

Polynomial Time Algorithms in Invariant Theory for Torus Actions

Peter Bürgisser ✉

Institut für Mathematik, Technische Universität Berlin, Germany

M. Levent Doğan ✉

Institut für Mathematik, Technische Universität Berlin, Germany

Visu Makam ✉

Institute for Advanced Study, Princeton, NJ, CA

School of Mathematics and Statistics, University of Melbourne, Australia

Michael Walter ✉

Korteweg-de Vries Institute for Mathematics,

Institute for Theoretical Physics, Institute for Logic, Language and Computation,

University of Amsterdam, The Netherlands

Avi Wigderson ✉

Institute for Advanced Study, Princeton, NJ, USA

Abstract

An action of a group on a vector space partitions the latter into a set of orbits. We consider three natural and useful algorithmic “isomorphism” or “classification” problems, namely, *orbit equality*, *orbit closure intersection*, and *orbit closure containment*. These capture and relate to a variety of problems within mathematics, physics and computer science, optimization and statistics. These orbit problems extend the more basic null cone problem, whose algorithmic complexity has seen significant progress in recent years.

In this paper, we initiate a study of these problems by focusing on the actions of commutative groups (namely, tori). We explain how this setting is motivated from questions in algebraic complexity, and is still rich enough to capture interesting combinatorial algorithmic problems. While the structural theory of commutative actions is well understood, no general efficient algorithms were known for the aforementioned problems. Our main results are polynomial time algorithms for all three problems. We also show how to efficiently find separating invariants for orbits, and how to compute systems of generating rational invariants for these actions (in contrast, for polynomial invariants the latter is known to be hard). Our techniques are based on a combination of fundamental results in invariant theory, linear programming, and algorithmic lattice theory.

2012 ACM Subject Classification Computing methodologies → Algebraic algorithms; Theory of computation → Algebraic complexity theory

Keywords and phrases computational invariant theory, geometric complexity theory, orbit closure intersection problem

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.32

Related Version *Full Version*: <https://arxiv.org/abs/2102.07727>

Funding *Peter Bürgisser*: ERC under the European Union’s Horizon 2020 Research and Innovation Programme (grant agreement no. 787840).

M. Levent Doğan: ERC under the European Union’s Horizon 2020 Research and Innovation Programme (grant agreement no. 787840).

Visu Makam: University of Melbourne and NSF grant CCF-1900460.

Michael Walter: NWO Veni grant no. 680-47-459 and NWO grant OCENW.KLEIN.267.

Avi Wigderson: NSF Grant CCF-1900460.



© Peter Bürgisser, M. Levent Doğan, Visu Makam, Michael Walter, and Avi Wigderson;

licensed under Creative Commons License CC-BY 4.0

36th Computational Complexity Conference (CCC 2021).

Editor: Valentine Kabanets; Article No. 32; pp. 32:1–32:30



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Consider the following two problems, which on the face of it have nothing to do with each other:

1. Will the cue ball's trajectory on a billiards table ever end up in a pocket?
2. Given a bipartite graph G , and two functions w, w' assigning weights to edges, is it the case that they assign the same weight to *every* perfect matching M of G ?

Both turn out to be orbit problems for torus actions, and exemplify the class of problems we study in this paper.

As our introduction is somewhat long, we break it up as follows. We start with general background to algorithmic invariant theory in §1.1 and discuss general orbit problems in §1.2. In §1.3 we define torus actions, discuss our main results, and explain their motivation from the perspective of algebraic complexity. In §1.4, we give examples of how these orbit problems for torus actions arise in and capture natural problems in physics and optimization. In §1.5, we discuss the organization of the paper and logical structure of our results.

1.1 Algorithms in invariant theory

Computational invariant theory is a subject whose origins can be traced back to “masters of computation” in the 19th century such as Boole, Gordan, Sylvester and Cayley among others. The second half of the 20th century injected a major impetus to both structural and computational aspects of these mathematical areas. On the one hand, the advent of digital computers allowed mathematicians means to study much larger such algebraic structures than could be accessed by hand. On the other, the parallel development of computational complexity provided a mathematical theory with precise computational models for algorithms and their efficiency analysis. This combination has injected many new ideas and questions into invariant theory and related fields, leading to the development of algorithmic techniques such as Gröbner bases and many others, which supported faster and faster algorithms. Texts on this large body of work can be found, for example, in the books [17, 60, 14]. While the computational complexity put focus on polynomial time as the staple of efficiency, it also provided means to argue the likely impossibility of such fast algorithms for certain tasks, through the Cook-Karp-Levin theory [13, 44, 48] of NP-completeness (for Boolean computation) and Valiant's theory of VNP-completeness.

More recently, a further surge in collaboration between algebraists and complexity theorists on these algorithmic questions in invariant theory and representation theory arose from two (related) sources starting in the turn of this century. Both imply that these very algorithmic questions in algebra are deeply entwined with the core complexity questions of P vs. NP and VP vs. VNP. Not surprisingly, new enriching connections between these two research directions are newly found as they develop, providing an exciting collaboration.

The first source is Mulmuley and Sohoni's Geometric Complexity Theory (GCT) [53], which highlights the inherent symmetries of complete problems of these complexity classes, and through these suggests concrete invariant theoretic and representation theoretic attacks on the questions above. This has led to many new questions, techniques, and much faster algorithms (see, for example, [52, 26, 7, 51]).

The second source is the work of Impagliazzo and Kabanets [42], using Valiant's completeness theory for VP and VNP to again attack these major complexity problems directly through the development of efficient deterministic algorithms for the basic PIT (Polynomial Identity Testing) problem. This problem, which (again, thanks to Valiant's completeness) has natural symmetries, is very similar to basic invariant theory problems. Major progress was

recently made on resolving such related algorithmic problems, starting with [33, 27, 40, 41, 19]. Many others continue to follow, see, for example, [21, 1, 28, 10, 9, 11, 8]. We refer to [8] for a recent description of the state-of-art.

1.2 Orbit problems

We now briefly describe the basic setting and problems of interest, postponing some of the technical details to later sections for the sake of brevity. A group homomorphism $\rho: G \rightarrow \text{GL}(V)$, where V is a vector space (always complex and finite-dimensional) is called a representation of G . One can think of this as a (linear) action of G on V , i.e., a map $G \times V \rightarrow V$ where $(g, v) \mapsto \rho(g)v$ satisfies the usual axioms of a group action. For us, groups will always be algebraic and representations rational, that is, morphisms of algebraic groups. We will denote $\rho(g)v$ by gv or $g \cdot v$.

For $v \in V$, we define its *orbit* $O_v := \{gv \mid g \in G\}$ (denoted $O_{G,v}$ if the group is not clear from context) to be the subset of points that can be reached from v by applying a group element. We denote by $\overline{O_v}$ the topological closure of O_v . These notions are extremely basic and in many concrete instances very familiar. One simple example is the action of $\text{GL}_n \times \text{GL}_n$ on $n \times n$ matrices by left and right multiplication: clearly, the orbit of a matrix A consists of the matrices having the same rank as A ; moreover, the orbit closure of A is the set of matrices whose rank is at most the rank of A . Another example is the conjugation action of GL_n on $n \times n$ matrices, where the orbits are characterized by Jordan normal forms.¹

Understanding the space of orbits of a given group action is perhaps the most basic task of invariant theory. The following three basic algorithmic problems will be the focus of this paper.

► **Problem 1.1.** Let $\rho: G \rightarrow \text{GL}(V)$ be a representation of a group G . Given $v, w \in V$:

1. **Orbit equality:** Decide if $O_v = O_w$;
2. **Orbit closure intersection:** Decide if $\overline{O_v} \cap \overline{O_w} \neq \emptyset$;
3. **Orbit closure containment:** Decide if $w \in \overline{O_v}$.²

As we will discuss the computational complexity of algorithms for these problems, one needs to specify how inputs are given and how we measure their size. We will discuss this, but for now it suffices to think of $n = \dim(V)$, the degree of ρ (assuming it is a polynomial function), and the bit-length of the input vectors v, w as the key size parameters.

The aforementioned problems capture and are related to a natural class of “isomorphism” or “classification” problems across many domains in mathematics, physics and computer science. Examples include the graph isomorphism problem [16], non-commutative rational identity testing [27, 41], equivalence problems on quiver representations [18, 20], matrix and tensor scaling [10, 9], classification of quantum states [4] and module isomorphism problems [6].

To briefly hint at the role of invariant theory, let us take a closer look at problem (2), that is, the problem of orbit closure intersection. We denote by $\mathbb{C}[V]$ the ring of polynomial functions on V . A polynomial function f on V is called *invariant* if it is constant along

¹ The orbit closures of two matrices intersect if and only if the matrices have the same eigenvalues (counted with multiplicity).

² The special case of $w = 0$ is called the null cone membership problem. In fact, many of the recent algorithmic advances mentioned above efficiently solve the null cone problem for specific group actions, see [8] and references therein. The motivation of this paper is to extend that understanding to these more general problems.

orbits, i.e., $f(gv) = f(v)$ for all $g \in G$ and $v \in V$. The collection of all invariant polynomials forms a subring $\mathbb{C}[V]^G$, called the *invariant ring*. Since polynomials are continuous, invariant polynomials are constant along orbit closures. In particular, two points v and w are indistinguishable by invariant polynomials when their orbit closures intersect. Amazingly, the converse is also true for a large class of group actions thanks to a result due to Mumford: if the orbit closures of v and w do not intersect, then they can always be distinguished by an invariant polynomial. See Theorem 2.1 for a precise statement.

Mumford's theorem suggests an approach to orbit closure intersection – test if $f(v) = f(w)$ for all invariant polynomials f . For this strategy to be effective, one needs a computational handle on invariant polynomials. Naively there are infinitely many polynomials, but a foundational result of Hilbert helps tackle this issue. A *system of generating polynomial invariants* is a collection of invariant polynomials $\{f_1, \dots, f_r\}$ such that any other invariant polynomial can be written as a polynomial in the f_i 's. In particular, to test for orbit closure intersection it suffices to test whether each of the f_i take the same value on both points. Hilbert showed the existence of a finite system of generating polynomial invariants and also gave an algorithm to produce them [36]. Since then, many improvements on the complexity of such algorithms were developed, but even today this task is, in general, infeasible. One basic obstacle is the very description of such a system of generating invariants, coming both from the size of this set and the degree of each polynomial in it.

Nearly a century later, a (singly) exponential bound (in n) on the degrees of a system of generating polynomial invariants was achieved for a very general class of group actions [15], which is unfortunately the best possible in this generality, see [22]. A singly exponential bound is necessary to capture a polynomial with a poly-sized (in n) arithmetic circuit, but is by no means sufficient.³ Another issue that one has to deal with is the number of invariants in a system of generating polynomial invariants, and it is often the case that there are exponentially many in any system.⁴ This led Mulmuley [52] to suggest the notion of a *succinct circuit* as a way to capture a system of generating polynomial invariants with a view towards using them for orbit closure intersection. Unfortunately, this approach does not seem to be computationally feasible either. See [29] where Mulmuley's conjecture [52, Conjecture 5.3] on the existence of succinct circuits was disproved under natural complexity assumptions. What is perhaps most surprising is that this already happens for a *commutative* group action, namely when G is a torus. Further, an example of a group action was given where any system of generating polynomial invariants must contain a VNP-hard polynomial.

The negative result above seem to suggest that the algorithmic tasks at hand are infeasible, even for torus actions, i.e., groups of the form $(\mathbb{C}^\times)^d$. The main results of our paper show the opposite: *all of them are efficiently solvable for torus actions!*

The main novelty on our approach is using *rational invariants* instead of polynomial invariants. A rational invariant is a quotient of polynomials that is invariant, see Section 1.3 for a precise definition. This is a bit unexpected since Mumford's theorem simply does not extend to rational invariants: it is easy to construct examples where two points whose orbit closures intersect are distinguished by a rational invariant. Yet, for representations of tori, we show that (a certain special collection of) rational invariants can be used (in a delicate way) to capture not just orbit closure intersection, but orbit closure containment and orbit equality as well. Moreover, we show that rational invariants are computationally easy in this case, in stark contrast with the aforementioned hardness results for polynomial invariants [29].

³ For example, the permanent of an $n \times n$ matrix, which has degree n , is believed to require exponential circuit size. This is essentially the content of Valiant's proof that the permanent is complete for the class VNP, combined with the hypothesis that $\text{VNP} \neq \text{VP}$.

⁴ This is already the case for the matrix scaling action discussed in Section 1.4.

Inspired by the connections to the P vs. NP problem, the GCT program makes several predictions in invariant theory. The setting in which most of the predictions and conjectures are formulated is the setting of rational representations of connected reductive groups (which we will define later). Here, we want to point out that among connected reductive groups, the class of commutative groups happen to be precisely tori. Thus, our main results should be viewed as conclusively verifying several predictions of GCT in the commutative case. Moreover, the barrier result on the computational efficiency of polynomial invariants [29] along with our results on rational invariants suggest that a more thorough investigation of rational invariants is needed in the case where the acting group is non-commutative, e.g., SL_n .

1.3 Torus actions and main results

We now discuss the main contributions of our paper in more detail and precision. Our results concern torus actions, so we specialize the discussion of the preceding section and consider a d -dimensional complex torus $T = (\mathbb{C}^\times)^d$ as the acting group G . The group law is just pointwise multiplication, i.e., $(t_1, \dots, t_d) \cdot (s_1, \dots, s_d) = (t_1 s_1, \dots, t_d s_d)$.

