PB_1Q^{-1} = I$.

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.
- Special case: $A = (I, A_2, \ldots, A_m)$
- Either $\det(B_1) = 1$ or their orbit closures do not intersect.
- If $\det(B_1) = 1$, then can find $(P, Q) \in \mathrm{SL}_n \times \mathrm{SL}_n$ s.t $PB_1Q^{-1} = I$.
- Replace B by $\widetilde{B} = (P, Q) \cdot B = (I, \widetilde{B_2}, \ldots, \widetilde{B_m})$.

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.
- Special case: $A = (I, A_2, \ldots, A_m)$
- Either $\det(B_1) = 1$ or their orbit closures do not intersect.
- If $\det(B_1) = 1$, then can find $(P, Q) \in \mathrm{SL}_n \times \mathrm{SL}_n$ s.t $PB_1Q^{-1} = I$.
- Replace B by $\widetilde{B} = (P, Q) \cdot B = (I, \widetilde{B_2}, \ldots, \widetilde{B_m})$.
- Suffices to check if orbit closures of $(A_2, \ldots, A_m)$ and $(\widetilde{B_2}, \ldots, \widetilde{B_m})$ intersect for the simultaneous conjugation action!

Orbit closure for matrix semi-invariants continued

- **Case 2**: A and B are both not in the null cone.
- Special case: $A = (I, A_2, \ldots, A_m)$
- Either $\det(B_1) = 1$ or their orbit closures do not intersect.
- If $\det(B_1) = 1$, then can find $(P, Q) \in \mathrm{SL}_n \times \mathrm{SL}_n$ s.t $PB_1Q^{-1} = I$.
- Replace B by $\widetilde{B} = (P, Q) \cdot B = (I, \widetilde{B_2}, \ldots, \widetilde{B_m})$.
- Suffices to check if orbit closures of $(A_2, \ldots, A_m)$ and $(\widetilde{B_2}, \ldots, \widetilde{B_m})$ intersect for the simultaneous conjugation action!

  Let's leave that as an exercise!

Orbit closure for matrix semi-invariants continued

- Special case 2: $\det(c_1A_1 + c_2A_2 + \cdots + c_mA_m) = 1$ for some $c_i \in K$.

Orbit closure for matrix semi-invariants continued

- Special case 2: $\det(c_1A_1 + c_2A_2 + \cdots + c_mA_m) = 1$ for some $c_i \in K$.
- There is an action of GL_m on m-tuples of matrices. For $H = \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mm} \end{pmatrix} \in \mathrm{GL}_m$, we have

$$H \cdot (X_1, \ldots, X_m) = (\sum_i h_{1i}X_i, \sum_i h_{2i}X_i, \ldots, \sum_i h_{mi}X_i).$$

Orbit closure for matrix semi-invariants continued

- Special case 2: $\det(c_1A_1 + c_2A_2 + \cdots + c_mA_m) = 1$ for some $c_i \in K$.
- There is an action of GL_m on m-tuples of matrices. For $H = \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mm} \end{pmatrix} \in \mathrm{GL}_m$, we have

$$H \cdot (X_1, \ldots, X_m) = (\sum_i h_{1i}X_i, \sum_i h_{2i}X_i, \ldots, \sum_i h_{mi}X_i).$$

- This GL_m action commutes with the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$.

Orbit closure for matrix semi-invariants continued

- Special case 2: $\det(c_1A_1 + c_2A_2 + \cdots + c_mA_m) = 1$ for some $c_i \in K$.
- There is an action of GL_m on m-tuples of matrices. For $H = \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mm} \end{pmatrix} \in \mathrm{GL}_m$, we have

$$H \cdot (X_1, \ldots, X_m) = (\sum_i h_{1i}X_i, \sum_i h_{2i}X_i, \ldots, \sum_i h_{mi}X_i).$$

- This GL_m action commutes with the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$.
- Now, using a twist by this action you land in the previous special case.

Orbit closure for matrix semi-invariants continued

- Special case 2: $\det(c_1A_1 + c_2A_2 + \cdots + c_mA_m) = 1$ for some $c_i \in K$.
- There is an action of GL_m on m-tuples of matrices. For $H = \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mm} \end{pmatrix} \in \mathrm{GL}_m$, we have

$$H \cdot (X_1, \ldots, X_m) = (\sum_i h_{1i}X_i, \sum_i h_{2i}X_i, \ldots, \sum_i h_{mi}X_i).$$

- This GL_m action commutes with the left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$.
- Now, using a twist by this action you land in the previous special case.
- **Upshot:** Suffices to have an invertible matrix in the span of the A_i's.

Orbit closure for matrix semi-invariants continued

Summary so far:

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.
2. If we can find an invertible matrix in the span of the A_i's, then we can detect orbit closure intersection.

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.
2. If we can find an invertible matrix in the span of the A_i's, then we can detect orbit closure intersection.

Two issues:

1. Even if such an invertible matrix exists, how do you find one efficiently?

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.
2. If we can find an invertible matrix in the span of the A_i's, then we can detect orbit closure intersection.

Two issues:

1. Even if such an invertible matrix exists, how do you find one efficiently?
2. Such an invertible matrix doesn't always exist.

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.
2. If we can find an invertible matrix in the span of the A_i's, then we can detect orbit closure intersection.

Two issues:

1. Even if such an invertible matrix exists, how do you find one efficiently?
2. Such an invertible matrix doesn't always exist.

Key idea: You can find an invertible matrix in a blow-up efficiently!

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.
2. If we can find an invertible matrix in the span of the A_i's, then we can detect orbit closure intersection.

Two issues:

1. Even if such an invertible matrix exists, how do you find one efficiently?
2. Such an invertible matrix doesn't always exist.

Key idea: You can find an invertible matrix in a blow-up efficiently!

- **Derksen, Makam:** Get a bound of degree $2n^{3.5}$ for separating invariants (vs degree bound of n^6).

