Invariant Theory and Computational Complexity

1 Introduction

This is a brief introduction to the central concepts in computational invariant theory and how they are related to fundamental questions in computational complexity. In the past few years, there has been a lot of activity in this area of computational invariant theory as many connections to diverse areas like algebraic circuit complexity, optimization, quantum information theory, polynomial identity testing (PIT) have been discovered. We will first explain the mathematical setup and state the main results which are often used in most of these works. We will then look at the connection to algebraic complexity theory and what the conjectures say. We will proceed to then look at the current state of art, and survey briefly the different approaches researchers have taken to attack this problem and its sub cases. The aim of this exposition is to provide a bird's eye view of the current literature and simultaneously equip the reader with necessary concepts required to delve into the area.

2 Basic Terminology

This section's a collection of the basic terms and results used in invariant theory. [DK] is an excellent reference for this area. Readers familiar with this terminology may skip this.

2.1 Group Representation

For a group G, a *$G-$representation* is a tuple (V, ρ) where V is a vector space and $\rho : G \to GL(V)$ is a group homomorphism. More concretely, if V is n-dimensional, the map ρ assigns to each group element g, a $n \times n$ invertible matrix $\rho(g)$ such that $\rho(gh) = \rho(g)\rho(h)$. This can be generalised to other structures by replacing G by other algebraic structures like Lie algebra, associative algebras etc.

2.2 Polynomial Algebra

Let $V = span_{\mathbb{F}}\{e_i\}$ i.e V is a vector space over $\mathbb{F}$. $\mathbb{F}[V]$ is defined to be the set of all polynomial functions on the vector space V. One natural way to do this is by declaring a basis of $V = \{v_1, \cdots, v_n\}$. The basis of the dual space, (also called coordinate functions) are $\{x_1, \cdots x_n\}$ where $x_i(v_j) = \delta_{ij}$. Then, $\mathbb{F}[V] \cong \mathbb{F}[x_1, \cdots, x_n]$.

Example - We usually need the vectorspace of d-tuples of $n \times n$ matrices over $\mathbb{F}$ i.e., $V = M(n, \mathbb{F})^{\oplus d}$ which has a natural basis, $\{e_{ij}^k \mid i, j \in [n]\, k \in [d]\}$ which are the elementary matrices. Their dual basically give us the $(i, j)^{th}$ co-ordinate and thus any polynomial in the entries like $tr(A), det(A)$ is an element of $\mathbb{F}[V]$

However, this construction is not basis-free, so, let's look at how to do it more abstractly. The following $S(V^*)$ construction is also used frequently in literature, so it is a nice idea to be familiar with.

The goal is to create polynomials on vectors. But what does this even mean? There's just one way to multiply vectors - tensor products. But this is non-commutative! We force commutativity i.e. quotient the tensor product space. Let's see how to get degree 2 polynomials.

Say, V has a basis $\{e_i \mid i \in [n]\}$. (I'm not cheating here but rather using a basis to ease explanation!) Then, $T^2(V) = V \otimes_{\mathbb{F}} V = span_{\mathbb{F}}\{e_i \otimes e_j \mid i, j \in [n]\}$. We want $e_i \otimes e_j = e_j \otimes e_i$ so we quotient by all such relations, $J = span_{\mathbb{F}}(e_j \otimes e_i - e_i \otimes e_j \mid i, j \in [n])$ to get, $S^2(V) = T^2(V)/J = span_{\mathbb{F}}(e_i \otimes e_j \mid i \leqslant j \in [n])$. This is called the symmetric product. However we need functions on V so we need to take $S^2(V^*)$. Thus, any homogeneous degree 2 polynomial in n variables can be written uniquely as $p(x_1, \cdots, x_n) = \sum_{i \leqslant j} a_{ij} x_i x_j \to \sum_{i \leqslant j} a_{ij}(e_i^*) \otimes (e_j^*) \in S^2(V^*)$. We can do this construction for each d and define $S(V^*) = \bigoplus_d S^d(V^*)$ which, by the above observation, gives $\mathbb{F}[V] := S(V^*) \cong \mathbb{F}[x_1, \cdots x_n]$.

