J Algebr Comb (2012) 36:103–110
DOI 10.1007/s10801-011-0325-1

Geometric complexity theory III: on deciding nonvanishing of a Littlewood–Richardson coefficient

Ketan D. Mulmuley · Hariharan Narayanan · Milind Sohoni

Received: 9 October 2010 / Accepted: 14 October 2011 / Published online: 4 November 2011

Abstract We point out that the positivity of a Littlewood–Richardson coefficient $c^{\gamma}_{\alpha,\beta}$ for sl_n can be decided in strongly polynomial time. This means that the number of arithmetic operations is polynomial in n and independent of the bit lengths of the specifications of the partitions α, β, and γ, and each operation involves numbers whose bitlength is polynomial in n and the bit lengths α, β, and γ.

Secondly, we observe that nonvanishing of a generalized Littlewood–Richardson coefficient of any type can be decided in strongly polynomial time assuming an analogue of the saturation conjecture for these types, and that for weights α, β, γ, the positivity of $c^{2\gamma}_{2\alpha,2\beta}$ can (unconditionally) be decided in strongly polynomial time.

Keywords Littlewood–Richardon coefficients · Geometric complexity theory · Algorithms

Dedicated to Sri Ramakrishna.

K.D. Mulmuley
Department of Computer Science, University of Chicago, 1100 E 58th Street, Chicago, IL 60637, USA
e-mail: mulmuley@cs.uchicago.edu

H. Narayanan (✉)
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
e-mail: har@mit.edu

M. Sohoni
Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Mumbai 400 076, India
e-mail: sohoni@cse.iitb.ac.in

1 Introduction

The fundamental Littlewood–Richardson rule in representation theory [4] states that the tensor product of two irreducible representations (Weyl modules) V_α and V_β of a complex semisimple Lie algebra $\mathcal{G}$ decomposes as follows:

$$V_\alpha \otimes V_\beta = \bigoplus_\gamma C^\gamma_{\alpha,\beta} V_\gamma, \tag{1}$$

where $C^\gamma_{\alpha,\beta}$ are generalized Littlewood–Richardson coefficients. Here α, β, and γ denote highest weights of $\mathcal{G}$. When $\mathcal{G} = sl_n(\mathbb{C})$ (type A), α and β are partitions (Young diagrams) with at most n rows, and the sum is over all Young diagrams γ of height at most n and size equal to the sum of the sizes of α and β.

We are interested in finding an efficient algorithm to decide if $C^\gamma_{\alpha,\beta}$ is nonvanishing (positive). As will be explained below (Sect. 1.1), this problem arises naturally in the geometric complexity theory approach [14–16] toward the P vs. NP and related problems.

It has been observed in [11, 12, 17] independently that when $\mathcal{G} = sl_n(\mathbb{C})$ (type A), nonvanishing of $C^\gamma_{\alpha,\beta}$ can be decided in polynomial time; i.e., in time that is polynomial in n and the bitlengths of the specifications of the partitions α, β, and γ. Furthermore, the algorithm in [17] is strongly polynomial in the sense of [13]. We say that an algorithm is strongly polynomial if the number of arithmetic steps in the algorithm is polynomial in the number of input parameters, independent of their total bitlength, and the bit length of each intermediate operand that arises in the algorithm is polynomial in the total bitlength of the input. We shall also use slight variants of this definition as per the problem. In particular, strong polynomiality here means: (1) the number of arithmetic steps in the algorithm is polynomial in n. It does not depend on the bit lengths of α_i, β_j, and γ_k's. (2) The bitlength of every intermediate operand that arises in the algorithm is polynomial in n and the total bitlength of α, β, and γ. One crucial ingredient in this algorithm is the saturation theorem in [10], which does not hold for simple Lie algebras of type B, C, or D [24]. The result in [17] was extended to other types in [18] assuming a positivity conjecture in [12]. This article combines the results of [17] and [18]. We now state these results in more detail.