Orbit closure for matrix semi-invariants continued

Summary so far:

1. If A or B are in the null cone, IQS algorithm suffices.
2. If we can find an invertible matrix in the span of the A_i's, then we can detect orbit closure intersection.

Two issues:

1. Even if such an invertible matrix exists, how do you find one efficiently?
2. Such an invertible matrix doesn't always exist.

Key idea: You can find an invertible matrix in a blow-up efficiently!

- **Derksen, Makam:** Get a bound of degree $2n^{3.5}$ for separating invariants (vs degree bound of n^6).
- A very different (analytic) algorithm in Yuanzhi Li's talk tomorrow!

Recap and further directions

- Polynomial degree bounds for matrix invariants and matrix semi-invariants.

Recap and further directions

- Polynomial degree bounds for matrix invariants and matrix semi-invariants.
- Immediately give randomized algorithms for null cone and orbit closure problem.

Recap and further directions

- Polynomial degree bounds for matrix invariants and matrix semi-invariants.
- Immediately give randomized algorithms for null cone and orbit closure problem.
- Deterministic algorithms for null cone and orbit closure problem.

Recap and further directions

- Polynomial degree bounds for matrix invariants and matrix semi-invariants.
- Immediately give randomized algorithms for null cone and orbit closure problem.
- Deterministic algorithms for null cone and orbit closure problem.
- Two immediate further directions

Recap and further directions

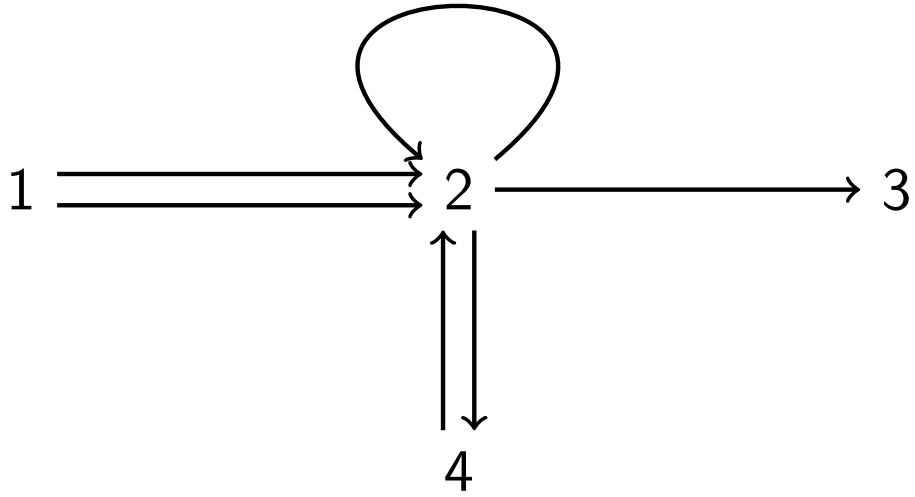
- Polynomial degree bounds for matrix invariants and matrix semi-invariants.
- Immediately give randomized algorithms for null cone and orbit closure problem.
- Deterministic algorithms for null cone and orbit closure problem.
- Two immediate further directions
  1. Generalize to quivers.

Recap and further directions

- Polynomial degree bounds for matrix invariants and matrix semi-invariants.
- Immediately give randomized algorithms for null cone and orbit closure problem.
- Deterministic algorithms for null cone and orbit closure problem.
- Two immediate further directions
  1. Generalize to quivers.
  2. Positive characteristic.

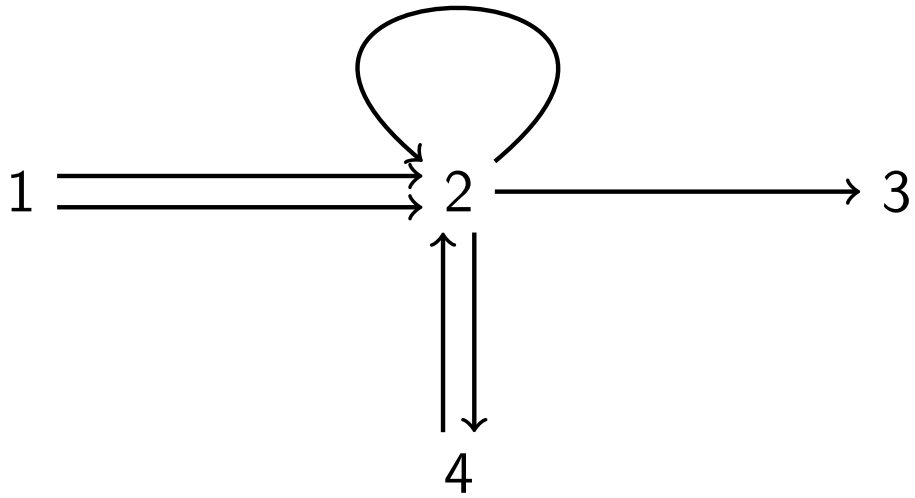
Quivers

- A quiver is a directed graph.



Quivers

- A quiver is a directed graph.



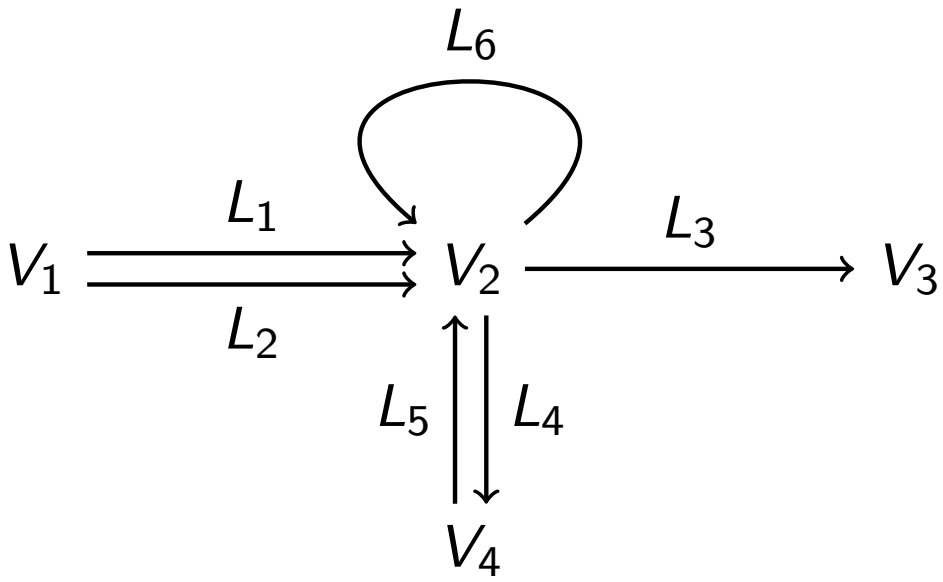
- In general can have directed cycles and multiple edges.