Note - As a shorthand, we write $p(v)$ and not $p(e_1, \cdots, e_n)$. For example, in $V = M(n, \mathbb{F})^2$, we write $p(B_1, B_2) = det(B_1 + B_2)$ but this is a polynomial in $2n^2$ and not 2 variables.

2.3 Invariant Ring

For a $G-$representation (V, ρ), a polynomial $p \in \mathbb{F}[V]$, is said to be *invariant* under the action of G if $p(v) = p(\rho(g)(v)) \ \forall g \in G$ Clearly, the set of all invariant polynomials forms a ring denoted as $\mathbb{F}[V]^G$. Hilbert in a landmark result proved that if G is good enough (*reductive*), this ring is finitely generated (as an $\mathbb{F}$- algebra), i.e there is a finite set of polynomials $\{p_i \mid i \in I\}$ such that any other invariant polynomial can be written as a polynomial in these. Since the set is finite, define $\beta(\mathbb{F}[V]^G) = \max_{i \in I} deg(p_i)$. Define $\mathbb{F}[V]^G_{>0} = \{f \in \mathbb{F}[V]^G \mid f(\mathbf{0}) = 0\}$ which is the set of polynomials with no constant term.

2.4 Nullcone and OCI

Definition 2.1 (Nullcone). $\mathcal{N}(V, G) = \{v \mid f(v) = 0 \ \forall f \in \mathbb{F}[V]^G_{>0}\}$.

Definition 2.2 (Orbit). $G.v = \{g \cdot v \mid \forall g \in G\}$.

Definition 2.3 (Orbit Closure). $\overline{G.v} = \{w \mid \forall f \in \mathbb{F}[V], f(x) = 0 \forall x \in G.v, \implies f(w) = 0\}$.

Example - If $V = \mathbb{R}$ and if S is any infinite set then $\overline{S} = \mathbb{R}$ as any polynomial that vanishes on an infinite set is 0. Conversely, any finite set is already closed as we can construct a polynomial ($f(x) = \prod_{s \in S}(x - s)$) that vanishes exactly on those points.

Since, the vector spaces are finite dimensional, we can identify them with $\mathbb{F}^n$ and thus can also view them as an algebraic variety i.e as the zero-set of a bunch of polynomials. The closure above is thus the closure under Zariski topology which just amounts to adding in all the missing roots of the set of defining polynomials. This is a natural closure to take as the nullcone is defined as a variety. However, we also have the usual Euclidean topology and the Euclidean closure would be defined something like, $\overline{G.v} = \{w \mid \exists (g_i)_{i \in \mathbb{N}}, \lim_{i \to \infty} g_i \cdot v = w\}$

An aside - Viewed thus, a V is also referred to as a G-*variety*. This is the approach taken in *geometric representation theory*. Moreover, the groups we generally work with are matrix groups like GL_n or SL_n are *algebraic groups* which means that they also have the structure of an algebraic variety.[1] In such cases, V is called an algebraic *G-variety*. Clearly, the above definitions of nullcone and orbit closures arise from the geometric viewpoint.

Lemma 2.4. *Euclidean closure is contained in the Zariski closure*

Proof. If $\exists (g_i)_{i \in \mathbb{N}},\ \lim_{i\to\infty} g_i \cdot v = w$. Then as every polynomial is a continuous function, we can apply it inside the limit. So for any invariant p_k, $\lim_{i\to\infty} p_k(g_i \cdot v) = p_k(v) = p_k(w)$ where the first equality is due to the fact that p_k is an invariant polynomial. Thus, $p_k(v) = 0 \iff p_k(w) = 0$. Therefore, $w \in \overline{G.v}$ ☐

The converse, in general, doesn't hold but a famous result is that if $\mathbb{F} = \mathbb{C}$, then the closures are same. Moreover there's a foundational result called the *Hilbert-Mumford* criterion which gives a partial "converse" for just the null-cone but for all algebraically closed fields.

Before we mention the theorem, let's restate, the nullcone in terms of orbit closures.