Any linear action of a torus can be described by an integer matrix $M \in \text{Mat}_{d,n}(\mathbb{Z})$ called the *weight matrix* (where $\text{Mat}_{d,n}(\mathbb{Z})$ denotes the space of $d \times n$ integer matrices). The representation $\rho_M: T \rightarrow \text{GL}_n(\mathbb{C})$ corresponding to a weight matrix $M = (m_{ij})$ looks as follows:

$$\rho_M(t) = \begin{pmatrix} \prod_{i=1}^d t_i^{m_{i1}} & & & \\ & \ddots & & \\ & & \prod_{i=1}^d t_i^{m_{in}} & \end{pmatrix} \tag{1}$$

Thus any torus action can be viewed as a scaling action, where each coordinate is scaled separately according to a Laurent monomial.⁵ The weight matrix (up to reordering of columns) determines the representation. Despite the simple description of commutative torus actions, they as well capture fundamental notions, and the associated orbits can be quite complex. One example is the matrix scaling problem, where the orbits capture weights of perfect matchings (see Problem 1.7).

In this paper, we will assume that a torus action is given by specifying the weight matrix. Thus the bit-length of the entries of the weight matrix are included in the input size of the problems. Moreover, we will allow complex number inputs. These can be described up to finite precision by elements in the field of Gaussian rationals $\mathbb{Q}(i) = \{s + it \mid s, t \in \mathbb{Q}\}$, which will be encoded in the standard way; see, e.g., [52].⁶ The following theorem captures the main results of our paper.

► **Theorem 1.2.** *Given as input a weight matrix $M \in \text{Mat}_{d,n}(\mathbb{Z})$ as well as vectors $v, w \in \mathbb{Q}(i)^n$, denote by b the maximal bit-length of the entries of v, w , and M . Then we can in time $\text{poly}(d, n, b)$:*

1. *decide whether $O_v = O_w$;*
2. *decide whether $\overline{O_v} \cap \overline{O_w} \neq \emptyset$;*
3. *decide whether $w \in \overline{O_v}$.*

In other words, for rational representations of tori, there are polynomial time algorithms for orbit equality, orbit closure intersection, and orbit closure containment.

⁵ We can also describe this action as follows: Identify $v \in \mathbb{C}^n$ with a Laurent polynomial $\sum_{j=1}^n v_j z_1^{m_{1j}} \cdots z_d^{m_{dj}}$; then the action of T corresponds precisely to rescaling the variables z_1, \dots, z_d [34].

⁶ In fact, our results hold more generally when the elements in $\mathbb{Q}(i)$ are given in a “floating point” format, namely in the form $(s + it)2^p$, with $s, t \in \mathbb{Q}$ and $p \in \mathbb{Z}$ encoded in binary in the standard way. The same is true for input of the form 2^p , with $p \in \mathbb{Q}$ encoded in binary. See Remark 5.5.

We note that the null cone membership problem mentioned earlier, namely Problems 1.1 (2)/(3) when the input vector w is the 0 vector, was known to have a polynomial time algorithm by a simple reduction to linear programming.⁷ There is no known way of doing the same for the orbit problems above, and indeed our theorem above takes an alternative route.

While one might hope for efficient algorithms for Problems 1.1 (1) and (2) in much more general situations than for tori (for general reductive group actions), our efficient algorithm for orbit closure containment is in stark contrast to the known NP-hardness of the general orbit closure containment problem [5]. Our work points to a key difference: namely, for torus group actions, one can use one-parameter subgroups combined with linear programming techniques to reduce orbit closure containment to orbit equality, while this is impossible in this form for general actions. See Section 7 for more details.

A common core underlying all our results is an efficient algorithm for computing invariant Laurent polynomials for torus actions. The key idea is the following. Invariant polynomials for torus actions can be quite complicated. However, suppose that we restrict to vectors of some fixed support, i.e., “nonzero pattern” of the coordinates. This restriction is without loss of generality, since two vectors can only be in the same orbit when their supports coincide. However, it allows us to study a richer class of functions, namely *Laurent polynomials* instead of ordinary polynomials. Allowing for negative exponents makes an important difference: while polynomial invariants naturally form a semigroup, invariant Laurent polynomials form a *lattice*, isomorphic to the integral vectors in the kernel of the weight matrix. Lattices are much better behaved than semigroups, for example they have small bases which can be found efficiently.

Before describing our results, let us define invariant Laurent polynomials more precisely. For a representation $\rho: G \rightarrow \mathrm{GL}(V)$ of a group G , we have an action of G on the polynomial ring $\mathbb{C}[V]$ defined by $(g \cdot f)(v) := f(\rho(g)^{-1}v)$. When $V = \mathbb{C}^n$, we can identify $\mathbb{C}[V] = \mathbb{C}[x_1, \dots, x_n]$ with the polynomial ring in n variables. Now consider the set of vectors with nonzero coordinates in $S \subseteq [n]$:

$$X_S = \{v \in \mathbb{C}^n \mid v_j \neq 0 \text{ if and only if } j \in S\}.$$

The Laurent polynomials in the variables x_j for $j \in S$ form the natural class of functions on X_S (since we can always divide by the nonzero coordinates). Accordingly, we will denote their collection by $\mathbb{C}[X_S]$.⁸ Now, for a torus action of the form (1), the group T acts on any monomial $x^c = x_1^{c_1} \cdots x_n^{c_n}$ by a simple rescaling. Accordingly, we also have an action of T on the algebra of Laurent polynomials $\mathbb{C}[X_S]$. A Laurent polynomial f is called *invariant* if $g \cdot f = f$ for all $g \in G$. Clearly, if f is invariant, then so are all the Laurent monomials occurring in f . The collection of all invariant Laurent polynomials forms the subalgebra $\mathbb{C}[X_S]^G$ of *invariant Laurent polynomials*. A collection of invariant Laurent polynomials f_1, \dots, f_r is called a *system of generating invariant Laurent polynomials in the variables $\{x_j\}_{j \in S}$* if they generate $\mathbb{C}[X_S]^G$ as an algebra. For torus actions, these can always be taken to be *Laurent monomials*, in which case we call them a *system of generating invariant Laurent monomials*. We can then state our key result:

⁷ Namely, a vector v is in the null cone if and only if the convex hull of the weights corresponding to the nonzero coordinates of v does not contain the origin.

⁸ In the language of algebraic geometry, these are the “regular” functions on X_S .

► **Theorem 1.3.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$, and let $S \subseteq [n]$. Assume that the bit-lengths of the entries of M are bounded by b . Then, in $\text{poly}(d, n, b)$ -time, we can construct an arithmetic circuit with division \mathcal{C} whose output gates compute a system of generating invariant Laurent monomials f_1, \dots, f_r in the variables $\{x_j\}_{j \in S}$, where $r \leq n$.*

Here we recall the notion of an *arithmetic circuit with division*, which is a directed acyclic graph as follows. Every node of indegree zero is called an input gate and is labeled by either a variable or a rational (complex) number. Nodes of indegree one and outdegree one are labeled by $^{-1}$ and are called division gates. Nodes of indegree two and outdegree one and is labeled either $+$ or \times ; in the first case it is a sum gate and in the second a product gate. The only other nodes allowed are output gates which have indegree one and outdegree zero. Given an arithmetic circuit with division, it computes a rational function at each output node in the obvious way. The bit size of such an arithmetic circuit is the total number of nodes plus the total bit-length of the specification of all rational numbers computed in *all* of its gates. The notion of (*division free*) *arithmetic circuits* is obtained by disallowing division gates. They compute polynomials in the obvious way.

We emphasize that the number of generators produced by Theorem 1.3 is at most n (in particular, independent of the bit-length b), in stark contrast to the situation for monomial invariants. Moreover, the bit-length of \mathcal{C} is polynomially bounded.

As a consequence of Theorem 1.3, we are also able to construct arithmetic circuits that compute a generating set of *rational invariants*. For a representation $\rho: G \rightarrow \text{GL}(V)$, the action of G on the polynomial ring $\mathbb{C}[V]$ always extends to an action on its field of rational functions, the rational functions $\mathbb{C}(V)$. A rational function $f \in \mathbb{C}(V)$ is called *invariant* if $g \cdot f = f$ for all $g \in G$. The collection of all rational invariants forms the sub-field $\mathbb{C}(V)^G$ of *rational invariants*. A collection of rational invariants $f_1, \dots, f_r \in \mathbb{C}(V)$ is called a *system of generating rational invariants* if they generate $\mathbb{C}(V)^G$ as a field extension of \mathbb{C} . Note that any invariant Laurent polynomial is a rational invariant, but the converse is not necessarily true. Nevertheless:

► **Corollary 1.4.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$. Assume that the bit-lengths of the entries of M are bounded by b . Then, in $\text{poly}(d, n, b)$ -time, we can construct an arithmetic circuit with division \mathcal{C} whose output gates compute a system of generating rational invariants $f_1, \dots, f_r \in \mathbb{C}(x_1, \dots, x_n)^T$, where $r \leq n$.*

This result is in distinct contrast to the impossibility of finding succinct circuits for generating *polynomial* invariants under natural complexity assumptions [29].

Furthermore, we can complement Theorem 1.2 in the following way: if two orbit closures do not intersect, $\overline{O_v} \cap \overline{O_w} = \emptyset$, then we can construct in polynomial time an arithmetic circuit computing a separating invariant monomial that can serve as a “witness” of the non-intersection.

► **Corollary 1.5.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$. Let $v, w \in \mathbb{Q}(i)$ be such that $\overline{O_v} \cap \overline{O_w} = \emptyset$. Assume the bit-lengths of the entries of v, w and M are bounded by b . Then, in $\text{poly}(d, n, b)$ -time, we can construct an arithmetic circuit of bit-length $\text{poly}(d, n, b)$, which computes an invariant monomial f such that $f(v) \neq f(w)$.*

So far, we have discussed orbit problems for complex tori $T = (\mathbb{C}^\times)^d$. It is interesting to ask to which extent our results hold for “compact” tori, which are groups of the form $K = (\mathbb{S}^1)^d$, where $\mathbb{S}^1 = \{z \in \mathbb{C}^\times \mid |z| = 1\}$.⁹ Besides the fundamental algorithmic interest in this

⁹ Note that K is indeed compact, and a subgroup of T . Moreover, any commutative compact connected

setting, such group actions are important in several areas. For example, the time evolution of periodic systems in Hamiltonian mechanics are naturally given by S^1 -actions, and important symmetries in classical and quantum physics are given by compact group actions.

In fact, the results discussed so far can also be used to give an efficient solution for orbit problems for compact tori. Any (continuous) finite-dimensional representation of $(S^1)^d$ extends to a representation of $(\mathbb{C}^\times)^d$, so representations are specified as before by a weight matrix $M \in \text{Mat}_{d,n}(\mathbb{Z})$. Moreover, the compactness implies that orbits $O_{K,v} = \{kv \mid k \in K\}$ are closed and so all three problems mentioned in Problem 1.1 coincide. Therefore, the following corollary solves all three problems for compact tori:

► **Corollary 1.6.** *Let the weight matrix $M \in \text{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$ and put $K = (S^1)^d$. Further, let $v, w \in \mathbb{Q}(i)^n$ and assume that the bit-lengths of the entries of v, w and M are bounded by b . Then, in $\text{poly}(d, n, b)$ -time, we can decide if $O_{K,v} = O_{K,w}$.*

To give additional context to this result, we briefly mention some recent results achieving polynomial time algorithms for orbit closure intersection of *specific* group actions. For the left-right action (of $\text{SL}_n \times \text{SL}_n$ on m -tuples of $n \times n$ matrices), one approach to solving the orbit closure intersection problem is to (approximately) reduce to the orbit equality problem for the maximal compact subgroup (which happens to be $\text{SU}(n) \times \text{SU}(n)$, where $\text{SU}(n)$ denotes the group of $n \times n$ unitary matrices with determinant 1), see[1]. This was achieved by using a geodesic convex optimization algorithm. Given the recent advances in this area (see, e.g., [8] and references therein), it is natural to ask if a similar approach could be useful for general reductive group actions. For torus actions, interestingly, we can also go in the other direction. Namely, our result for the orbit equality problem for the maximal compact subgroup, Corollary 1.6, is derived from our main result for complex tori, i.e., Theorem 1.2. More generally, we observe that for arbitrary reductive group actions, the orbit equality problem for the maximal compact subgroup is always equivalent to an orbit closure intersection (or equality) problem for a related action of the larger group, see Theorem 8.2 for a precise statement.

The results in this paper warrant the investigation of several interesting directions that we leave for future work, some of which we will discuss in Section 9.

1.4 Further motivation and algorithmic applications

As we saw above, orbit problems are related to a great number of applications. Despite significant progress, for general reductive group actions it is still an open problem to design fast algorithms for these problems. Our results fully resolve the situation in the case of torus actions and also show how to overcome barriers that had previously been pointed out in the literature [39, 29]. Apart from its fundamental complexity theoretic interest, there are also several algorithmic applications where torus actions arise naturally. One particular application in [37] shows how one can use torus invariants to simplify a system of differential equations with scaling symmetries. We provide and discuss in more detail some other concrete applications to combinatorial optimization and to dynamical systems, which were already mentioned briefly at the beginning of the introduction.