Quiver representation

- A quiver representation is a the data of a vector space at each vertex, and linear maps for each arrow.

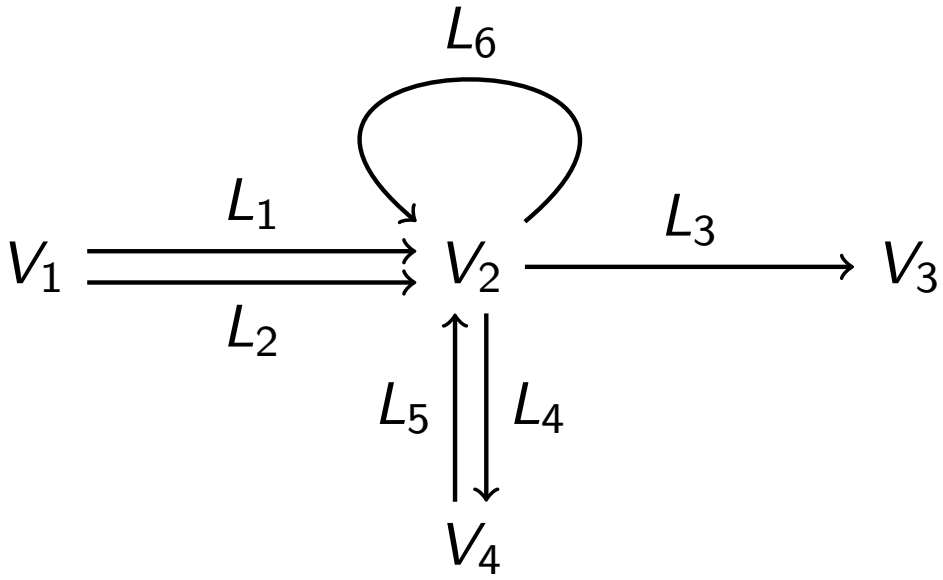
Quiver representation

- A quiver representation is a the data of a vector space at each vertex, and linear maps for each arrow.



Quiver representation

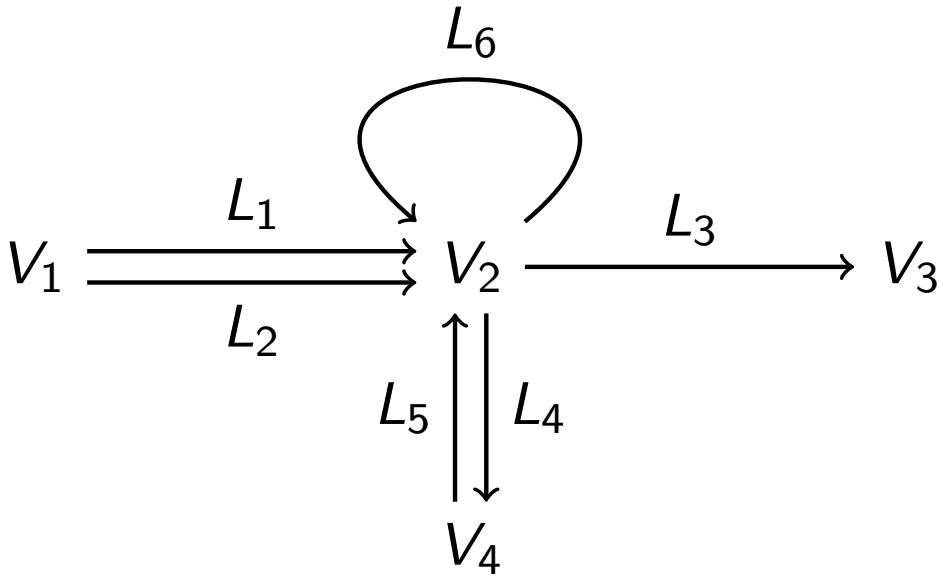
- A quiver representation is a the data of a vector space at each vertex, and linear maps for each arrow.



- Dimension vector is $(\dim(V_1), \dim(V_2), \dim(V_3), \dim(V_4))$.

Quiver representation

- A quiver representation is a the data of a vector space at each vertex, and linear maps for each arrow.



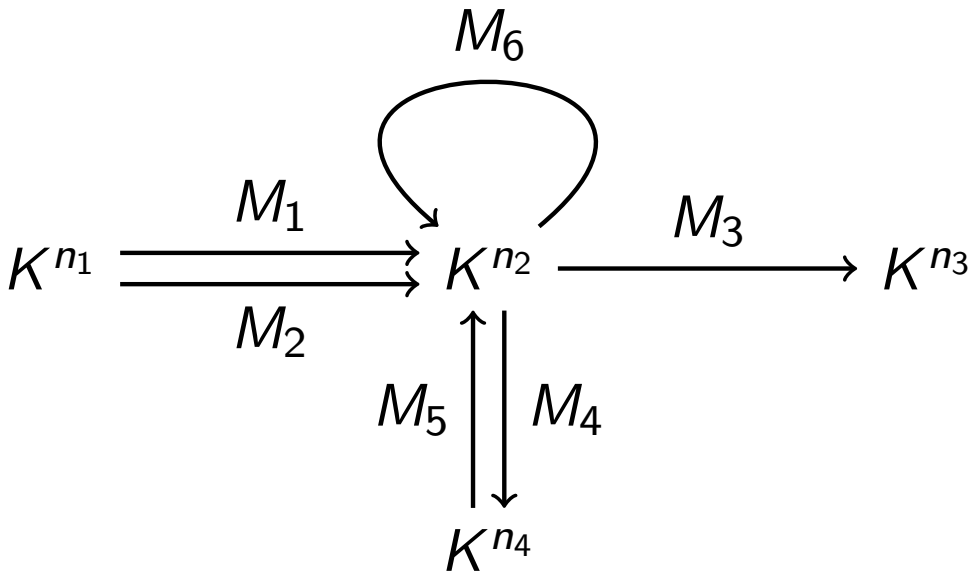
- Dimension vector is $(\dim(V_1), \dim(V_2), \dim(V_3), \dim(V_4))$.
- Might as well choose the vector spaces to be $K^{\dim(V_i)}$