Lemma 2.5. $\mathcal{N}(V, G) = \{v \mid \mathbf{0} \in \overline{G.v}\}$

Proof. Every $f \in \mathbb{F}[V]^G$ is $f_0 + f',\ f' \in \mathbb{F}[V]^G_{>0}$. If $f(v) = 0 \implies f_0 = 0$ because by definition of $\mathcal{N}(V, G)$, $f'(v) = 0$. But then, $f = f'$ and thus, $f(\mathbf{0}) = 0 \implies \mathbf{0} \in \overline{G.v}$.

Now let $\mathbf{0} \in \overline{G.v}$ and for a contradiction assume $f \in \mathbb{F}[V]^G_{>0}$, $f(v) \neq 0$. Define the polynomial $h = (f - f(v))$ $f(g \cdot v) = f(g \cdot v) - f(v) = f(v) - f(v) = 0$ but $h(0) = -f(v) \neq 0$, This contradicts that $\mathbf{0} \in \overline{G.v}$. Thus, $v \in \mathcal{N}(V, G)$. ☐

The above lemma says that if v is in the nullcone, we can conclude that $\mathbf{0}$ is in the (Zariski) orbit closure of v. The Hilbert-Mumford criterion says that in this case we can also say that its in the Euclidean closure. This is a stronger statement as we saw above that this closure is a smaller set. Moreover, the *witness* sequence comes from a *1-parameter subgroup*.

Theorem 2.6 (Hilbert-Mumford). $v \in \mathcal{N}(V, G) \iff \lim_{t\to 0} \phi(t) \cdot v = 0$ *where* $\phi : \mathbb{F}^\times \to G$ *is a homomorphism. The image of* ϕ *is called the 1-parameter subgroup.*

Now we have a natural computational question.

Definition 2.7 (Nullcone membership (NC))**.** Given a representation (G, V, ρ) and $v \in V$, decide if $v \in \mathcal{N}(V, G)$. If not, try to give a *witness* f, $f(v) \neq 0$. This is called the separating invariant.

We can generalize the problem as follows

Definition 2.8 (Orbit Closure Intersection (OCI))**.** Given a representation (G, V, ρ) and $v, w \in V$, decide if $\overline{G.v} \cap \overline{G.w} \neq \phi$. If not, give a *witness* f, such that $\forall g,\ 0 = f(g.v) \neq f(w)$.

To check that this is indeed a generalization, note that for $w = \mathbf{0}$. $\overline{G.\mathbf{0}} = \{\mathbf{0}\}$ and thus we recover the nullcone membership question.

[1] To see this, SL_n is defined as those matrices A where the polynomial $det(A) - 1 = 0$. To define GL_n, we introduce a new formal variable Y and say $\mathrm{GL}_n = \{A \mid det(A)Y - 1 = 0\}$, thus forcing $det(A) \neq 0$

3 The GCT-5 generalisation

We know that the set of generators is finite. A naive but simple idea to solve OCI is to simply compute the list of all generators and check for each if they evaluate to the same value on the 2 input points. We don't need generators but having separating invariants is enough for our purposes. To make this formal, we define the following.

Definition 3.1. A set $S \subset \mathbb{F}[V]^G$ is called *separating* if for every pair $v, w \in V$ if $\exists f \in \mathbb{F}[V]^G$ $f(v) \neq f(w)$ then $\exists g \in S,\ g(v) \neq g(w)$. Define $\sigma(\mathbb{F}[v]^G) = \min_S \max_{g \in S} deg(g)$ where the min is over all separating S.