We first explain a link to combinatorial optimization. Consider edge weights w for the complete bipartite graph on $2n$ labeled vertices (n on each side): the weight $w(e)$ of an edge e is assumed to be a rational number, encoded in binary. We define the weight $w(M)$ of a perfect matching M of G as the sum of the weights of the edges occurring in M .

Lie group is of this form.

► **Problem 1.7.** *Given edge weights w and w' as above, decide whether they assign the same weight to every perfect matching M of G .*

Perhaps surprisingly, this problem can be reformulated as an orbit intersection problem for a torus action (see below). Therefore, Theorem 1.2 implies that Problem 1.7 can be solved in polynomial time. This insight seems far from being obvious!

The relevant torus action here results from matrix scaling, which has been widely studied and has many applications; see [58] and [12] for more recent developments. Consider $\text{ST}_n := \{(t_1, \dots, t_n) \in \mathbb{C}^\times \mid t_1 \cdots t_n = 1\}$, which is isomorphic to the algebraic torus $(\mathbb{C}^\times)^{n-1}$. We let $\text{ST}_n \times \text{ST}_n$ act on $\text{Mat}_n(\mathbb{C})$ by left-right multiplication as follows:

$$((t_1, \dots, t_n), (s_1, \dots, s_n)) \cdot (v_{ij}) := (t_i v_{ij} s_j)_{ij}. \tag{2}$$

Moreover, we shall identify the edge weights w_{ij} , where $i, j \in [n]$, with the matrix $v_w = (2^{w_{ij}}) \in \text{Mat}_n(\mathbb{C})$.¹⁰ Then one can show that the answer to Problem 1.7 is affirmative if and only if the orbit closures of v_w and $v_{w'}$ in $\text{Mat}_n(\mathbb{C})$ intersect. This follows from Mumford’s theorem mentioned earlier, along with the fact that the invariant polynomials for this action are generated by the perfect matchings, namely the monomials $f_\pi = x_{1,\pi(1)} \cdots x_{n,\pi(n)}$ where $\pi \in S_n$ ranges over the permutations [47, Theorem 3]. Indeed, multiplying entries of v_w is the same as summing the corresponding edge weights in the exponent, hence $f_\pi(v_w) = 2^{w(M)}$, where M is the perfect matching defined by the permutation π .

We briefly comment on the 3-dimensional generalization of this action. $\text{ST}_n \times \text{ST}_n \times \text{ST}_n$ acts on 3-tensors in $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \mathbb{C}^n$ in the natural way:

$$((t_1, \dots, t_n), (s_1, \dots, s_n), (u_1, \dots, u_n)) \cdot (v_{ijk}) = (t_i s_j u_k v_{ijk})_{ijk}.$$

In this case, any system of generating polynomial invariants must include the (maximum) 3-dimensional matching monomials $f_{\pi,\tau} = x_{1,\pi(1),\tau(1)} \cdots x_{1,\pi(n),\tau(n)}$ for $\pi, \tau \in S_n$, which led to the barrier result for torus actions in [29]. Of course, in this case there are additional generating invariants, see, e.g., [49]. Our results show that the corresponding orbit problems can nevertheless be solved in polynomial time! Moreover, it is possible to efficiently exhibit *separating* polynomial invariants (whenever they exist) as well as to construct systems of generating invariant *Laurent polynomial* or *rational* invariants.

Our second example concerns a connection to dynamical systems. Consider a (massless) cue ball on a billiard table (assumed to be square to simplify the discussion). We can ask:

► **Problem 1.8.** *If we hit the cue ball at a given angle, will its trajectory end up in a pocket?*

It is well-known, and easy to see, that one can map trajectories on an ordinary billiard with reflecting boundaries to a billiard of twice the size with periodic boundaries, say $(\mathbb{R}/2\pi\mathbb{Z})^2$. The trajectory of the ball depends fundamentally on the angle or slope. If the slope is irrational, then the trajectory will be dense, so the answer to Problem 1.8 is trivially yes. Otherwise, the trajectory will be periodic and the problem is nontrivial. We can model it as an orbit problem as follows. Let the compact torus S^1 act on \mathbb{C}^2 by

$$t \cdot (x, y) := (t^p x, t^q y),$$

where $s = \frac{q}{p}$ is the slope by which we hit the ball. We can identify points (θ, ν) on the periodic billiard with points $(e^{i\theta}, e^{i\nu}) \in \mathbb{C}^2$. In this way, Problem 1.8 reduces to a constant number of orbit equality problems for this action (one for each pocket). While the problem is

¹⁰ As explained in footnote 6, our results also hold for input of this form, where the w_{ij} are specified in binary.

certainly easy to solve by a variety of methods, one can ask analogous questions for billiards in $n > 2$ dimensions and by allowing a d -dimensional hyperplane worth of allowed cue directions. Such generalizations similarly correspond to orbit problems for compact tori $(S^1)^d$ on some \mathbb{C}^n , and they can all be solved in polynomial time by using Corollary 1.6.

1.5 Organization of the paper

In Section 2, we give an introduction to basic results in invariant theory that we will need to establish our results. In Section 3, we focus on tori, their representations, and their invariants. In particular, we will show that the faces of a natural convex polyhedral “Newton cone” are in one-to-one correspondence with the orbits in an orbit closure, which will be an important ingredient later on.

In Section 4, we discuss the definition and computation of suitable *rational* invariants. As mentioned above, our key result is that for fixed support, a small generating set of invariant Laurent monomials can be computed efficiently. This result, which is Theorem 1.3, is at the heart of our algorithms, and also of independent interest. We achieve this using Smith normal forms. As an easy consequence, this also implies that we can efficiently compute a small generating set of rational invariants for a given representation, that is, Corollary 1.4.

In Section 5, we explain how to use the results of the preceding section to solve the orbit equality problem in polynomial time. This establishes part (1) of Theorem 1.2. Here we rely on known results for testing if a given Laurent monomial (of possibly exponential degree) evaluates to the same value on two given vectors, and we present a brief sketch for completeness.

In Sections 6 and 7, we show how to solve the orbit closure intersection and containment problems by reducing them to orbit equality. This establishes parts (2) and (3) of Theorem 1.2. Here we use the polyhedral description of the structure of orbit closures as furnished by the Newton cone. Furthermore, we show that given two points whose orbit closures do not intersect, we can efficiently construct a separating monomial invariant as a “witness”. This proves Corollary 1.5.

In Section 8, we show how to solve the orbit equality problem for compact tori. This establishes Corollary 1.6. We also give, for general reductive groups G , a reduction from orbit equality for a maximally compact subgroup $K \subseteq G$ to orbit equality and orbit closure intersection for G .

In Section 9, we summarize our results and discuss some important open problems and future directions.

Conventions

In this paper, sometimes we work with monomials and sometimes with Laurent monomials. Unless we use the prefix “Laurent”, by a monomial, we mean $\prod_j x_j^{c_j}$ where $c_j \in \mathbb{Z}_{\geq 0}$, i.e., all exponents are non-negative. Whenever exponents are allowed to be negative, we will be careful to specify that it is a Laurent monomial.

2 Preliminaries of invariant theory

We will briefly recall the main results in invariant theory that are relevant for us (see [46, 23, 17, 55] for details). We will take our ground field to be \mathbb{C} , the field of complex numbers, for simplicity. However, much of this theory works for any algebraically closed field. For a (finite-dimensional) vector space V , we denote by $\mathbb{C}[V]$ the ring of polynomial functions on V . For our purposes, if V is the standard vector space \mathbb{C}^n , then $\mathbb{C}[V] = \mathbb{C}[x_1, \dots, x_n]$, the polynomial ring in n variables, where x_i is to be interpreted as the i^{th} coordinate function.

Let G be an algebraic group, i.e., it has the structure of an algebraic variety (not necessarily irreducible) such that the multiplication map $m: G \times G \rightarrow G$ and the inverse map $\iota: G \rightarrow G$ are morphisms of varieties.¹¹ A morphism of algebraic groups $\rho: G \rightarrow \text{GL}(V)$ is called a rational representation of G .¹² We write gv or $g \cdot v$ for $\rho(g)v$. For a point $v \in V$, its orbit O_v (or $O_{G,v}$ when the group is not clear from context) is the set of all points that can be reached from v by the action of an element of the group, i.e.,

$$O_v := \{gv \mid g \in G\}.$$

We denote by $\overline{O_v}$ the closure of the orbit O_v . The closure is to be taken either with respect to the Euclidean topology or the Zariski topology. Indeed, the closures in both topologies coincide, a well-known fact that relies on a fundamental result in algebraic geometry due to Chevalley (see [54, I.§10]). A polynomial function $f \in \mathbb{C}[V]$ is called *invariant* if it is oblivious to the group action, i.e., $f(gv) = f(v)$ for all $g \in G, v \in V$. The collection of all invariant polynomials forms a subring

$$\mathbb{C}[V]^G := \{f \in \mathbb{C}[V] \mid \forall g \in G, v \in V \ f(gv) = f(v)\}.$$

One key observation is that invariant functions are constant along orbits and hence constant along orbit closures as well. Hence, if the orbit closures of two points intersect, then they cannot be distinguished by an invariant function. The converse was proved by Mumford for a special class of groups called reductive groups [55] (see also [17, Corollary 2.3.8]). An algebraic group G is called *reductive* if every rational representation is a direct sum of irreducible representations, wherein a representation is called irreducible if it has no non-trivial subrepresentations. Examples of reductive groups include $\text{SL}_n, \text{GL}_n, \text{Sp}_n, \text{O}_n$, finite groups, and most importantly for us, tori (which we define formally in the next section), as well as direct products thereof.¹³ Reductive groups have played a central role for a number of mathematical fields for over a century. A particularly important result in the invariant theory of reductive groups is that invariant rings are finitely generated [36, 35, 62].

To state Mumford’s result in the generality we need, we will define rational actions on varieties (a notion that naturally generalizes rational representations). Let X be an algebraic variety and let $\mathbb{C}[X]$ denote the ring of regular functions on X . A rational action of an algebraic group G on X is a morphism of varieties $G \times X \rightarrow X, (g, x) \mapsto g \cdot x$ satisfying $g \cdot (g' \cdot x) = (gg') \cdot x$ and $e \cdot x = x$ for all $x \in X, g, g' \in G$. As in the vector space case, we denote the orbit of a vector $v \in X$ by O_v .

► **Theorem 2.1** (Mumford, [55]). *Let G be a reductive group. Let X be an algebraic variety and suppose we have a rational action of G on X . For $v, w \in X$ we have $\overline{O_v} \cap \overline{O_w} = \emptyset$ if and only if there exists $f \in \mathbb{C}[X]^G$ such that $f(v) \neq f(w)$.*

Another well-known important structural result states that every orbit closure $\overline{O_v}$ contains a *unique* closed orbit.

¹¹ A morphism of varieties simply means that in local coordinates the map is given by ratios of polynomials. For concreteness, the reader may simply think of an algebraic group as a matrix group, i.e., a subgroup of $\text{GL}_n(\mathbb{C})$ that is described as the zero locus of a collection of polynomials.

¹² One can interpret this action as the action of the subgroup $\rho(G) \subseteq \text{GL}(V)$ on V by matrix-vector multiplication, where $\rho(G)$ is parametrized algebraically by an algebraic group G .

¹³ The group B_n of upper triangular $n \times n$ invertible matrices is a typical example of a group that is not reductive.

32:12 Polynomial Time Algorithms in Invariant Theory for Torus Actions

► **Theorem 2.2.** *Let $\rho: G \rightarrow \mathrm{GL}(V)$ be a rational representation of a reductive group G . Then:*

1. *For any $v \in V$, the orbit closure $\overline{O_v}$ contains a unique closed orbit, that we denote by $O_{\tilde{v}}$.*
2. *If $v, w \in V$, then*

$$\overline{O_v} \cap \overline{O_w} \neq \emptyset \iff O_{\tilde{v}} = O_{\tilde{w}}.$$

Proof. (1) The first assertion is [17, Theorem 2.3.6].

(2) For the second assertion, if the orbit closures $\overline{O_v}$ and $\overline{O_w}$ are disjoint, then so are the orbits $O_{\tilde{v}}$ and $O_{\tilde{w}}$, which therefore must be different. Conversely, suppose $O_{\tilde{v}} \neq O_{\tilde{w}}$. Since these orbits are closed, by Theorem 2.1, there is an invariant $f \in \mathbb{C}[V]^G$ such that $f(\tilde{v}) \neq f(\tilde{w})$. By continuity, $f(v) = f(\tilde{v}) \neq f(\tilde{w}) = f(w)$, which implies $\overline{O_v} \cap \overline{O_w} = \emptyset$ by another application of Theorem 2.1. ◀

Part(2) of this theorem shows that the orbit closure intersection problem can be reduced to the orbit equality problem, provided we can compute the unique closed orbit $O_{\tilde{v}}$ contained in $\overline{O_v}$. We will see in Section 6 that if the group G is a torus, this can be achieved in polynomial time.

Another key result in understanding orbit closures is the Hilbert–Mumford criterion. A *one-parameter subgroup* of G is a morphism of algebraic groups $\sigma: \mathbb{C}^\times \rightarrow G$. For a representation of G on a vector space V , we say that a subset $S \subseteq V$ is G -stable if $g \cdot s \in S$ for all $g \in G, s \in S$.