Quiver representation: Take 2

- Fix standard vector spaces at the vertices.

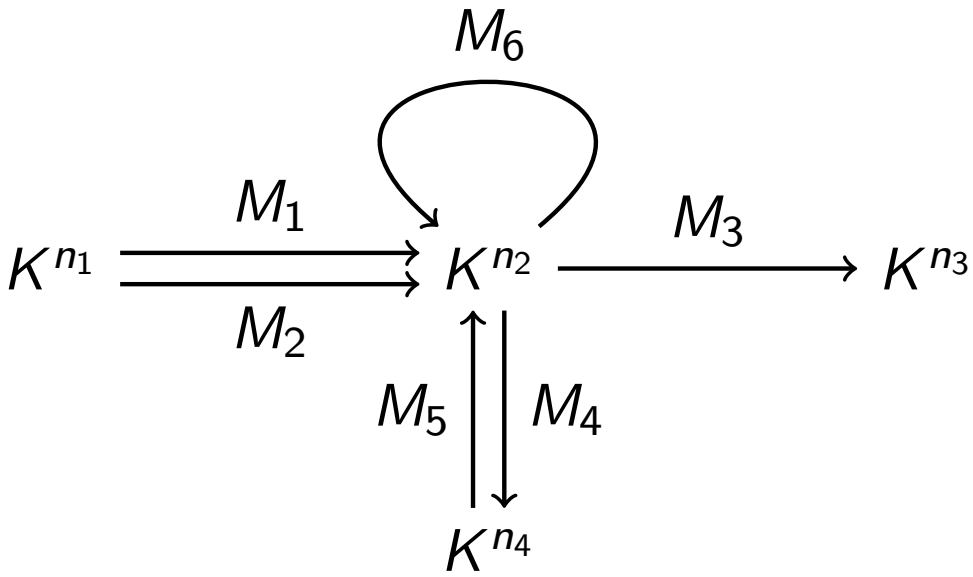
Quiver representation: Take 2

- Fix standard vector spaces at the vertices.



Quiver representation: Take 2

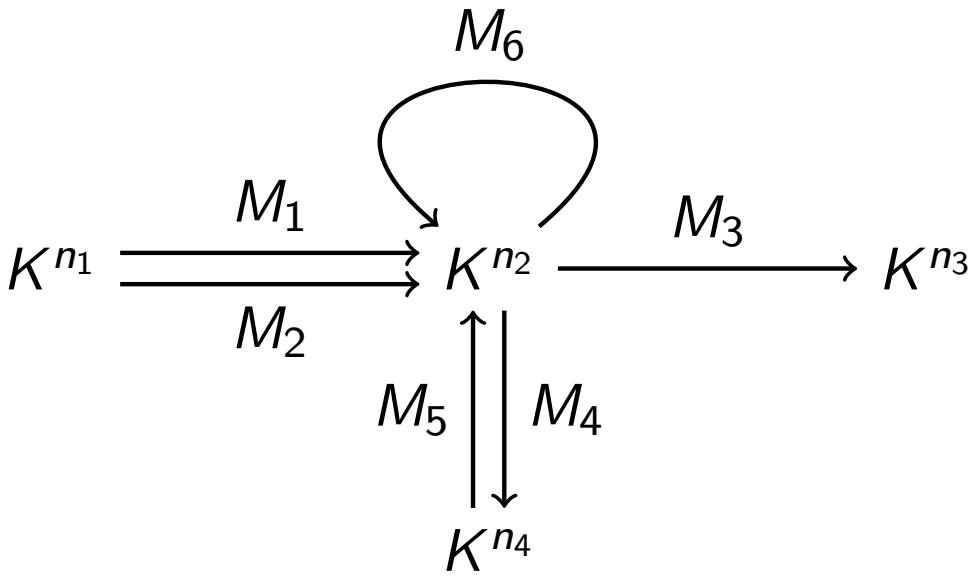
- Fix standard vector spaces at the vertices.



- M_i's are now matrices of the appropriate size!

Quiver representation: Take 2

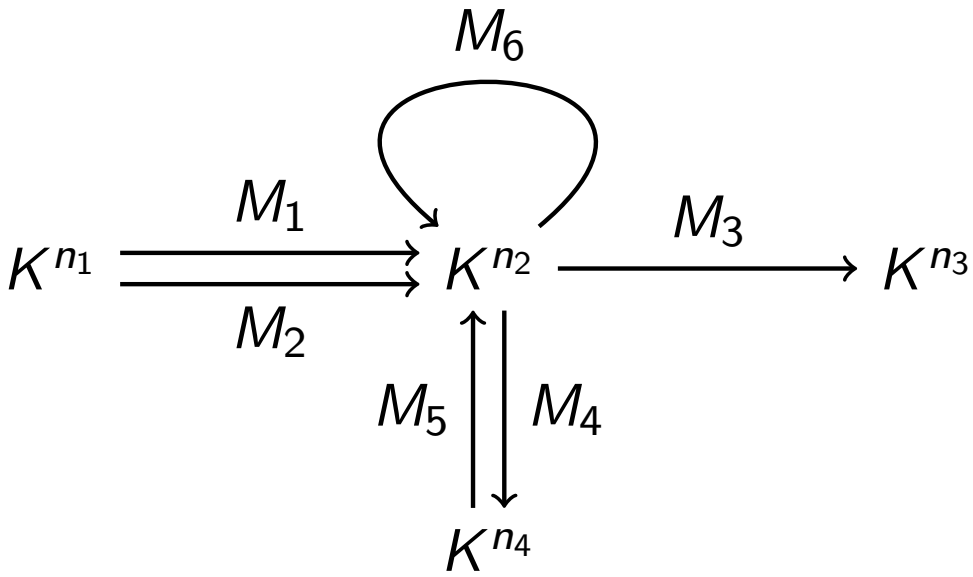
- Fix standard vector spaces at the vertices.