The set of generators is clearly separating and thus $\sigma(\mathbb{F}[v]^G) \leqslant \beta(\mathbb{F}[v]^G)$ but a surprising result is that even if $\mathbb{F}[V]^G$ is not finitely generated, then there may exist a finite separating S. OCI is thus reduced to computing S and evaluating it at all points. This may not be feasible as either $|S|$ or $\sigma(\mathbb{F}[v]^G)$ may be exponential. One of the key idea in [Mul16] is that instead of asking for the exact set of generators (or separating invariants), we ask (all of) them to be encoded into a *succinct* circuit along with additional variables that help us recover the generators. Formally, we require a circuit $C[V, G](\mathbf{x}, v) = \sum_j^N f_j(v) g_j(x_1, \cdots, x_m)$, such that m, N are polynomially bounded, $\{f_j \mid j \in [N]\}$ is separating and the $g_j \in \mathbb{F}[V]^G$ are linearly independent. The paper defines V/G to be *explicit* if this $C[V, G]$ can be computed in polynomial time[2]. This clearly, is harder than OCI as orbit closures of v, w intersect iff $C[v, x] - C[w, x]$ is identically 0. The paper shows that computing $C[V, G] \in EXPSPACE$ unconditionally and in $EXPH$ assuming the Generalized Riemann Hypothesis. The recent work [GSS18] can be used to bring this down to $PSPACE$. Moreover, [Mul16] shows that it is polynomial time computable for certain cases(discussed in next section) and conjectures that it must be so for every reductive G and a rational representation V.

An aside - We have already seen that V is also an algebraic variety. The paper generalizes the *explicitness* criteria (and the results) to an any variety W and asks for a circuit that encodes an S. The separation condition is generalized by an integrality condition which needs that $\mathbb{F}[W]$ is integral over S. This is called the NNL[3] problem for the variety W.

4 Current Status

In full generality, since we can build the circuit of all generators in PSPACE and then solve OCI by a PIT test which can be done in PSPACE and thus the problem $OCI \in PSPACE$. [Mul16] also gives a polynomial time randomized Monte-Carlo algorithm to construct the generators. But this is far from being in P that is conjectured. So why does this conjecture make sense? One piece of evidence is that for a certain set of groups and representations we indeed have polynomial time algorithms (these will be discussed later). But there is a more fundamental reason.

[2]You might wonder what the input size is or even how the input is provided. This is a bit technical for our purposes but the thing is that the representation V of every reductive group G breaks as $V = \oplus_\lambda m_\lambda V_\lambda$

[3]NNL stands for Noether normalization lemma which says that for any W a random S of size $> d$ (d is the dimension) works with high probability but no smaller set does. The computational question then is to derandomize this construction maybe by relaxing the size of S to be $poly(d)$ instead of $d + 1$

4.1 “Morally” in $\cap$

A simple way to get a certificate is by giving a separating polynomial f. The problem is that the f might have large degree and/or large coefficients.

A certificate is intuitively harder but look back at the Hilbert-Mumford criterion and at least for the nullcone membership problem (NC) we have a 1-parameter subgroup. This is expected to be much more succinct as the images are usually (conjugates of) diagonal matrices of the form $diag(z^{a_1}, \cdots, z^{a_n})$. These a_i can be shown to be small and thus amazingly the certificate seems easier to obtain.

If V is over $\mathbb{C}$ or $\mathbb{R}$ then we have another amazing result that seems like an easier way of getting a certificate.

Theorem 4.1 (Kempf-Ness)**.** *Let G be a complex reductive group and let (V, ρ) be a G-representation where V is a complex vector space with an inner product. Define $\mu(v) = \frac{d\|g\cdot v\|}{dg}\Big|_{g=e}$ Then $v \in V$ is not in the null-cone iff $\exists 0 \neq y \in \overline{G \cdot v},\ \mu(y) = 0$*

Such a y, along with a certificate for y being in $\overline{G \cdot v}$, is a certificate. Again, the issue is bounds on its size. But, this seems like a more fruitful approach than bounding size of f and this theorem is crucially used the analytic line of works.

5 Current Approaches

5.1 PIT

Given the links to circuits and PIT as outlined in GCT-5, a natural idea is to try to look at groups and their actions such that solving the NNL for them reduces to PIT for a circuit class which has already been derandomized.(Note that *white-box* derandomization suffices). For this to happen, we must know explicitly what the generators of the invariant ring are and whether these can be computed by a restricted circuit class. Utilizing this idea are the works of [Mul16] and [FS13a]. The details are briefly given below.