► **Theorem 2.3** (Hilbert–Mumford criterion, [35, 55]). *Let $\rho: G \rightarrow \mathrm{GL}(V)$ be a rational representation of a reductive group G . Suppose $S \subseteq V$ is a G -stable closed subvariety of V and let $v \in V$ such that $\overline{O_v} \cap S \neq \emptyset$. Then there exists a one-parameter subgroup $\sigma: \mathbb{C}^\times \rightarrow G$ such that $\lim_{\epsilon \rightarrow 0} \sigma(\epsilon) \cdot v \in S$.*

A particular use of the above theorem is to take $S = \{0\}$ or $S = O_{\tilde{v}}$. When G is a torus, the set of one-parameter subgroups has the structure of a \mathbb{Z} -lattice. We will discuss this further in the next section.

We end this section by introducing a key notion in invariant theory called the *null cone*, whose significance will become clear in later sections. For a collection F of polynomials in $\mathbb{C}[V]$, we denote by $\mathbb{V}(F)$ their common zero locus in V .

► **Definition 2.4** (Null cone). *Let $\rho: G \rightarrow \mathrm{GL}(V)$ be a rational representation of a reductive group G . Then the null cone is defined as*

$$\mathcal{N}_G(V) := \mathcal{N}(\rho) := \{v \in V \mid 0 \in \overline{O_v}\}.$$

It can also be defined as the common zero locus of all invariant polynomials without constant part:

$$\mathcal{N}_G(V) := \mathcal{N}(\rho) := \mathbb{V}\left(\bigcup_{d>0} \mathbb{C}[V]_d^G\right),$$

where $\mathbb{C}[V]_d^G$ denotes the space of invariant polynomials that are homogeneous of degree d . The equivalence of the two definitions of the null cone follows from Theorem 2.1.

3 Invariants and orbit closures of torus actions

Invariant theory for general reductive groups can get very complicated. However, for representations of tori, that is, *commutative* connected reductive groups, a lot of the theory can be viewed as a combination of linear algebra and the study of convex polytopes. We will collect important results regarding torus actions in this section and refer the reader to [61, 17] for more details. All the results in this section are already known or can be deduced from the existing literature, and we provide proof sketches for completeness. Note that tori are reductive groups, so the results of the previous section hold in this setting.

We will first briefly recall torus actions and the notions of characters/weights, one-parameter subgroups and how weight matrices define a representation. Then, we give a linear algebraic description of invariant rings by determining the monomials that are invariant. Then, we describe a polyhedral perspective on orbits. In particular given a point v in the vector space of the representation, we define a polyhedral cone, called the Newton cone. The Newton cone can be used to determine whether v is in the null cone and moreover we give a correspondence between the faces of the Newton cone to orbits in the orbit closure of v , which is crucial in understanding the orbit closure containment problem.

For this entire section, fix a torus $T = (\mathbb{C}^\times)^d$.¹⁴

3.1 Representations and invariants

As described in Section 1.3, any representation of a torus T is a “scaling” action (after identifying V with \mathbb{C}^n by an appropriate choice of basis). Namely, each coordinate of $v \in \mathbb{C}^n$ is multiplied by some (Laurent) monomial $\prod_{i=1}^d t_i^{\lambda_i}$ for integers $\lambda_i \in \mathbb{Z}$. These monomials (succinctly described by the so-called weight matrix, see below) together specify the representation. We now make this more precise.

A 1-dimensional (rational) representation is called a *character* or a *weight*. Let $\mathcal{X}(T)$ denote the set of weights of T , which forms a group where the binary operation is (pointwise) multiplication of functions. To each $\lambda = (\lambda_1, \dots, \lambda_d) \in \mathbb{Z}^d$, we associate a weight, also denoted λ by slight abuse of notation, namely

$$\lambda: T \rightarrow \mathbb{C}^\times, \quad \lambda(t) = \prod_{i=1}^d t_i^{\lambda_i},$$

which gives an identification of abelian groups $\mathbb{Z}^d \cong \mathcal{X}(T)$.

Let $\rho: T \rightarrow \mathrm{GL}(V)$ be a (rational) representation of T where V is an n -dimensional vector space. We can choose a basis of V consisting of weight vectors, wherein a vector $v \in V$ is called a *weight vector* of weight $\lambda \in \mathcal{X}(T)$ if $t \cdot v = \lambda(t)v$ for all $t \in T$. Once we have chosen a weight basis, using the identification $\mathcal{X}(T) \cong \mathbb{Z}^d$, the corresponding n weights can be collected into a $d \times n$ matrix with integer entries, which we call the *weight matrix* of the representation. Up to permutation of the columns, it is independent of the choice of weight basis, and it classifies the representation up to isomorphism. Concretely, a matrix $M = (m_{ij}) \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ describes the representation $\rho_M: T \rightarrow \mathrm{GL}_n(\mathbb{C})$ defined in (1). That is, for $t = (t_1, \dots, t_d)$ and $v = (v_1, \dots, v_n) \in \mathbb{C}^n$, we have

¹⁴Any commutative connected reductive group is isomorphic to some $(\mathbb{C}^\times)^d$. Important examples include T_d , the group of diagonal $d \times d$ invertible matrices and its subgroup ST_d consisting of diagonal matrices with determinant 1.

$$t \cdot v = \rho_M(t)v = \left(\left(\prod_{i=1}^d t_i^{m_{i1}} \right) v_1, \left(\prod_{i=1}^d t_i^{m_{i2}} \right) v_2, \dots, \left(\prod_{i=1}^d t_i^{m_{in}} \right) v_n \right).$$

The matrix M is the weight matrix for this action. The j^{th} standard basis vector e_j is a weight vector of weight $m^{(j)} = (m_{1j}, m_{2j}, \dots, m_{dj}) \in \mathbb{Z}^d = \mathcal{X}(T)$. Note that $m^{(j)}$ is the j^{th} column vector of M .

For the rest of this section, we fix an n -dimensional representation $\rho_M: T \rightarrow \text{GL}_n(\mathbb{C})$ of the torus $T = (\mathbb{C}^\times)^d$ given by a weight matrix $M \in \text{Mat}_{d,n}(\mathbb{Z})$ with columns $m^{(j)}$ for $j \in [n]$. The following well-known result describes the invariant ring of this action (see, e.g., [22, Section 3]):

► **Proposition 3.1.**

1. Let $c \in \mathbb{Z}_{\geq 0}^n$. A monomial $x^c = \prod_j x_j^{c_j}$ is invariant if and only if $\sum_j c_j m^{(j)} = 0$;
2. The invariant ring $\mathbb{C}[x_1, \dots, x_n]^T$ is spanned as a vector space by the invariant monomials.

Proof. For the action ρ of G on V , there is a natural induced action of G on the ring of polynomial functions $\mathbb{C}[V]$ defined by the formula $g \cdot f(v) := f(\rho(g)^{-1}v)$. Applying this for the action ρ_M , we get an induced action of T on $\mathbb{C}[x_1, \dots, x_n]$. It is easy to compute this action: for a monomial x^c and $t \in T$, we have $t \cdot x^c = \lambda(t)^{-1} x^c$, where $\lambda \in \mathcal{X}(T)$ is the character corresponding to $\sum_j c_j m^{(j)} \in \mathbb{Z}^d$. It follows that the monomials which are invariant are precisely the ones for which $\sum_j c_j m^{(j)} = 0$, the trivial character, proving the first part. The second part follows from the observation that a polynomial is invariant if and only if each monomial that occurs in it is invariant. ◀

Part (1) of Proposition 3.1 shows that the invariant monomials are in one-to-one correspondence with the nonnegative integer vectors in the kernel of the weight matrix. Accordingly, they form a semigroup. In general, such semigroups can have a large number of generators, which explains the difficulty of using polynomial invariants [24]. Our key idea to obtain efficient algorithms will be to instead consider invariant Laurent monomials, which form a lattice rather than a semigroup. We will return to this in Section 4.

It turns out that the weights lead to a strong link to convex polyhedral geometry, which in turn characterizes the orbits in an orbit closure. For this, we make the following definitions. The *support* of a vector $v \in \mathbb{C}^n$ is defined as

$$\text{supp}(v) := \{j \in [n] \mid v_j \neq 0\}.$$

Let us record some of the properties of the support. By dimension (of an orbit, orbit closure, algebraic group, etc), we mean the dimension of the underlying variety.

► **Lemma 3.2.** For $v, w \in \mathbb{C}^n$ we have:

1. If $O_v = O_w$, then $\text{supp}(v) = \text{supp}(w)$.
2. If $\text{supp}(v) = \text{supp}(w)$, then $\dim O_v = \dim O_w$.
3. If $w \in \overline{O_v}$, then $\text{supp}(w) \subseteq \text{supp}(v)$. This inclusion is strict if and only if $w \in \overline{O_v} \setminus O_v$.

Proof. (1) is clear, since each coordinate simply gets rescaled by a nonzero number by the group action. For (2) we note that the stabilizer group $\text{stab}(v)$ of v only depends on $\text{supp}(v)$. The claim follows using $\dim O_v = d - \dim \text{stab}(v)$. For (3), the inclusion of supports holds since taking limits can never increase the support. Finally, it is known [38, §8.3] that $\overline{O_v} \setminus O_v$ is a Zariski closed subset of dimension strictly less than $\dim O_v$. Hence $w \in \overline{O_v} \setminus O_v$ implies $\dim O_w < \dim O_v$ and therefore $\text{supp}(w) \subsetneq \text{supp}(v)$ by part (2). ◀

3.2 Newton cone and orbit closures

We define the *Newton cone* $C(v)$ of a vector $v \in \mathbb{C}^n$ to be the rational polyhedral cone generated by the weights corresponding to the indices in the support, that is,

$$C(v) := \left\{ \sum_{j \in \text{supp}(v)} c_j m^{(j)} \mid c_j \geq 0 \right\} \subseteq \mathbb{R}^d.$$

The *lineality space* of the cone $C(v)$ is defined as $L(v) := C(v) \cap (-C(v))$. Clearly, it is the largest linear subspace contained in $C(v)$. The cone $C(v)$ is called *pointed* iff $L(v) = 0$. (Compare [57] for the structure of polyhedral cones.)

These notions are standard in geometric programming, which essentially studies optimization problems associated with torus actions, albeit often with a different representation and motivation; see, e.g., [11] and references therein. The connection is particularly apparent and useful in the study of polynomial capacities which have important applications to approximate counting [50, 34].

We will see that the Newton cone contains all the information about the orbits contained in an orbit closure. To start, we show that membership in the null cone can be characterized as follows. Define the *essential support* of a vector $v \in V$ as

$$\text{e-supp}(v) := \{j \in \text{supp}(v) \mid m^{(j)} \in L(v)\}. \quad (3)$$

► **Lemma 3.3.** *Let $k \in \text{supp}(v)$. We have $k \in \text{e-supp}(v)$ if and only if there exists an invariant monomial $\prod_{j \in \text{supp}(v)} x_j^{c_j}$ with $c_j \in \mathbb{Z}_{\geq 0}$ such that $c_k > 0$.*

Proof. It is easy to see that $m^{(k)} \in L(v)$ if and only if there is a non-negative integral linear combination $\sum_{j \in \text{supp}(v)} c_j m^{(j)} = 0$ with $c_k > 0$. By Proposition 3.1, this is equivalent to the existence of an invariant monomial $\prod_{j \in \text{supp}(v)} x_j^{c_j}$ with $c_j \in \mathbb{Z}_{\geq 0}$ such that $c_k > 0$. ◀

► **Corollary 3.4.** *We have that v is in the null cone $\mathcal{N}(\rho_M)$ if and only if $\text{e-supp}(v) = \emptyset$.*

Equivalently, v is in the null cone if and only if $C(v)$ is pointed and $m^{(j)} \neq 0$ for all $j \in \text{supp}(v)$.

In fact, much more can be said. Let us first recall the notion of faces of polyhedral cones. If $C(v)$ is contained in a closed halfspace H_+ of \mathbb{R}^d bounded by a linear hyperplane H , then we call the intersection $F = H \cap C(v)$ a *face* of $C(v)$ when it is non-empty. The cone itself is also considered a face of $C(v)$: by definition, it is the largest face of $C(v)$. On the other hand, each face of $C(v)$ must contain the lineality space $L(v)$, which is therefore the smallest face of $C(v)$.

We will see shortly that the faces of $C(v)$ are in bijective correspondence with the orbits contained in $\overline{O_v}$. For this, we need to introduce some more notation. For a subset $J \subseteq \text{supp}(v)$, we define the *restriction* $v|_J$ to be the vector with entries

$$(v|_J)_j = \begin{cases} v_j & \text{if } j \in J, \\ 0 & \text{otherwise,} \end{cases}$$

as its j -th coordinate. Let now F be a face of $C(v)$ defined by a closed half-space $H_+ = \{y \in \mathbb{R}^d \mid \nu \cdot y \geq 0\}$ for some $\nu \in \mathbb{R}^d$, that is,

$$F = \{y \in C(v) \mid \nu \cdot y = 0\}.$$

32:16 Polynomial Time Algorithms in Invariant Theory for Torus Actions

Since $C(v)$ is rational, we may assume that ν has integer components. We assign to F the subset of indices

$$S_F := \{j \in \text{supp}(v) \mid m^{(j)} \in F\}$$

and define $v_F := v|_{S_F}$. Let us check that the orbit O_{v_F} of v_F is contained in $\overline{O_v}$. The one-parameter subgroup $\sigma: \mathbb{C}^\times \rightarrow T$ given by $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$ satisfies

$$\sigma(\epsilon) \cdot v = \rho_M(\sigma(\epsilon))v = (\epsilon^{\nu \cdot m^{(1)}} v_1, \dots, \epsilon^{\nu \cdot m^{(n)}} v_n). \quad (4)$$

It follows that $\lim_{\epsilon \rightarrow 0} \sigma(\epsilon) \cdot v = v_F$ and hence $v_F \in \overline{O_v}$. The same reasoning shows that $v_F \in \overline{O_{v_{F'}}}$ if F is a face contained in the face F' .