- M_i's are now matrices of the appropriate size!
- When are two representations isomorphic?

Quiver representation: Take 2

- Fix standard vector spaces at the vertices.



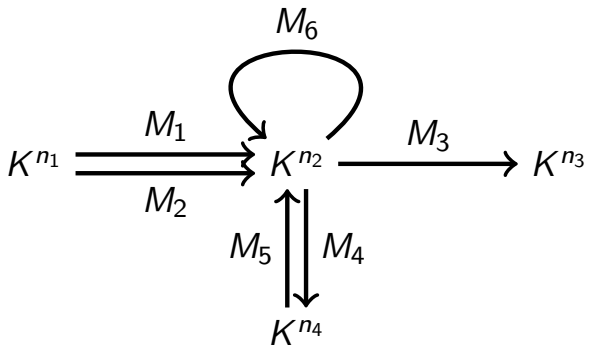
- M_i's are now matrices of the appropriate size!
- When are two representations isomorphic?
- Precisely when they are related by base changes (at the vertices).

Quiver representation: Take 2

- Fix a dimension vector $\underline{n} = (n_1, n_2, n_3, n_4)$.

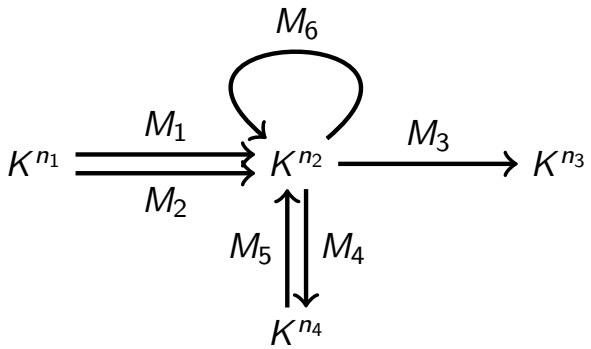
Quiver representation: Take 2

- Fix a dimension vector $\underline{n} = (n_1, n_2, n_3, n_4)$.



Quiver representation: Take 2

- Fix a dimension vector $\underline{n} = (n_1, n_2, n_3, n_4)$.

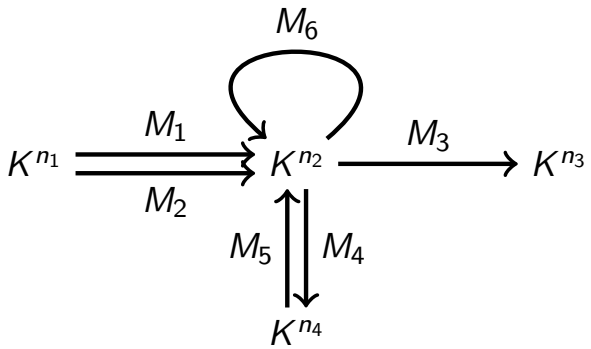


- An $\underline{n}$-dimensional representation is given by a point in

$$\mathrm{Rep}(Q, \underline{n}) = \mathrm{Mat}_{n_2,n_1}^{\oplus 2} \oplus \mathrm{Mat}_{n_3,n_2} \oplus \mathrm{Mat}_{n_4,n_2} \oplus \mathrm{Mat}_{n_2,n_4} \oplus \mathrm{Mat}_{n_2,n_2}$$

Quiver representation: Take 2

- Fix a dimension vector $\underline{n} = (n_1, n_2, n_3, n_4)$.



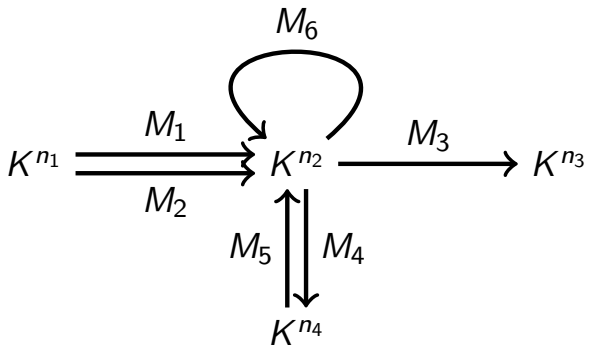
- An $\underline{n}$-dimensional representation is given by a point in

$$\mathrm{Rep}(Q, \underline{n}) = \mathrm{Mat}_{n_2,n_1}^{\oplus 2} \oplus \mathrm{Mat}_{n_3,n_2} \oplus \mathrm{Mat}_{n_4,n_2} \oplus \mathrm{Mat}_{n_2,n_4} \oplus \mathrm{Mat}_{n_2,n_2}$$

- Base change group is $\mathrm{GL}(\underline{n}) = \mathrm{GL}_{n_1} \times \mathrm{GL}_{n_2} \times \mathrm{GL}_{n_3} \times \mathrm{GL}_{n_4}$.

Quiver representation: Take 2

- Fix a dimension vector $\underline{n} = (n_1, n_2, n_3, n_4)$.