1. $G = \mathrm{GL}_n(\text{or } \mathrm{SL}_n),\ V = M(n, \mathbb{F})^r = \{(B_1, \cdots, B_r) \mid B_i \in M(n, \mathbb{F})\}$ and the action is conjugation $M \cdot (B_1, \cdots, B_r) = (MB_1M^{-1}, \cdots, MB_rM^{-1})$ The invariants are generated by trace of matrix powers and these invariants can naively be encoded as an *algebraic branching program* (ABP). ([Mul16, FS13a]) showed that it can in fact be encoded in a restricted circuit class called *read-once oblivious algebraic branching program* (ROABP). PIT for this was (quasi)derandomized by [FS13b] and thus, OCI has a polynomial time algorithm.

2. G is reductive and $\dim(G)$ is a constant and V is any finite dimensional representation. The generality is achieved by using a very general mechanism that holds for all reductive groups (kind of by definition). A short detour follows!

5.1.1 Reynold's Operator

Given a element $v \in V$, there is a projection map to the space of invariant vectors V^G which in the case of finite groups is just averaging i.e $R(v) = \frac{1}{|G|}\sum_{g \in G} g \cdot v$. For compact groups, the summation can be generalized by integrating using a Haar measure. This operator exists in general for *reductive groups* and is called the Reynold's operator. This operator $R_G : V \to V^G$ also induces a map $R_G : \mathbb{F}[V] \to \mathbb{F}[V]^G$. It thus maps polynomials to invariant polynomials. It has the property that is preserves the degree. This means that if we have a degree bound, $\beta(\mathbb{F}[V]^G) \leqslant D$, then we can apply the operator to each monomial of degree $\leqslant D$ and these would generate the invariant ring $\mathbb{F}[V]^G$. Two issues remain which prohibit using these in general. One is the degree bound and the second is actually computing these. There are general procedures to compute these and the most used one is called the *Cayley's omega process*. We won't discuss that and the interested reader is referred to [DK]

When the $dim(G)$ is a constant the known degree bounds become polynomially bounded and [Mul16] shows that the operator can be effectively computed and all of the invariants can be packed into a diagonal depth-3 circuit which was studied and (white-box)derandomized by [RS05, AJS09, Sax08]. Thus, when $dim(G)$ is a constant $OCI \in P$. [BGO+18] also uses this approach but not in the algebraic circuit model. It analyses the case of SL_n and shows that applying the Reynold's operator (judiciously) gives coefficients that are not too large (i.e. exponentially large in n).

5.2 Analytic

A line of very interesting work originated with [GGOW15] giving analytic algorithms for these nullcone and orbit closure intersection questions. The group here usually is $\mathrm{SL}_{n_1} \times \mathrm{SL}_{n_2} \cdots \times \mathrm{SL}_{n_d}$ and it acts on $V = \mathbb{C}^{n_0} \otimes \mathbb{C}^{n_1} \otimes \cdots \mathbb{C}^{n_d}$ The case of $d = 2$ which is equivalent to $M(n_1 \times n_2, \mathbb{F})^{n_0}$ is called *operator scaling* as it has origins and application in operator theory and the general one is called tensor scaling. For the case of operator scaling, [GGOW15] gave a polynomial time algorithm for NC and [AGL+18] extended it to OCI. For the general tensor case, [BGO+18] gave a singly exponential algorithm. Other works [GGdOW17] give connections to Brascamp-Lieb inequalities which is a vast generalization of the much-loved AM-GM inequality. Read the beautifully written survey [GdO18] for details on these works.

The algorithms here are of the simple alternating minimization kind which have been used for a long time but the main contribution of these works is to rigorously analyze these using invariant theory tools namely the Hilbert-Mumford criterion and the Kempf-Ness criterion. By their analytic nature, they work only for representations over $\mathbb{C}$ or $\mathbb{R}$. The main idea is as follows

- In every iteration, the input vector v is *scaled* by a simple alternating procedure.
- Associated with v is a progress function called the *capacity* which is some function of its norm. Dually, we can associate another norm-based function ($ds()$) which is related to the moment-map μ.
- We then calculate the decrease in the capacity in each iteration. And thus for any

given ϵ, we know the number of iterations $I(\epsilon)$ for which the algorithm must be run to decrease the capacity to ϵ. This is something like a polynomial in $\log(\frac{1}{\epsilon})$ or $\frac{1}{\epsilon}$