The following result is well known, see e.g., [56, Example 1.3], but we sketch a proof for completeness.

► **Proposition 3.5.** *The map $F \mapsto O_{v_F}$ is a bijection between the set of faces of $C(v)$ and the set of orbits contained in $\overline{O_v}$. Moreover, we have*

$$F \subseteq F' \iff \overline{O_{v_F}} \subseteq \overline{O_{v_{F'}}}.$$

The proof of surjectivity relies on a strengthening of the Hilbert–Mumford criterion (Theorem 2.3). Recall this states that if we consider a closed subset S that is stable under the group action and intersects the orbit closure of some point v , then there is a one-parameter subgroup that will drive v to a point in S in the limit. However, a subtle point is that this requires S to be closed. In general, orbits are not closed, so a point w could be in the orbit closure of a point v , but the orbit of w may not be closed. In this case, Theorem 2.3 does not apply to $S = O_w$, and indeed the orbit of w need not be reachable from v by a limit of a one-parameter subgroup. The following theorem shows that for torus actions such a phenomenon does not happen. This crucial fact will also prove useful for us algorithmically in Section 7.

► **Theorem 3.6** ([46], Kapitel III.2.2). *Let $\rho: T \rightarrow \text{GL}(V)$ be a rational representation. Suppose $v, w \in V$ are such that $w \in \overline{O_v}$. Then there exists a one-parameter subgroup $\sigma: \mathbb{C}^\times \rightarrow T$ such that*

$$\lim_{\epsilon \rightarrow 0} \sigma(\epsilon) \cdot v \in O_w.$$

Before we prove Proposition 3.5, we discuss a bit about the structure of one-parameter subgroups. For each $\nu \in \mathbb{Z}^d$, we define a one-parameter subgroup of T , namely $\sigma: \mathbb{C}^\times \rightarrow T$ defined by $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$. Any one-parameter subgroup of T is of this form. This gives an identification of abelian groups $\mathbb{Z}^d \cong \mathcal{Y}(T)$, where $\mathcal{Y}(T)$ denotes the collection of all one-parameter subgroups of T .

We leave the proof of the following well known lemma to the reader.

► **Lemma 3.7.** *Let $\sigma: \mathbb{C}^\times \rightarrow T$ be a one-parameter subgroup, so $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$ for some $\nu \in \mathbb{Z}^d$, and let $v \in \mathbb{C}^n$.*

1. *The limit $\lim_{t \rightarrow 0} \sigma(t) \cdot v$ exists if and only if $m^{(j)} \cdot \sigma \geq 0$ for all $j \in \text{supp}(v)$.*
2. *If the limit exists, then $\lim_{t \rightarrow 0} \sigma(t) \cdot v = v|_S$, where $S = \{j \in \text{supp}(v) \mid m^{(j)} \cdot \sigma = 0\}$.*

Proof of Proposition 3.5. We have already verified that O_{v_F} is an orbit contained in $\overline{O_v}$, hence $F \mapsto O_{v_F}$ is well-defined as a map from the set of faces of $C(v)$ to the set of orbits contained in $\overline{O_v}$. To see that it is injective, note that F is the cone generated by

$\text{supp}(v_F) = S_F$. For surjectivity, let O_w be an orbit contained in $\overline{O_v}$ and $\sigma: \mathbb{C}^\times \rightarrow T$ be a one-parameter subgroup as in Theorem 3.6. There is $\nu \in \mathbb{Z}^d$ such that $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$. By Lemma 3.7, the existence of $\lim_{\epsilon \rightarrow 0} \sigma(\epsilon) \cdot v$ means that $\nu \cdot m^{(j)} \geq 0$ for all $j \in \text{supp}(v)$. In other words, $C(v)$ is contained in the halfspace $\{y \in \mathbb{R}^d \mid \nu \cdot y \geq 0\}$. Moreover, the limit equals v_F , where F is the face $F := \{y \in C(v) \mid \nu \cdot y = 0\}$ of $C(v)$. Therefore, $v_F \in O_w$, hence $O_{v_F} = O_w$, and we have shown surjectivity.

In order to show the remaining equivalence, recall that we argued below (4) that if $F \subseteq F'$ then $v_F \in \overline{O_{v_{F'}}$. The preceding argument also implies the converse. ◀

As an immediate consequence of Proposition 3.5, we get the following result, which not only improves Lemma 3.3 but also characterizes the closed orbit in an orbit closure. For this, define

$$\tilde{v} := v|_{L(v)} = v|_{\text{e-supp}(v)}.$$

► **Corollary 3.8.** *The orbit $O_{\tilde{v}}$ corresponding to the lineality space $L(v)$ is contained in every orbit closure contained in $\overline{O_v}$. Therefore, it is the unique closed orbit contained in $\overline{O_v}$.*

In particular, the orbit O_v is closed if and only if $C(v) = L(v)$, i.e., $C(v)$ equals its linear span. Moreover, v is in the null cone if and only if $\text{e-supp}(v) = \emptyset$.

4 Generating Laurent polynomials and rational invariants

In this section, we discuss the computation of suitable rational invariants, which is the heart of our algorithms, and the main novelty of this paper. As explained in the introduction, the starting point is the simple observation that two orbits can only be equal when they have the same support (Lemma 3.2). But once we restrict to vectors of fixed support, it is natural to consider a larger class of invariants, namely Laurent polynomials, which are polynomials that can also have negative exponents. In Section 4.1 we will see that the invariant Laurent polynomials for a given support naturally form a lattice that can be computed from the weight matrix. This allows us to give an efficient algorithm for computing small sets of generators. As a consequence, we can also efficiently compute a system of generating rational invariants.

For the rest of this section, we fix an n -dimensional representation $\rho_M: T \rightarrow \text{GL}_n(\mathbb{C})$ of the torus $T = (\mathbb{C}^\times)^d$ given by a weight matrix $M \in \text{Mat}_{d,n}(\mathbb{Z})$ with columns $m^{(j)}$ for $j \in [n]$.

4.1 Invariant Laurent polynomials

For $S \subseteq [n]$, consider the set of vectors with support S , that is, the variety

$$X_S = \{v \in \mathbb{C}^n \mid \text{supp}(v) = S\} = \{v \in \mathbb{C}^n \mid v_j \neq 0 \text{ if and only if } j \in S\}. \tag{5}$$

The ring of regular functions on X_S , denoted $\mathbb{C}[X_S]$, is naturally identified with the ring of Laurent polynomials in variables $\{x_j\}_{j \in S}$. That is,

$$\mathbb{C}[X_S] = \mathbb{C}[x_j, x_j^{-1} \mid j \in S].$$

We observe that ρ_M restricts to an action of T on X_S , and induces an action on $\mathbb{C}[X_S]$. The proposition below shows that the algebra $\mathbb{C}[X_S]^T$ of *invariant Laurent polynomials* can be succinctly described in terms of the lattice

$$L_S = \left\{ c \in \mathbb{Z}^S \mid \sum_{j \in S} c_j m^{(j)} = 0 \right\} = \ker(M_S) \cap \mathbb{Z}^{|S|}, \tag{6}$$

where $\mathbb{Z}^S := \{c \in \mathbb{R}^n \mid c_j = 0 \text{ for all } j \notin S\} \cong \mathbb{Z}^{|S|}$, and M_S denotes the submatrix of the weight matrix M , obtained by removing all columns except those labeled by S .

► **Proposition 4.1.**

1. Let $c \in \mathbb{Z}^S$. A Laurent monomial $x^c = \prod_{j \in S} x_j^{c_j}$ is invariant if and only if $c \in L_S$.
2. The algebra of invariant Laurent polynomials $\mathbb{C}[X_S]^T$ is spanned as a vector space by the invariant Laurent monomials.
3. If $\{c^{(1)}, c^{(2)}, \dots, c^{(r)}\}$ is a lattice basis of L_S , then $\mathbb{C}[X_S]^T$ is generated as an algebra by the invariant Laurent monomials $\{x^{c^{(1)}}, \dots, x^{c^{(r)}}\}$.

Proof. The first two parts are shown using an argument similar to the proof of Proposition 3.1. The third statement is an immediate consequence. ◀

It is instructive to compare this with the discussion below Proposition 3.1, where we saw that the invariant polynomials are similarly described by the *semigroup* of nonnegative vectors in the kernel of the weight matrix. By working with vectors of fixed support, we instead obtain a natural lattice structure, which simplifies the situation considerably. For example, the lattice L_S and hence the algebra of invariant Laurent polynomials $\mathbb{C}[X_S]^T$ have at most $|S| \leq n$ generators – in stark contrast to the situation for invariant polynomials.

We now discuss how to compute lattice bases as in Proposition 4.1. It is well known that every integer matrix M can be diagonalized by multiplying from left and right with unimodular matrices. This is known as the *Smith normal form* [59]. The Smith normal form can be computed in polynomial time [43]. We record these facts in the following theorem.

► **Theorem 4.2 (Smith normal form).** Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$. Then, there exist unimodular matrices $U \in \text{Mat}_{d,d}(\mathbb{Z}), W \in \text{Mat}_{n,n}(\mathbb{Z})$ such that

$$UMW = \begin{bmatrix} \alpha_1 & 0 & 0 & \dots & 0 \\ 0 & \alpha_2 & 0 & \dots & 0 \\ 0 & 0 & \ddots & & 0 \\ & & & \alpha_r & \vdots \\ \vdots & \vdots & & 0 & \\ & & & & \ddots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

and the diagonal elements satisfy $\alpha_i \mid \alpha_{i+1}$ for $i = 1, 2, \dots, r - 1$, where r equals the rank of M . The matrix UMW is unique and called the *Smith normal form* of M .

Moreover, if the bit-lengths of the entries of M are bounded by b , then the matrices U, W , and UMW can be computed in $\text{poly}(d, n, b)$ -time.

Using the Smith normal form it is easy to compute a basis of the lattice L_S . We state this in the following algorithm and corollary.

■ **Algorithm 1** Computation of a basis of the lattice of invariant Laurent monomials.

Input $M \in \text{Mat}_{d,n}(\mathbb{Z})$ and $S \subseteq [n]$.

Step 1 Compute the submatrix M_S of M obtained by deleting all columns except those in S .

Step 2 Compute the Smith normal form $UM_S W$ of M_S (as in Theorem 4.2).

Step 3 Return $\{w^{(r+1)}, w^{(r+2)}, \dots, w^{(n)}\}$, where $w^{(j)}$ denotes the j^{th} column of W .

► **Corollary 4.3.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ and $S \subseteq [n]$, and suppose the bit-lengths of the entries of M are bounded by b . Then Algorithm 1 computes a basis for the lattice L_S defined in (6) in $\text{poly}(d, n, b)$ -time. In particular, each $w^{(j)}$ has bit-length $\text{poly}(d, n, b)$.*

Alternatively, one can use lattice algorithms; we refer the interested reader to [31, Corollary 5.4.10].

► **Remark 4.4.** It is easy to see that given an exponent vector $c = (c_1, \dots, c_n) \in \mathbb{Z}_{\geq 0}^n$, where the bit-lengths of the c_i s are bounded by b , an arithmetic circuit computing the monomial x^c of size $\text{poly}(n, b)$ can be constructed in $\text{poly}(n, b)$ -time. Similarly, if $c \in \mathbb{Z}^n$, an arithmetic circuit with division computing the Laurent monomial x^c can be constructed in $\text{poly}(n, b)$ -time.

Proof of Theorem 1.3. This follows from Proposition 4.1, Corollary 4.3, and Remark 4.4. ◀

4.2 Rational invariants

In the remainder of this section we will discuss rational invariants. For $V = \mathbb{C}^n$, recall that $\mathbb{C}[V] = \mathbb{C}[x_1, \dots, x_n]$ is the polynomial ring in n variables. Let $\mathbb{C}(V) = \mathbb{C}(x_1, \dots, x_n)$ the field of rational functions (its fraction field). In other words, any element in $\mathbb{C}(V)$ is a ratio of two polynomials. The action of T on $\mathbb{C}[V]$ extends to $\mathbb{C}(V)$. Then $\mathbb{C}(V)^T$ is the field of *rational invariants*. Clearly, any invariant Laurent polynomial is a rational invariant, but the converse need not be the case.

Nevertheless, we can show that the invariant Laurent polynomials in all variables (that is, for support $S = [n]$) generate the rational invariants as a field.