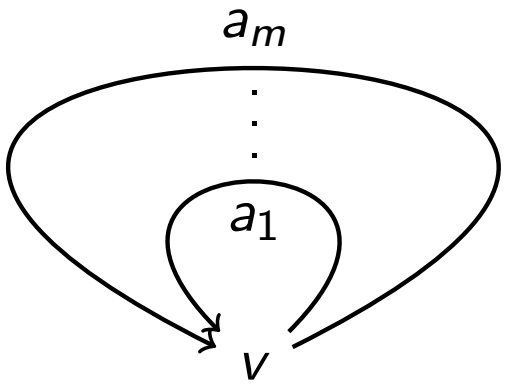
- An $\underline{n}$-dimensional representation is given by a point in

$$\mathrm{Rep}(Q, \underline{n}) = \mathrm{Mat}_{n_2,n_1}^{\oplus 2} \oplus \mathrm{Mat}_{n_3,n_2} \oplus \mathrm{Mat}_{n_4,n_2} \oplus \mathrm{Mat}_{n_2,n_4} \oplus \mathrm{Mat}_{n_2,n_2}$$

- Base change group is $\mathrm{GL}(\underline{n}) = \mathrm{GL}_{n_1} \times \mathrm{GL}_{n_2} \times \mathrm{GL}_{n_3} \times \mathrm{GL}_{n_4}$.
- $\mathrm{Rep}(Q, \underline{n})/\mathrm{GL}(\underline{n})$ = isomorphism classes of $\underline{n}$-dim'l reps.

The loop quiver

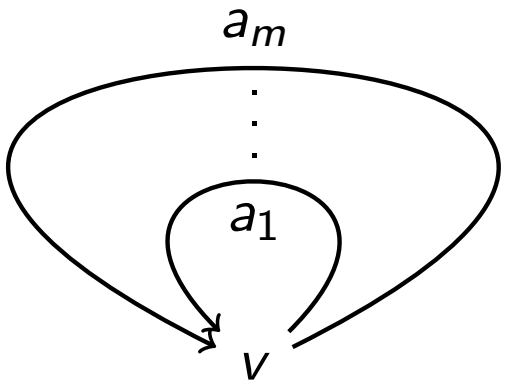
- Consider m-loop quiver:



- If you fix the vector space K^n at the vertex, then a representation is just an m-tuple of $n \times n$ matrices.

The loop quiver

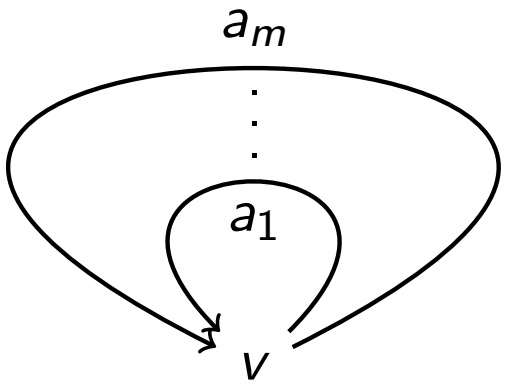
- Consider m-loop quiver:



- If you fix the vector space K^n at the vertex, then a representation is just an m-tuple of $n \times n$ matrices.
- Two representations are isomorphic if there are related by a change of basis of the K^n at the vertex.

The loop quiver

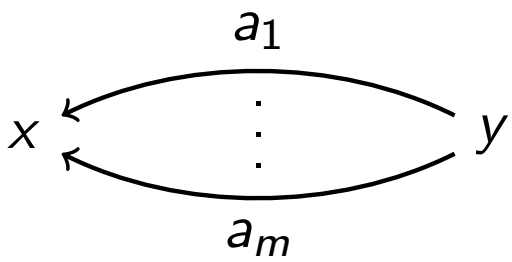
- Consider m-loop quiver:



- If you fix the vector space K^n at the vertex, then a representation is just an m-tuple of $n \times n$ matrices.
- Two representations are isomorphic if there are related by a change of basis of the K^n at the vertex.
- This is precisely the action of GL_n by simultaneous conjugation.

The Kronecker quiver

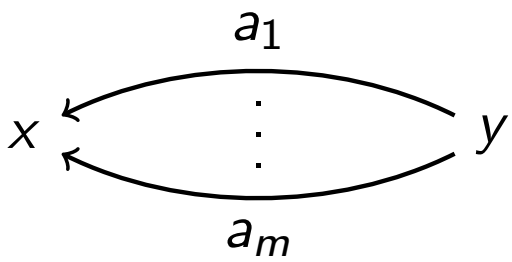
- Consider the m-Kronecker quiver:



- Fixing vector spaces K^p and K^q, a representation is the data of an m-tuple of $p \times q$ matrices.

The Kronecker quiver

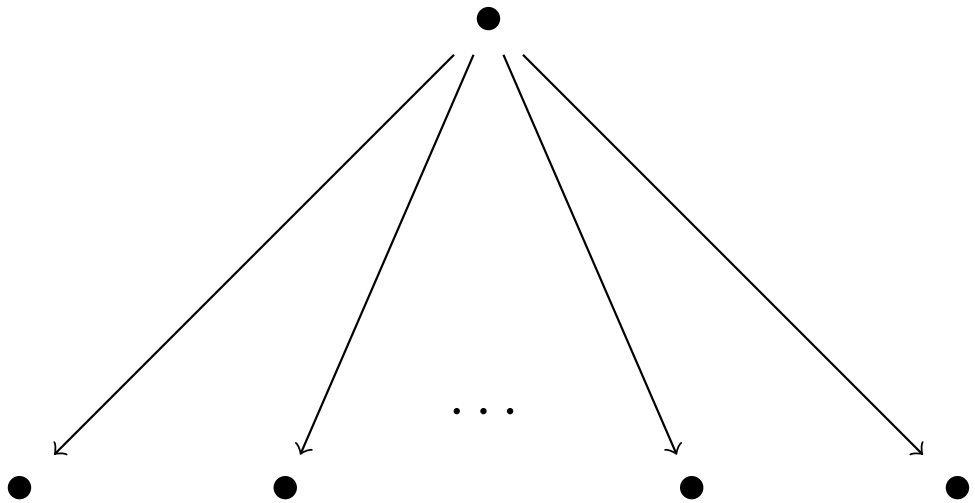
- Consider the m-Kronecker quiver:



- Fixing vector spaces K^p and K^q, a representation is the data of an m-tuple of $p \times q$ matrices.
- Consider the action of $\mathrm{SL}_p \times \mathrm{SL}_q$ via base change, this is the left-right action!