- Clearly, if $v \in \mathcal{N}_G$ then the norm goes to 0. If not try to find a lower bound, ϵ_0, on the capacity.
- Run the scaling algorithm for $I(\epsilon_0)$ steps. If you can decrease the capacity to $\leqslant \epsilon_0$ then $v \in N_G$, else it's not. Dually, we can show that either $ds(v) \geqslant \tau_0$ or it goes to 0. Thus, if $ds(v) < \tau_0$, then $v \notin N_G$
- Therefore, these algorithms can be viewed as minimization optimization procedures over the functions $cap()$ or the $ds()$. These aren't convex but are geodesically convex and [AGL+18] adapts convex optimisation procedures like gradient descent to tackle this problem.

The scaling step is easy and thus always efficient. The 2 main bottlenecks for this are the convergence rate $I(\epsilon)$, which for the operator scaling case is $poly(\log(\frac{1}{\epsilon}))$ but is $poly(\frac{1}{\epsilon})$ for the generalized case of tensors, and the lower bound ϵ_0, τ_0 which is usually (singly) exponentially small.

5.3 Algebraic

While there is no single unifying tool used in these, they use a variety of interesting and surprising algebraic techniques. Apart from the algorithmic papers there are also many works giving lower/upper degree bounds for the generators of the invariant rings, i.e. $\beta(\mathbb{F}[V]^G)$. Let's just list the algorithmic results and briefly discuss their contents.

1. For the left-right action of $G = \mathrm{SL}_n \times \mathrm{SL}_n, V = M(n, \mathbb{F})^r \cong \mathbb{F}^r \otimes \mathbb{F}^n \otimes \mathbb{F}^n$ which is the matrix scaling one that we saw above but for a general field. [DM17, IQS15] gave degree bounds . [IQS17, IQS18] gave a polynomial time algorithm for NC and [DM18] used this as a subroutine to extend it to OCI. The most interesting contribution of the work is a constructive regularity lemma which says the following. Given a tuple of matrices $M_1, \cdots, M_m$ then for every d we define the matrix space $B^d := \{\sum_{i=1}^{l} M_i \otimes B_i \mid B_i \in M(d, \mathbb{F})\}$. Given $A \in B^d$ with rank $> rd$ for some r we can compute in polynomial time $A' \in B^d$ of rank $\geqslant (r+1)d$. Thus, the maximum rank matrix is always a multiple of d. [DM16] gave an alternate proof of the non-constructive, (i.e. that an A' exists but not how to obtain it) version of this. [BJP17, BBJP19] have used ideas from this work to give PTAS for the commutative and algebraic rank.
2. In [DM18], the authors also give an efficient reduction of OCI for the previous action to that of $G = \mathrm{GL}_n\, V =$ acting by conjugation i.e $g \cdot M = gMg^{-1}$. Thus, even this action has a polytime algorithm for OCI. Earlier, for this particular action, [IKS10] had given a polytime algorithm for checking membership in orbit (and not the closure). This action as which we have already seen in PIT also appears in matrix completion problems.
3. For the conjugation action of GL_n (same as above) but on the restricted space of symmetric or skew-symmetric matrices (instead of all matrices), [IQ18] gave a polynomial time-algorithm for checking if the orbits intersect, i.e., $?\exists g\,;(gB_ig^{-1})_i = (C_i)_i$

where B_i, C_i are (skew-)symmetric. They also extend this to decide if given arbitrary matrices $(B_i)_i$, $?\exists g\,;(gB_ig^{-1})_i$ is (skew-)symmetric. This is a very interesting work as it generalises the idea of (skew)symmetry to a $*$-algebra which are algebras with an operation denoted $*$ which is of order 2. It uses the fact that all such *simple* algebras are classified (due to Weyl) and that in the simple case the problem can be restricted for tuples of length 1 which can be easily solved. The algorithm also contains many Lie-algebraic subroutines such as decomposition and radical computation of Lie algebras to basically compute and simplify the $*-$algebra to the *simple* case.