► **Proposition 4.5.** *Let $A := \mathbb{C}[X_{[n]}] = \mathbb{C}[x_1, x_1^{-1}, \dots, x_n, x_n^{-1}]^T$ denote the algebra of invariant Laurent polynomials, and let $F := \mathbb{C}(x_1, \dots, x_n)^T$ denote the field of rational invariants. Then, A generates F as a field, i.e., the field of fractions of A is F .*

Proof. Let $f \in F^\times$ and write $f = \frac{p}{q}$, where $p, q \in \mathbb{C}[x_1, \dots, x_n]$ have no common factors. Since f is invariant, we have for any $t \in T$ that

$$\frac{t \cdot p}{t \cdot q} = t \cdot f = f = \frac{p}{q}.$$

Accordingly, $t \cdot p = \alpha(t)p$ and $t \cdot q = \alpha(t)q$ for some $\alpha(t) \in \mathbb{C}^\times$. Thus, p and q span one-dimensional representations. This in turn implies that $\alpha: T \rightarrow \mathbb{C}^\times$ is a character, as discussed in Section 3.1, and further that p (and also q) is a sum of monomials with the same weight, i.e., $p = \sum_e p_e x^e$ such that $t \cdot x^e = \alpha(t)x^e$ for $p_e \neq 0$. In particular, $f_e = \frac{p_e}{q_e}$ is a Laurent polynomial *invariant* if $p_e \neq 0$, and we can write

$$f = \frac{p}{q} = \sum_e p_e \frac{x^e}{q} = \sum_e p_e \frac{1}{f_e},$$

which concludes the proof. ◀

As a direct consequence, any system of generating invariant Laurent polynomials (as an algebra) also serves as a system of generating rational invariants (as a field extension of \mathbb{C}). Thus we obtain:

Proof of Corollary 1.4. This follows from Theorem 1.3 (with $S = [n]$) and Proposition 4.5. ◀

5 Orbit equality problem

In this section, we will give a polynomial time algorithm for the orbit equality problem. Given two points, the strategy is to compute a small collection of invariant Laurent monomials (using the result of Section 4) whose evaluations at the two given points will determine whether the two points are in the same orbit. The efficient testing of whether two Laurent monomials evaluate to the same value actually requires an idea: this has already been studied in the literature and we briefly sketch in Section 5.1 how to do this.

We still assume an n -dimensional representation $\rho_M: T \rightarrow \mathrm{GL}_n(\mathbb{C})$ of the torus $T = (\mathbb{C}^\times)^d$ given by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ with columns $m^{(j)}$ for $j \in [n]$.

In general, invariants can only decide orbit closure intersection, not orbit equality. However, the crucial point is that in the varieties (5) consisting of vectors of fixed support any T -orbit is closed.

► **Proposition 5.1.** *Let $S \in [n]$, X_S be the variety defined in (5), and $v \in X_S$. Then the orbit O_v is a closed subset of X_S .*

Proof. By Lemma 3.2 (3) we have $O_v = \overline{O_v} \cap X_S$ which implies that the orbits are closed in X_S . ◀

Orbit equality in V can always be reduced to orbit equality in some X_S , since equality of supports is a necessary condition (Lemma 3.2 (1)). The importance of the above result is that the latter orbit equality and orbit closure intersection are equivalent in X_S . Together with Theorem 2.1 we obtain the following result.

► **Corollary 5.2.** *Suppose $\mathrm{supp}(v) = \mathrm{supp}(w) = S$. Then, $O_v \neq O_w$ if and only if there is an invariant Laurent monomial $f = \prod_{j \in S} x_j^{c_j}$ such that $f(v) \neq f(w)$.*

Thus, we obtain the following algorithm for the orbit equality problem.

■ **Algorithm 2** Deciding orbit equality.

Input $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ and $v, w \in \mathbb{Q}(i)^n$.

Step 1 Check if $\mathrm{supp}(v) = \mathrm{supp}(w)$. If not, $O_v \neq O_w$, so we can stop.

Step 2 Use Algorithm 1 to compute a lattice basis \mathcal{B} for the lattice L_S defined in (6).

Step 3 For each $e \in \mathcal{B}$, we check if $v^e = w^e$ (as described in Section 5.1 below).

If they are all equal, then $O_v = O_w$. Else, $O_v \neq O_w$.

Proof of Theorem 1.2, part (1). The correctness of Algorithm 2 follows from Proposition 4.1 and Corollary 5.2. We now analyze its runtime. Clearly, the first step can be implemented efficiently. For the second step, we can appeal to Corollary 4.3. For step 3, we first observe that, again by Corollary 4.3, the exponents e have bit-length $\mathrm{poly}(d, n, b)$. Then Proposition 5.4 below shows that this step can also be implemented in time $\mathrm{poly}(d, n, b)$. ◀

5.1 Laurent monomial equivalence

We now discuss how to test if a Laurent monomial x^e evaluates to the same value at two points v and w . In our context, where each component e_j of the exponent vector $e = (e_1, \dots, e_n)$ has poly-sized bit-lengths, it is unreasonable to evaluate the Laurent monomials explicitly, because the answer may very well require exponentially large bit-length. Yet, it is possible

to check if $v^e = w^e$ efficiently. We describe a simple algorithm based on g.c.d.'s, which has appeared before (see, for example, [25]) in the case where the entries of v and w are in \mathbb{Z} (or equivalently \mathbb{Q}). The result is much older; for example, it follows from the results in [3], as mentioned in [30], which gives a generalization to number fields.¹⁵ Here we present a short self-contained proof and then follow up with the rather simple extension to Gaussian rationals.

► **Lemma 5.3.** *Suppose $a_1, \dots, a_k, b_1, \dots, b_r \in \mathbb{Q}$ and $e_1, \dots, e_k, f_1, \dots, f_r \in \mathbb{Z}$ have bit-lengths at most s . Then, in $\text{poly}(k, r, s)$ -time, we can decide if $\prod_{i=1}^k a_i^{e_i} = \prod_{j=1}^r b_j^{f_j}$.*

Proof. By clearing denominators, we may assume that $a_1, \dots, a_k, b_1, \dots, b_r$ are integers. By moving terms to the other side, we can further assume w.l.o.g. that all $e_i, f_j \geq 0$. Pick some a_l and some b_m that are not coprime. Then, consider $d = \text{gcd}(a_l, b_m) \geq 2$. W.l.o.g., we can assume $e_l \geq f_m$. Then, test if $d^{e_l - f_m} (a_l')^{e_l} \prod_{i \neq l} a_i^{e_i} = (b_m')^{f_m} \prod_{j \neq m} b_j^{f_j}$, where $a_l' = a_l/d$ and $b_m' = b_m/d$. This is an iterative procedure which stops when each a_i is coprime to b_j . At which point, unless all a_i 's and b_j 's are equal to 1, both sides cannot be equal.

The question is how long does such an iterative procedure take. Consider the quantity $P := |a_1 \cdots a_k b_1 \cdots b_r|$. After applying one step, the resulting quantity P' satisfies $P' = P/d^2 \leq P/4$. Since initially, P is $2^{\text{poly}(k, r, s)}$ -sized, there are at most a polynomial number of iterative steps. Hence, the entire procedure takes $\text{poly}(k, r, s)$ -time. ◀

An analogous result with the same proof holds for the ring $\mathbb{Z}[i]$ of Gaussian integers and its quotient field $\mathbb{Q}(i)$ of Gaussian rationals, using that this ring has unique factorization into irreducible elements. In the following proposition, we assume that a Gaussian rational $a = \alpha + i\beta \in \mathbb{Q}(i)$ is described by giving the encodings of α and β in binary.

► **Proposition 5.4.** *Suppose $a_1, \dots, a_k, b_1, \dots, b_r \in \mathbb{Q}(i)$ and $e_1, \dots, e_k, f_1, \dots, f_r \in \mathbb{Z}$ all have bit-lengths bounded by s . Then, in $\text{poly}(k, r, s)$ -time, we can decide if $\prod_{i=1}^k a_i^{e_i} = \prod_{j=1}^r b_j^{f_j}$.*

► **Remark 5.5.** For computational purposes, in many instances, numbers are described by their “floating point” representations. The floating point description of a Gaussian rational $a \in \mathbb{Q}(i)$ is described by giving the binary encodings of $\alpha, \beta \in \mathbb{Q}$ and $p \in \mathbb{Z}$ such that $a = (\alpha + i\beta)2^p$. If we assume that $a_1, \dots, a_k, b_1, \dots, b_r \in \mathbb{Q}(i)$ in the proposition above are given by their floating point descriptions, we can still decide monomial equivalence in polynomial time. Indeed, if we write each $a_j = (\alpha_j + i\beta_j)2^{p_j}$ and $b_j = (\gamma_j + i\delta_j)2^{q_j}$, then deciding whether $\prod_{j=1}^k a_j^{e_j} = \prod_{j=1}^r b_j^{f_j}$ simplifies to deciding if

$$\left(\prod_{j=1}^k (\alpha_j + i\beta_j)^{e_j} \right) \cdot 2^{\sum_{j=1}^k e_j p_j} = \left(\prod_{j=1}^r (\gamma_j + i\delta_j)^{f_j} \right) \cdot 2^{\sum_{j=1}^r f_j q_j},$$

which can again be interpreted as an instance of Proposition 5.4 and hence can be checked in polynomial time. Since all other computations in our algorithms only involve supports of vectors, it follows that all results in this paper generalize to this input model, as claimed in footnote 6.

An even easier special case arises for numbers of the form $a = 2^p$, with $p \in \mathbb{Q}$ specified by its binary encoding, as in the perfect matching application discussed in Section 1.4. Indeed, if $a_j = 2^{p_j}$ and $b_j = 2^{q_j}$ for $j \in [n]$, then deciding whether $\prod_{j=1}^k a_j^{e_j} = \prod_{j=1}^r b_j^{f_j}$ simply amounts to verifying whether $\sum_{j=1}^k p_j e_j = \sum_{j=1}^r q_j f_j$, which is clearly possible in polynomial time.

¹⁵In particular Ge’s result [30] implies that Theorem 1.2 extends to the case where the entries of v and w are taken from some algebraic number field.

6 Orbit closure intersection and explicit separating invariants

In this section, we discuss how to solve the orbit closure intersection problem in polynomial time by efficiently reducing it to the orbit equality problem. The problem of orbit closure intersection has a manifestly analytic point of view, but also an algebraic point of view by Mumford's theorem, Theorem 2.1. In other words, when orbit closures of two points do not intersect, there is an invariant polynomial that takes different values on both points, serving as a “witness” to the fact that the orbit closures do not intersect. Accordingly, given two vectors whose orbit closures do not intersect, we also explain how to efficiently construct an arithmetic circuit which computes an invariant monomial separating the two vectors.

6.1 Reduction to orbit equality

The key idea is the following. Recall from Theorem 2.2 that any orbit closure $\overline{O_v}$ contains as unique closed orbit $O_{\tilde{v}}$, and that two orbit closures intersect if and only if they contain the same closed orbit. In Corollary 3.8, we showed that the unique closed orbit has a concrete polyhedral characterization: we can take $\tilde{v} = v|_{\text{e-supp}(v)}$, the restriction of the vector v to its essential support. Accordingly, the map $v \mapsto \tilde{v}$ provides a reduction of the orbit closure intersection problem for ρ_M to the orbit equality problem for ρ_M . The following lemma shows that the essential support (and hence the reduction map) can be computed in polynomial time by using linear programming.

► **Lemma 6.1.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$, and let $v \in \mathbb{C}^n$. For $k \in \text{supp}(v)$, we have $k \in \text{e-supp}(v)$ if and only if there is a non-negative linear combination $\sum_{j \in \text{supp}(v)} c_j m^{(j)} = 0$ such that $c_k > 0$. If the bit-lengths of the entries of M are bounded by b , the latter can be decided in $\text{poly}(d, n, b)$ -time by using linear programming.*

Proof. The characterization follows from Proposition 3.1 and Lemma 3.3. It amounts to a basic decisional problem of linear programming, which is well known to be solvable in polynomial time, see [31]. ◀

The above proof also shows that a nonvanishing invariant monomial as in Lemma 6.1 can be computed in polynomial time. As explained above, we arrive at the following algorithm and results.

■ **Algorithm 3** Reduction of orbit closure intersection to orbit equality.

Input $M \in \text{Mat}_{d,n}(\mathbb{Z}), v, w \in \mathbb{Q}(i)^n$.

Step 1 Compute $\text{e-supp}(v)$ in the following way: For each $k \in \text{supp}(v)$, use linear programming to determine if there is a non-negative linear combination $\sum_{j \in \text{supp}(v)} c_j m^{(j)} = 0$ with $c_k > 0$. The set $\text{e-supp}(v)$ consists of all $k \in \text{supp}(v)$ for which this is the case.

Step 2 Compute $\text{e-supp}(w)$ in the same way.

Step 3 Return $\tilde{v} = v|_{\text{e-supp}(v)}$ and $\tilde{w} = w|_{\text{e-supp}(w)}$.

► **Corollary 6.2.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ describe an n -dimensional representation of $T = (\mathbb{C}^\times)^d$. Further, let $v, w \in \mathbb{Q}(i)^n$ and assume the bit-lengths of the entries of M, v , and w are bounded by b . Then there is a $\text{poly}(d, n, b)$ -time reduction that reduces the problem of deciding $\overline{O_v} \cap \overline{O_w} \neq \emptyset$ to the problem of deciding if $O_{\tilde{v}} = O_{\tilde{w}}$, where \tilde{v} and \tilde{w} have bit-lengths bounded by b .*

Proof of Theorem 1.2, part (2). This follows from part (1), combined with Corollary 6.2. ◀

6.2 Explicit separating invariant

For torus actions, our reduction of orbit closure intersection to orbit equality will give us an invariant Laurent monomial that takes different values on the two points. But a separating invariant Laurent monomial itself does *not* serve as a witness (at least not naively, one needs further properties about the support of the Laurent monomial for it to serve as a witness). We now prove Corollary 1.5, which asserts that given two vectors we can nevertheless efficiently construct an arithmetic circuit which computes an invariant *monomial* separating them.