The quiver for Brascamp-Lieb inequalities

The quiver that governs the results on Brascamp-Lieb inequalites is the star quiver



Problems in positive characteristic

- Analytic algorithms cannot be adapted! So, you have to deal with the algebra.

Problems in positive characteristic

- Analytic algorithms cannot be adapted! So, you have to deal with the algebra.
- Description of invariants is harder to get.

Problems in positive characteristic

- Analytic algorithms cannot be adapted! So, you have to deal with the algebra.
- Description of invariants is harder to get.
- Representation theory is more complicated.

Problems in positive characteristic

- Analytic algorithms cannot be adapted! So, you have to deal with the algebra.
- Description of invariants is harder to get.
- Representation theory is more complicated.
- Commutative algebra statements do not always go through.

Example: Matrix invariants

- The traces description misses some invariants.

Example: Matrix invariants

- The traces description misses some invariants.

$$\operatorname{char}(A) = \det(I + tA) = \sum_{i=0}^{n} \sigma_i(A) t^i$$

$$\sigma_1(A) = \operatorname{Tr}(A)$$

$$\sigma_2(A) = \frac{1}{2}(\operatorname{Tr}(A)^2 - \operatorname{Tr}(A^2))$$

$$\sigma_n(A) = \det(A).$$

Example: Matrix invariants

- The traces description misses some invariants.

$$\mathrm{char}(A) = \det(I + tA) = \sum_{i=0}^{n} \sigma_i(A) t^i$$

$$\sigma_1(A) = \mathrm{Tr}(A)$$

$$\sigma_2(A) = \frac{1}{2}(\mathrm{Tr}(A)^2 - \mathrm{Tr}(A^2))$$

$$\sigma_n(A) = \det(A).$$

- σ_i is also an invariant polynomial.

Example: Matrix invariants

- The traces description misses some invariants.

$$\mathrm{char}(A) = \det(I + tA) = \sum_{i=0}^{n} \sigma_i(A) t^i$$

$$\sigma_1(A) = \mathrm{Tr}(A)$$

$$\sigma_2(A) = \frac{1}{2}(\mathrm{Tr}(A)^2 - \mathrm{Tr}(A^2))$$

$$\sigma_n(A) = \det(A).$$

- σ_i is also an invariant polynomial.
- In char p, can't always write σ_i in terms of traces of powers.

Example: Matrix invariants

- The traces description misses some invariants.

$$\mathrm{char}(A) = \det(I + tA) = \sum_{i=0}^{n} \sigma_i(A) t^i$$

$$\sigma_1(A) = \mathrm{Tr}(A)$$

$$\sigma_2(A) = \frac{1}{2}(\mathrm{Tr}(A)^2 - \mathrm{Tr}(A^2))$$

$$\sigma_n(A) = \det(A).$$

- σ_i is also an invariant polynomial.
- In char p, can't always write σ_i in terms of traces of powers.
- **Donkin:** The set $\sigma_i(A_w)$ is a generating set of invariants (highly non-trivial!)

Problem 1: Description of invariants

- How does one get a description of invariants?

Problem 1: Description of invariants

- How does one get a description of invariants?
- **Char** 0 **ans:** Weyl's polarization and restitution method reduces us to understanding multilinear invariants.

Problem 1: Description of invariants

- How does one get a description of invariants?
- **Char** 0 **ans:** Weyl's polarization and restitution method reduces us to understanding multilinear invariants.
- Well understood representation theory $\Longrightarrow$ multilinear invariants easy to understand.

Problem 1: Description of invariants

- How does one get a description of invariants?
- **Char** 0 **ans:** Weyl's polarization and restitution method reduces us to understanding multilinear invariants.
- Well understood representation theory $\Longrightarrow$ multilinear invariants easy to understand.
- Weyl's method doesn't work in positive characteristic.

Problem 1: Description of invariants

- How does one get a description of invariants?
- **Char** 0 **ans:** Weyl's polarization and restitution method reduces us to understanding multilinear invariants.
- Well understood representation theory $\Longrightarrow$ multilinear invariants easy to understand.
- Weyl's method doesn't work in positive characteristic.
- Weyl's method gives traces description for matrix invariants.

Example: A strange phenomenon for the cyclic group in char p

- **Maschke's theorem**: In char 0, any representation of a finite group splits into a direct sum of irreducibles.

Example: A strange phenomenon for the cyclic group in char p

- **Maschke's theorem**: In char 0, any representation of a finite group splits into a direct sum of irreducibles.
- This is because the "averaging" operator allows you to find a "complementary" subrepresentation.

Example: A strange phenomenon for the cyclic group in char p

- **Maschke's theorem**: In char 0, any representation of a finite group splits into a direct sum of irreducibles.
- This is because the "averaging" operator allows you to find a "complementary" subrepresentation.
- Consider the cyclic group $C_p = \{1, a, a^2, \ldots, a^{p-1}\}$.

Example: A strange phenomenon for the cyclic group in char p

- **Maschke's theorem**: In char 0, any representation of a finite group splits into a direct sum of irreducibles.
- This is because the "averaging" operator allows you to find a "complementary" subrepresentation.
- Consider the cyclic group $C_p = \{1, a, a^2, \ldots, a^{p-1}\}$.
- Consider the 2-dimensional representation where a^n acts by $\begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}$.