One common advantage of these works is that they work on a large variety of fields, almost all except small fields or certain characteristics, and output a witness i.e. a separating invariant when the orbit closures don't intersect.

6 Conclusion

The reader would(should!) be convinced by now that this area contains a ton of open problems with a wide array of interesting ideas and possible techniques. One one hand, there are general questions like - can we improve the complexity from PSPACE to PH in general?, show the existence of succinct or certificates?, and on the other, there are questions related to analyzing specific representations. Can general ideas emerging in these works like geodesic convex optimization, regularity-like lemmas, degree bounds, be used to solve other problems? Let me just end now with an obvious disclaimer. While I've tried to not omit any major results, guaranteeing comprehensiveness is hard (NP-hard?).

References

[AGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. *CoRR*, abs/1804.01076, 2018. 5.2

[AJS09] Vikraman Arvind, Pushkar S. Joglekar, and Srikanth Srinivasan. Arithmetic circuits and the hadamard product of polynomials. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, pages 25–36, 2009. 5.1.1

[BBJP19] Vishwas Bhargava, Markus Bläser, Gorav Jindal, and Anurag Pandey. A deterministic PTAS for the algebraic rank of bounded degree polynomials. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 647–661, 2019. 1

[BGO+18] Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating Minimization, Scaling Algorithms, and the Null-Cone Problem from Invariant Theory. In Anna R. Karlin, editor, *9th Innovations in*

Theoretical Computer Science Conference (ITCS 2018), volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. 5.1.1, 5.2

[BJP17] Markus Bläser, Gorav Jindal, and Anurag Pandey. Greedy strikes again: A deterministic PTAS for commutative rank of matrix spaces. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 33:1–33:16, 2017. 1

[DK] Harm Derksen and Gregor Kemper. *Computational Invariant Theory*, volume 130 of *Encyclopaedia of Mathematical Sciences*. Springer-Verlag Berlin Heidelberg. 2, 5.1.1

[DM16] Harm Derksen and Visu Makam. On non-commutative rank and tensor rank. *CoRR*, abs/1606.06701, 2016. 1

[DM17] Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. *Advances in Mathematics*, 310:44 – 63, 2017. 1

[DM18] Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *CoRR*, abs/1801.02043, 2018. 1, 2

[FS13a] Michael A. Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. *CoRR*, abs/1303.0084, 2013. 5.1, 1

[FS13b] Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 243–252, 2013. 1

[GdO18] Ankit Garg and Rafael Mendes de Oliveira. Recent progress on scaling algorithms and applications. *CoRR*, abs/1808.09669, 2018. 5.2

[GGdOW17] Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. Algorithmic and optimization aspects of brascamp-lieb inequalities, via operator scaling. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 397–409, 2017. 5.2

[GGOW15] Ankit Garg, Leonid Gurvits, Rafael Mendes De Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. *CoRR*, abs/1511.03730, 2015. 5.2

[GSS18] Zeyu Guo, Nitin Saxena, and Amit Sinhababu. Algebraic dependencies and PSPACE algorithms in approximative complexity. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 10:1–10:21, 2018. 3

[IKS10] Gábor Ivanyos, Marek Karpinski, and Nitin Saxena. Deterministic polynomial time algorithms for matrix completion problems. *SIAM J. Comput.*, 39(8):3736–3751, 2010. 2

[IQ18] Gábor Ivanyos and Youming Qiao. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2357–2376, 2018. 3

[IQS15] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. On generating the ring of matrix semi-invariants. *CoRR*, abs/1508.01554, 2015. 1

[IQS17] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative edmonds' problem and matrix semi-invariants. *Computational Complexity*, 26(3):717–763, 2017. 1

[IQS18] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Computational Complexity*, 27(4):561–593, 2018. 1

[Mul16] Ketan D. Mulmuley. Geometric complexity theory v: Efficient algorithms for noether normalization. *J. Amer. Math. Soc.*, June 2016. 3, 4, 5.1, 1, 5.1.1

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *computational complexity*, 14(1):1–19, Apr 2005. 5.1.1

[Sax08] Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *Automata, Languages and Programming*, pages 60–71, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. 5.1.1