Proof of Corollary 1.5. We already noted that, by linear programming, we can compute the essential supports of v and w in $\text{poly}(d, n, b)$ -time. We distinguish two cases.

Case 1: $\text{e-supp}(v) \neq \text{e-supp}(w)$

Suppose $k \in \text{e-supp}(v) \setminus \text{e-supp}(w)$ without loss of generality. By Lemma 3.3 there is an invariant monomial $f = \prod_{j \in \text{supp}(v)} x_j^{c_j}$ such that $c_k > 0$. Let us verify that $f(v) \neq f(w)$. We clearly have $f(v) \neq 0$. On the other hand, $f(w) = f(\tilde{w}) = 0$, since $\tilde{w} \in \overline{O_w}$, but k is not contained in $\text{supp}(\tilde{w}) = \text{e-supp}(w)$. So we indeed have $f(v) \neq f(w)$. In addition, we can find (c_1, \dots, c_n) in $\text{poly}(d, n, b)$ -time by linear programming (Lemma 6.1), so we can construct an arithmetic circuit for f in $\text{poly}(d, n, b)$ -time by Remark 4.4.

Case 2: $\text{e-supp}(v) = \text{e-supp}(w)$

Let $S := \text{e-supp}(v) = \text{e-supp}(w)$. We assume that $\overline{O_v} \cap \overline{O_w} = \emptyset$, which implies $O_{\tilde{v}} \cap O_{\tilde{w}} = \emptyset$. Thus, by Corollary 5.2, there is an invariant Laurent monomial $f = x^e$ with the property that $f(\tilde{v}) \neq f(\tilde{w})$, and hence $f(v) \neq f(w)$. Just like in Algorithm 2, we can in $\text{poly}(d, n, b)$ -time compute such an exponent vector $e \in \mathbb{Z}^n$, with bit-length of the e_i bounded above by $\text{poly}(d, n, b)$.

Our goal is to produce an invariant monomial that separates v and w , so we need to modify f so as to get rid of the negative exponents. In the process, we must ensure that the bit-length of the circuit does not explode. By Lemma 3.3, for each $k \in S$, there exists $c^{(k)} \in \mathbb{Z}_{\geq 0}^n$ such that $\sum_{j \in \text{supp}(v)} c_j^{(k)} m_j^{(j)} = 0$ and $c_k^{(k)} > 0$. We can compute $c^{(k)}$ in $\text{poly}(d, n, b)$ -time by linear programming. Let $m_k = x^{c^{(k)}}$ denote the corresponding invariant monomial. Put $S_- := \{j \in S \mid e_j < 0\}$. If $m_j(v) \neq m_j(w)$ for some $j \in S_-$, then m_j is an explicit separating invariant monomial and we are done by Remark 4.4. Assume now $m_j(v) = m_j(w)$ for all $j \in S_-$. Then $\tilde{f} := x^d := f \cdot \prod_{j \in S_-} m_j^{-e_j}$ is a Laurent monomial that separates v and w . We verify now that the exponent vector d has non-negative entries. By construction, we have for $k \in S_-$,

$$d_k = e_k + (-e_k)c_k^{(k)} + \sum_{j \in S_-, j \neq k} (-e_j) \cdot c_k^{(j)} \geq 0,$$

since $e_k < 0$ and $e_j < 0$ for all $j \in S_-$, while $c_k^{(k)} \geq 1$, and $c_k^{(j)} \geq 0$. For $k \in [n] \setminus S_-$, we have

$$d_k = e_k + \sum_{j \in S_-} (-e_j) \cdot c_k^{(j)} \geq 0,$$

since $e_k \geq 0$ for $k \in S \setminus S_-$ and $e_k = 0$ for $k \notin S$, while $e_j < 0$ for $j \in S_-$. Altogether, we have shown that indeed all components of d are non-negative. We finally note that d can be computed in polynomial time, in particular, it has bit-length $\text{poly}(d, n, b)$. So by Remark 4.4, we can construct an arithmetic circuit of size $\text{poly}(d, n, b)$ that computes \tilde{f} in $\text{poly}(d, n, b)$ -time. ◀

7 Orbit closure containment

In this section, we discuss how to solve the orbit closure containment problem in polynomial time by efficiently reducing it to the orbit equality problem.

The notion of orbit closure containment is in general quite tricky to capture. Polynomial invariants do not suffice, since two orbit closures can intersect (hence all polynomial invariants agree) with neither being contained in the other – this is precisely the difference between the orbit closure intersection and the orbit closure containment problem. Instead, the key idea for the reduction comes from one-parameter subgroups. We already discussed in Section 3 that if $w \in \overline{O_v}$ then O_w can be reached from v by a one-parameter subgroup. The following proposition gives a concrete polyhedral description of the relevant one-parameter subgroups.

► **Lemma 7.1.** *Let $M \in \text{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$, and let $v, w \in \mathbb{C}^n$. Then $w \in \overline{O_v}$ if and only if there exists a one-parameter subgroup $\sigma: \mathbb{C}^\times \rightarrow T$, so $\sigma(\epsilon) = (\epsilon^{\nu_1}, \dots, \epsilon^{\nu_d})$ for some $\nu \in \mathbb{Z}^d$, such that*

1. $\{j \in \text{supp}(v) \mid m^{(j)} \cdot \nu = 0\} = \text{supp}(w)$ and $m^{(k)} \cdot \nu > 0$ for all $k \in \text{supp}(v) \setminus \text{supp}(w)$;
2. $O_{(v|_{\text{supp}(w)})} = O_w$.

Proof. If $w \in \overline{O_v}$, then by Theorem 3.6, we know that there is a one-parameter subgroup σ such that $\lim_{t \rightarrow 0} \sigma(t)v \in O_w$. In particular this implies that $\lim_{t \rightarrow 0} \sigma(t)v$ has the same support as w and has the same orbit as w . Now, both (1) and (2) follow from Lemma 3.7.

For the converse, note that, again by Lemma 3.7, (1) implies that $\lim_{t \rightarrow 0} \sigma(t)v = v|_{\text{supp}(w)} \in \overline{O_v}$, hence it follows that $O_w = O_{(v|_{\text{supp}(w)})} \subseteq \overline{O_v}$ by (2). ◀

Now, we can give our algorithm to test if w is in the orbit closure of v .

■ **Algorithm 4** Orbit closure containment.

Input $M \in \text{Mat}_{d,n}(\mathbb{Z})$ and $v, w \in \mathbb{Q}(i)^n$.

Step 1 Check if $\text{supp}(w) \subseteq \text{supp}(v)$. If not, $w \notin \overline{O_v}$, so we can stop.

Step 2 Using linear programming, determine whether there exists a solution $y \in \mathbb{R}^d$ to the collection of linear equalities $m^{(j)} \cdot \nu = 0$ for each $j \in \text{supp}(w)$ and linear inequalities $m^{(k)} \cdot \nu > 0$ for all $k \in \text{supp}(v) \setminus \text{supp}(w)$. If there is no solution, then $w \notin \overline{O_v}$, so we can stop.

Step 3 Use Algorithm 2 check whether $O_{(v|_{\text{supp}(w)})} = O_w$. If yes, then $w \in \overline{O_v}$. Else, it is not.

Proof of Theorem 1.2, part (3). The correctness of Algorithm 4 follows from Lemma 7.1. Indeed, condition (1) in the lemma is satisfied if and only if the algorithm passes the first two steps, and then condition (2) is tested in the last step.

We still need to argue about the efficiency of the algorithm. Clearly, Step 1 can be done in linear time. Step 2 can be done in $\text{poly}(d, n, b)$ -time by linear programming. Step 3 appeals to the orbit equality problem, which by part (1) of the theorem can be done in $\text{poly}(d, n, b)$ -time. ◀

8 Orbit problems for compact tori

So far, we have studied orbit problems for algebraic tori, that is, groups of the form $T = (\mathbb{C}^\times)^d$. In this section we consider the groups $K = (\mathbb{S}^1)^d$, where $\mathbb{S}^1 = \{z \in \mathbb{C}^\times \mid |z| = 1\}$. Such groups are often called *compact tori*. Indeed, any commutative compact connected Lie group is of this form. Besides the fundamental algorithmic interest in this setting, it is also important in applications. For example, in physics, symmetries are often given by compact group actions, such as compact tori [32, 2]. We give further complexity-theoretic motivation below.

The compactness implies that orbits are closed and so the three problems in Problem 1.1 coincide. In this section, we show how to solve the orbit equality problem for a compact torus by reducing it to orbit equality for the corresponding algebraic torus. Subsequently, we give an alternative reduction that works not only for tori but in fact for any connected reductive group such as SL_n .

To start, we note that it is known that any (continuous) finite-dimensional representation of $K = (\mathbb{S}^1)^d$ extends to a representation of $T = (\mathbb{C}^\times)^d$ [62]. In particular, representations can be specified as before by a weight matrix $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$. Then we have the following result:

► **Proposition 8.1.** *Let $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ define an n -dimensional representation of $T = (\mathbb{C}^\times)^d$ and $K = (\mathbb{S}^1)^d$. Let $v, w \in \mathbb{C}^n$. Then, $O_{K,v} = O_{K,w}$ if and only if $O_{T,v} = O_{T,w}$ and $|v_j| = |w_j|$ for all j .*

Proof. Since $K \subseteq T$, it is clear that if $O_{K,v} = O_{K,w}$, then $O_{T,v} = O_{T,w}$ and $|v_j| = |w_j|$ for all j .

Conversely, suppose $O_{T,v} = O_{T,w}$ and $|v_j| = |w_j|$ for all j . Then, there is some $t \in T$ such that $t \cdot v = w$. Write $t = (t_1, \dots, t_d)$ and write each $t_i = r_i \cdot e^{i\theta_i}$, with $r_i > 0$ and $\theta_i \in \mathbb{R}$. Then, it is easy to see that we must have $(e^{i\theta_1}, \dots, e^{i\theta_d}) \cdot v = w$. Thus v and w are in the same K -orbit. ◀

Proof of Corollary 1.6. We are given $M \in \mathrm{Mat}_{d,n}(\mathbb{Z})$ and $v, w \in \mathbb{Q}(i)^n$. By the above proposition, we need to check if $O_{T,v} = O_{T,w}$ and if $|v_j| = |w_j|$ for all j . The former can be done in polynomial time by Theorem 1.2 and the latter can clearly be done in polynomial time. ◀

Before proceeding we give some further context and motivation. Algorithms for the null cone membership problem (given a rational representation $\rho : G \rightarrow \mathrm{GL}(V)$ of a reductive group G and $v \in V$, decide if $0 \in \overline{O_v}$) based on optimization methods have emerged in recent years. They take advantage of the fact that $0 \in \overline{O_v}$ if and only if one can drive the norm to 0 along the orbit O_v . This can be viewed as an optimization problem where one tries to minimize (infimize) the norm along the orbit. While this is not a convex optimization problem, it is geodesically convex by the Kempf-Ness theory [45], which allows for many of the ideas to be modified appropriately. As far as the orbit closure intersection problem is concerned, the natural extension of this idea is as follows: Given $v, w \in V$, first use an optimization algorithm to approximately find a point in each orbit closure with minimal norm; let us call these points \check{v}, \check{w} . Then, appealing to the Kempf-Ness theory again, we have that $\overline{O_v} \cap \overline{O_w} \neq \emptyset$ if and only if \check{v} and \check{w} are in the same orbit for a maximal compact subgroup K of G . In this way, the orbit closure intersection problem for G can be reduced to the orbit equality problem for the maximal compact subgroup K . In fact, for the so-called left-right action of $\mathrm{SL}_n \times \mathrm{SL}_n$ on matrix-tuples, this idea was carried out successfully to obtain a polynomial-time algorithm for orbit closure intersection [1]. This further emphasizes the importance of the orbit equality problem for compact Lie group actions.

Here we report on an interesting phenomenon, which provides a kind of converse to the strategy explained above. Namely, for any action of a connected reductive group G , the orbit equality problem for the maximal compact subgroup $K \subseteq G$ is equivalent to an orbit intersection (or equality) problem for a related action of G ! As this result is not crucial to the rest of the paper and requires significantly different background, we will be brief in our explanations. We denote by V^* the contragredient or dual representation of V .

► **Theorem 8.2.** *Let $\rho: G \rightarrow \mathrm{GL}(V)$ be a finite-dimensional representation of a connected reductive group G . Let K be a maximal compact subgroup of G , and $\langle \cdot, \cdot \rangle$ be a K -invariant Hermitian inner product on V . For $v \in V$, let $\widehat{v} \in V^*$ be defined by $\widehat{v}(w) := \langle v, w \rangle$. Then, for $v, w \in V$, the following are equivalent:*

1. $O_{K,v} = O_{K,w}$;
2. $O_{G,(v,\widehat{v})} = O_{G,(w,\widehat{w})}$ in $V \oplus V^*$;
3. The G -orbit closures of (v, \widehat{v}) and (w, \widehat{w}) in $V \oplus V^*$ intersect.

Proof. Let $\mathrm{Lie}(G) \subseteq L(V)$ denote the Lie algebra of G . For any linear action of G on a vector space U , we get an induced action of $\mathrm{Lie}(G)$ on U . Given a K -invariant Hermitian form $\langle \cdot, \cdot \rangle$ on U , we define the so-called moment map $\mu_U: U \rightarrow \mathrm{Lie}(G)^*$ by the formula $\mu_U(u)(X) = \langle u, X \cdot u \rangle$ for $u \in U$ and $X \in \mathrm{Lie}(G)$ (up to a scalar which is not relevant for our purposes). The celebrated Kempf-Ness theorem says that if $\mu_U(u) = 0$ then the G -orbit of u is closed. Moreover, it asserts that if $u' \in U$ is another point such that $\mu_U(u') = 0$, then $O_{G,u} = O_{G,u'}$ if and only if $O_{K,u} = O_{K,u'}$.