Example: A strange phenomenon for the cyclic group in char p

- **Maschke's theorem**: In char 0, any representation of a finite group splits into a direct sum of irreducibles.
- This is because the "averaging" operator allows you to find a "complementary" subrepresentation.
- Consider the cyclic group $C_p = \{1, a, a^2, \ldots, a^{p-1}\}$.
- Consider the 2-dimensional representation where a^n acts by $\begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}$.
- $\langle e_1 \rangle$ is a 1-dim'l subrepresentation.

Example: A strange phenomenon for the cyclic group in char p

- **Maschke's theorem**: In char 0, any representation of a finite group splits into a direct sum of irreducibles.
- This is because the "averaging" operator allows you to find a "complementary" subrepresentation.
- Consider the cyclic group $C_p = \{1, a, a^2, \ldots, a^{p-1}\}$.
- Consider the 2-dimensional representation where a^n acts by $\begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}$.
- $\langle e_1 \rangle$ is a 1-dim'l subrepresentation.
- But cannot find a complementary subrepresentation.

Problem 2: Reductive $\neq$ linearly reductive

- In characteristic 0, reductive $\implies$ every representation splits into direct sum of irreducibles (linearly reductive).

Problem 2: Reductive $\neq$ linearly reductive

- In characteristic 0, reductive $\implies$ every representation splits into direct sum of irreducibles (linearly reductive).
- In char p, this is rarely true. Even classical groups (e.g GL_n) don't have this property.

Problem 2: Reductive $\neq$ linearly reductive

- In characteristic 0, reductive $\implies$ every representation splits into direct sum of irreducibles (linearly reductive).
- In char p, this is rarely true. Even classical groups (e.g GL_n) don't have this property.
- Invariant rings for linearly reductive groups have some good commutative algebraic properties:
  1. **Hochster–Roberts:** Cohen-Macaulay
  2. **Kempf:** Hilbert series is a proper rational function.

Problem 2: Reductive $\neq$ linearly reductive

- In characteristic 0, reductive $\implies$ every representation splits into direct sum of irreducibles (linearly reductive).
- In char p, this is rarely true. Even classical groups (e.g GL_n) don't have this property.
- Invariant rings for linearly reductive groups have some good commutative algebraic properties:
  1. **Hochster–Roberts:** Cohen-Macaulay
  2. **Kempf:** Hilbert series is a proper rational function.
- These are not always true for actions of reductive groups.

Problem 2 continued..

- A general bound on null cone in char 0 exists (due to Popov/Derksen), but this doesn't hold in char p.

Problem 2 continued..

- A general bound on null cone in char 0 exists (due to Popov/Derksen), but this doesn't hold in char p.
- Recall Derksen's result:

  Degree bounds for generators = Poly(degree bounds for null cone)

Problem 2 continued..

- A general bound on null cone in char 0 exists (due to Popov/Derksen), but this doesn't hold in char p.
- Recall Derksen's result:

  Degree bounds for generators = Poly(degree bounds for null cone)

- This relies on Hochster–Roberts and Kempf. So, doesn't hold in char p.

Problem 2 continued..

- A general bound on null cone in char 0 exists (due to Popov/Derksen), but this doesn't hold in char p.
- Recall Derksen's result:

  Degree bounds for generators = Poly(degree bounds for null cone)

- This relies on Hochster–Roberts and Kempf. So, doesn't hold in char p.
- Curiously, the bound on null cone for matrix semi-invariants holds in all characteristics!

Matrix semi-invariants

- For matrix semi-invariants, the description of invariants holds in char p as well.

Matrix semi-invariants

- For matrix semi-invariants, the description of invariants holds in char p as well.
- The null cone bound of $n(n-1)$ for matrix semi-invariants remains true in char p. IQS algorithm remains polynomial time.

Matrix semi-invariants

- For matrix semi-invariants, the description of invariants holds in char p as well.
- The null cone bound of $n(n-1)$ for matrix semi-invariants remains true in char p. IQS algorithm remains polynomial time.
- The theory of good filtrations allows us to pass from results in char 0 to char p!

Matrix semi-invariants

- For matrix semi-invariants, the description of invariants holds in char p as well.
- The null cone bound of $n(n-1)$ for matrix semi-invariants remains true in char p. IQS algorithm remains polynomial time.
- The theory of good filtrations allows us to pass from results in char 0 to char p!
- Have to thank Akin-Buchsbaum-Weyman (80's), Donkin (90's) and Hashimoto (00's) for this wonderful and delicate theory.

Results in positive characteristic (Derksen, Makam)

- **Matrix semi-invariants:** A degree bound for generating invariants of mn^4 (and a bound of n^6 if $p > \Omega(n^6)$).

Results in positive characteristic (Derksen, Makam)

- **Matrix semi-invariants:** A degree bound for generating invariants of mn^4 (and a bound of n^6 if $p > \Omega(n^6)$).
- **Matrix invariants:** A degree bound of $(m+1)n^4$ for generating invariants (previous known was $O(n^7 m^n)$ due to Domokos).

Results in positive characteristic (Derksen, Makam)

- **Matrix semi-invariants:** A degree bound for generating invariants of mn^4 (and a bound of n^6 if $p > \Omega(n^6)$).
- **Matrix invariants:** A degree bound of $(m+1)n^4$ for generating invariants (previous known was $O(n^7 m^n)$ due to Domokos).
- Our algorithm for orbit closure adapts well, and continues to run in polynomial time!

Further questions

- Can we understand better the invariant ring for the tensor actions?

Further questions

- Can we understand better the invariant ring for the tensor actions?
- What other invariant rings should we care about?

Further questions

- Can we understand better the invariant ring for the tensor actions?
- What other invariant rings should we care about?
- Can we approach PIT from an invariant theoretic perspective?

Further questions

- Can we understand better the invariant ring for the tensor actions?
- What other invariant rings should we care about?
- Can we approach PIT from an invariant theoretic perspective?
- Is the orbit closure problem harder than the null cone problem?

Further questions

- Can we understand better the invariant ring for the tensor actions?
- What other invariant rings should we care about?
- Can we approach PIT from an invariant theoretic perspective?
- Is the orbit closure problem harder than the null cone problem?