Applying the preceding to (v, \widehat{v}) and (w, \widehat{w}) in $U = V \oplus V^*$, a simple calculation shows that the moment map vanishes at either point, so the two orbits are closed. This shows the equivalence between (2) and (3). The equivalence between (1) and (2) follows immediately from the second part of the Kempf-Ness theorem, using that $k\widehat{v} = k\widehat{v}$ for any $k \in K$, since K acts unitarily. ◀

9 Concluding remarks, future directions, and open problems

To better understand the context of our results and their potential impact on future progress, we briefly discuss some results in literature and then suggest further research directions. In very high level, we feel that the following aspects are highlighted by this work: the relative power and interplay between algebraic and analytical algorithms, the importance of understanding commutative actions as a stepping stone towards understanding general actions, the role of rational (as opposed to polynomial) invariants, and the subtlety of “no go” results, which evidently can be surpassed.

There has been an explosion of interest over the last decade in understanding invariant theory from a complexity theoretic perspective (we survey some of this literature in the introduction). This rapidly developing field can be seen as an endeavour to classifying computational problems in invariant theory according to their difficulty, finding efficient algorithms whenever possible, as well as connecting to applications in mathematics, physics, optimization, and statistics.

Invariant theory in the setting of a rational representation of a connected reductive group is the most relevant for complexity theory. The commutative case of tori is an important special case. Despite the well-understood structural simplicity of the corresponding invariant theory, even basic algorithmic problems are non-trivial. Null cone membership, arguably the most basic problem, has long been known to have an efficient algorithm, as it reduces to linear programming, which non-trivially admits polynomial-time algorithms. The problems

of orbit equality, orbit closure intersection, and orbit closure containment have polynomial time algorithms, as shown in this paper. We stress that while efficient algorithms for linear programming are “continuous” or “analytic” in nature, our algorithms use a combination of *both* analytic *and* algebraic techniques. The more general problem of succinct circuits for generating polynomial invariants, which is one of the basic challenges proposed in [52], has recently shown to be impossible under natural complexity assumptions [29]. Yet, in this paper, we bypass this negative result, and see that *rational* invariants for torus actions can be captured in a computationally efficient way without the need for succinct circuits. It is an interesting open problem to determine if there are succinct circuits for separating invariants or null cone definers, see [29, Problems 1.14, 1.15].

The invariant theory of non-commutative groups has a different flavor from, and is far more complex than, the commutative case, see, for example, [38]. Many interesting problems in computational invariant theory remain open in the non-commutative case. We list a few. First and foremost, the results in this paper motivate the investigation of the computational efficiency of systems of generating *rational* invariants. Further, it is natural to wonder if rational invariants can help capture orbit closure intersection and orbit equality for non-commutative group actions. Another open problem is to give *any* polynomial time algorithm for orbit closure intersection (and the subproblem of null cone membership). An intermediate challenge is to ascertain whether null cone membership is in $\text{NP} \cap \text{co-NP}$. Note that in [5] it is shown that the general orbit closure containment problem is NP-hard.

References

- 1 Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *STOC’18—Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 172–181. ACM, New York, 2018. doi:10.1145/3188745.3188942.
- 2 Michele Audin. *Torus actions on symplectic manifolds*, volume 93. Birkhäuser, 2012.
- 3 Eric Bach, James R. Driscoll, and Jeffrey O. Shallit. Factor refinement. In David S. Johnson, editor, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California, USA*, pages 201–211. SIAM, 1990. URL: <http://dl.acm.org/citation.cfm?id=320176.320199>.
- 4 Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical review letters*, 70(13):1895, 1993.
- 5 Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. Variety membership testing, algebraic natural proofs, and geometric complexity theory. arXiv, 2020. arXiv:1911.02534.
- 6 Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *Journal of Algebra*, 320(11):4020–4029, 2008. Computational Algebra. doi:10.1016/j.jalgebra.2008.07.014.
- 7 Peter Bürgisser, Matthias Christandl, Ketan D. Mulmuley, and Michael Walter. Membership in moment polytopes is in NP and coNP. *SIAM J. Comput.*, 46(3):972–991, 2017. doi:10.1137/15M1048859.
- 8 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: geodesic first and second order methods for moment maps and polytopes. In *60th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2019*, pages 845–861. IEEE Computer Soc., Los Alamitos, CA, 2019. arXiv:1910.12375.
- 9 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In *59th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2018*, pages 883–897. IEEE Computer Soc., Los Alamitos, CA, 2018. doi:10.1109/FOCS.2018.00088.

- 10 Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. In *9th Innovations in Theoretical Computer Science*, volume 94 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 24, 20. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.
- 11 Peter Bürgisser, Yinan Li, Harold Nieuwboer, and Michael Walter. Interior-point methods for unconstrained geometric programming and scaling problems. arXiv, 2020. arXiv:2008.12110.
- 12 Michael B Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained Newton’s method and interior point methods. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS 2017)*, pages 902–913. IEEE, 2017. arXiv:1704.02310.
- 13 Stephen A. Cook. The complexity of theorem proving procedures. In *Proc. 3rd ACM STOC*, pages 151–158, 1971.
- 14 David A. Cox, John Little, and Donal O’Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997.
- 15 Harm Derksen. Polynomial bounds for rings of invariants. *Proc. Amer. Math. Soc.*, 129(4):955–963, 2001. doi:10.1090/S0002-9939-00-05698-7.
- 16 Harm Derksen. The graph isomorphism problem and approximate categories. *J. Symb. Comput.*, 59:81–112, 2013. doi:10.1016/j.jsc.2013.06.002.
- 17 Harm Derksen and Gregor Kemper. *Computational invariant theory*, volume 130 of *Encyclopedia of Mathematical Sciences*. Springer, Heidelberg, enlarged edition, 2015. With two appendices by Vladimir L. Popov, and an addendum by Norbert A’Campo and Popov, Invariant Theory and Algebraic Transformation Groups, VIII. doi:10.1007/978-3-662-48422-7.
- 18 Harm Derksen and Visu Makam. Generating invariant rings of quivers in arbitrary characteristic. *J. Algebra*, 489:435–445, 2017. doi:10.1016/j.jalgebra.2017.06.035.
- 19 Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. *Adv. Math.*, 310:44–63, 2017. doi:10.1016/j.aim.2017.01.018.
- 20 Harm Derksen and Visu Makam. Degree bounds for semi-invariant rings of quivers. *J. Pure Appl. Algebra*, 222(10):3282–3292, 2018. doi:10.1016/j.jpaa.2017.12.007.
- 21 Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *Algebra Number Theory*, 14(10):2791–2813, 2020. doi:10.2140/ant.2020.14.2791.
- 22 Harm Derksen and Visu Makam. An exponential lower bound for the degrees of invariants of cubic forms and tensor actions. *Adv. Math.*, 368:107136, 25, 2020. doi:10.1016/j.aim.2020.107136.
- 23 Igor Dolgachev. *Lectures on invariant theory*, volume 296 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2003. doi:10.1017/CB09780511615436.
- 24 Arnaud Durand, Miki Hermann, and Laurent Juban. On the complexity of recognizing the Hilbert basis of a linear Diophantine system. *Theoret. Comput. Sci.*, 270(1-2):625–642, 2002. doi:10.1016/S0304-3975(01)00017-2.
- 25 Kousha Etessami, Alistair Stewart, and Mihalis Yannakakis. A note on the complexity of comparing succinctly represented integers, with an application to maximum probability parsing. *ACM Trans. Comput. Theory*, 6(2):9:1–9:23, 2014. doi:10.1145/2601327.
- 26 Michael A. Forbes and Amir Shpilka. Explicit Noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, randomization, and combinatorial optimization*, volume 8096 of *Lecture Notes in Comput. Sci.*, pages 527–542. Springer, Heidelberg, 2013. doi:10.1007/978-3-642-40328-6_37.
- 27 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *57th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2016*, pages 109–117. IEEE Computer Soc., Los Alamitos, CA, 2016. doi:10.1109/FOCS.2016.95.

- 28 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Operator scaling: theory and applications. *Found. Comput. Math.*, 20(2):223–290, 2020. doi:10.1007/s10208-019-09417-z.
- 29 Ankit Garg, Christian Ikenmeyer, Visu Makam, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Search problems in algebraic complexity, GCT, and hardness of generators for invariant rings. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPICs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CCC.2020.12.
- 30 Guoqiang Ge. Testing equalities of multiplicative representations in polynomial time (extended abstract). In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 422–426. IEEE Computer Society, 1993. doi:10.1109/SFCS.1993.366845.
- 31 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993. doi:10.1007/978-3-642-78240-4.
- 32 Victor Guillemin and Shlomo Sternberg. *Symplectic techniques in physics*. Cambridge university press, 1990.
- 33 Leonid Gurvits. Classical complexity and quantum entanglement. *J. Comput. Syst. Sci.*, 69(3):448–484, 2004. doi:10.1016/j.jcss.2004.06.003.
- 34 Leonid Gurvits. Combinatorial and algorithmic aspects of hyperbolic polynomials. *arXiv preprint*, 2004. arXiv:math/0404474.
- 35 D. Hilbert. Über die vollen Invariantensysteme. *Math. Ann.*, 42:313–373, 1893. URL: <http://eudml.org/doc/157652>.
- 36 David Hilbert. Über die Theorie der algebraischen Formen. *Math. Ann.*, 36(4):473–534, 1890. doi:10.1007/BF01208503.
- 37 Evelynne Hubert and George Labahn. Scaling invariants and symmetry reduction of dynamical systems. *Foundations of Computational Mathematics*, 13, 2013. doi:10.1007/s10208-013-9165-9.
- 38 James E. Humphreys. *Linear algebraic groups*. Springer-Verlag, New York-Heidelberg, 1975. Graduate Texts in Mathematics, No. 21.
- 39 Christian Ikenmeyer, Ketan D Mulmuley, and Michael Walter. On vanishing of Kronecker coefficients. *Computational Complexity*, 26(4):949–992, 2017.
- 40 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds’ problem and matrix semi-invariants. *Comput. Complexity*, 26(3):717–763, 2017. doi:10.1007/s00037-016-0143-x.
- 41 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Comput. Complexity*, 27(4):561–593, 2018. doi:10.1007/s00037-018-0165-7.
- 42 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. doi:10.1007/s00037-004-0182-6.
- 43 Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979. doi:10.1137/0208040.
- 44 Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103, 1972.
- 45 George Kempf and Linda Ness. The length of vectors in representation spaces. In *Algebraic geometry (Proc. Summer Meeting, Univ. Copenhagen, Copenhagen, 1978)*, volume 732 of *Lecture Notes in Math.*, pages 233–243. Springer, Berlin, 1979.

- 46 Hanspeter Kraft. *Geometrische Methoden in der Invariantentheorie*. Aspects of Mathematics, D1. Friedr. Vieweg & Sohn, Braunschweig, 1984. doi:10.1007/978-3-322-83813-1.
- 47 David B Leep and Gerry Myerson. Marriage, magic, and solitaire. *The American Mathematical Monthly*, 106(5):419–429, 1999.
- 48 L. A. Levin. Universal enumeration problems. *Problemy Peredači Informacii*, 9(3):115–116, 1973.
- 49 Nathan Linial and Zur Luria. On the vertices of the d -dimensional Birkhoff polytope. *Discrete & Computational Geometry*, 51(1):161–170, 2014.
- 50 Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20(4):545–568, 2000.
- 51 Visu Makam and Avi Wigderson. Singular tuples of matrices is not a null cone (and, the symmetries of algebraic varieties). *CoRR*, abs/1909.00857, 2019. arXiv:1909.00857.
- 52 Ketan D. Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *J. Amer. Math. Soc.*, 30(1):225–309, 2017. doi:10.1090/jams/864.
- 53 Ketan D Mulmuley and Milind Sohoni. Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM Journal on Computing*, 31(2):496–526, 2001.
- 54 David Mumford. *The red book of varieties and schemes*, volume 1358 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1988. doi:10.1007/978-3-662-21581-4.
- 55 David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory, Third Edition*, volume 34 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, 1994.
- 56 Vladimir L Popov. Two orbits: When is one in the closure of the other? *Proceedings of the Steklov Institute of Mathematics*, 264(1):146–158, 2009.
- 57 Alexander Schrijver. *Theory of linear and integer programming*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Ltd., Chichester, 1986. A Wiley-Interscience Publication.
- 58 R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35:876–879, 1964.
- 59 Henry J. Stephen Smith. On systems of linear indeterminate equations and congruences. *Philosophical Transactions of the Royal Society of London*, 151:293–326, 1861. URL: <http://www.jstor.org/stable/108738>.
- 60 Bernd Sturmfels. *Algorithms in Invariant Theory*. Texts & Monographs in Symbolic Computation. Springer, 2008. doi:10.1007/978-3-211-77417-5.
- 61 David Wehlau. Constructive invariant theory for tori. *Annales de l'institut Fourier*, 43(4):1055–1066, 1993. URL: <http://eudml.org/doc/75025>.
- 62 Hermann Weyl. *The Classical Groups. Their Invariants and Representations*. Princeton University Press, Princeton, N.J., 1939.