First, we consider type A, i.e., when $\mathcal{G} = sl_n(\mathbb{C})$. Let $\lambda = (\lambda_1, \ldots, \lambda_n)$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$, be a partition (Young diagram); here each λ_i is integral, and $\lambda_i = 0$ for i higher than the height of λ. By the bitlength $\langle\lambda\rangle$ of λ we mean the total bitlength of its specification.

Theorem 1.1 *Given partitions α, β, and γ, whether $C^\gamma_{\alpha,\beta}$ is positive can be decided in polynomial time and, in fact, in strongly polynomial time.*

For general types, we have:

Theorem 1.2 *The positivity of a generalized Littlewood–Richardson coefficient $C^\nu_{\lambda,\mu}$ for any complex semisimple Lie algebra $\mathcal{G}$ can be decided in strongly polynomial time, assuming the following positivity conjecture made in* [12].

Let $\tilde{C}(n) = \tilde{C}^{\mu}_{\lambda,\mu}(n) = C^{n\nu}_{n\lambda,n\mu}$ denote the stretching function associated with $C^{\nu}_{\lambda,\mu}$. Assume that the type of $\mathcal{G}$ is B, C, or D. Then $\tilde{C}(n)$ is a quasi-polynomial of period at most two [12]. That is, there exist polynomials $\tilde{C}_1(n)$ and $\tilde{C}_2(n)$ such that

$$C^{n\nu}_{n\lambda,n\mu} = \begin{cases} \tilde{C}_1(n) & \text{if } n \text{ is odd;} \\ \tilde{C}_2(n) & \text{if } n \text{ is even.} \end{cases}$$

Conjecture 1.3 (Positivity conjecture) [12] *The quasi-polynomial* $\tilde{C}(n) = \tilde{C}^{\nu}_{\lambda,\mu}(n)$ *is positive, i.e., the coefficients of* $\tilde{C}_i(n)$, $i = 1, 2$, *are nonnegative.*

This is an extension of an analogous earlier conjecture in [9] for type A. Considerable experimental evidence for these conjectures has been given in these papers.

Here it is assumed that each highest weight is specified by giving its coordinates in the basis of fundamental weights. The bitlength $\langle\lambda\rangle$ is defined to be the total bitlength of all coordinates.

Remark 1.4 For Theorem 1.2 to hold, we do not need the full statement of the Positivity Conjecture, but only the following analogue of saturation for Lie groups of types B, C, D:

$$C^{\nu}_{\lambda\mu} = 0 \quad \Longrightarrow \quad \forall \text{ odd } n, \quad C^{n\nu}_{n\lambda,n\mu} = 0.$$

In fact, the following weaker hypothesis suffices: A generalized Littlewood–Richardson coefficient is nonzero if the affine span of the corresponding BZ-polytope [1] contains an integer point.

Finally, we observe that the proof of Theorem 1.1 can be extended to general types using the recent results in [2, 20]:

Theorem 1.5 *The positivity of a generalized Littlewood–Richardson coefficient* $C^{2\nu}_{2\lambda,2\mu}$ *for any complex semisimple Lie algebra* $\mathcal{G}$ *can be decided in strongly polynomial time.*

1.1 Significance in geometric complexity theory

Now we explain the significance of the results in this paper in the context of the geometric complexity theory (GCT) [14–16] approach to the permanent vs. determinant problem [23], an algebraic prototype of the P vs. NP problem.

The problem is to show that $\mathrm{perm}(X)$, the permanent of the $n \times n$ variable matrix, cannot be represented as the determinant of an $m \times m$ matrix Y whose each entry is a rational (or complex) affine combination of the entries of X, if $m = \mathrm{poly}(n)$. The articles [15, 16] reduce this problem to the problem of proving the existence of a *geometric obstruction* (proof certificate of hardness) when $m = \mathrm{poly}(n)$. A geometric obstruction is a Weyl module $V_\lambda(G)$, $G = GL_{m^2}(\mathbb{C})$, that occurs as a G-subrepresentation of the (dual of the) homogeneous coordinate ring of a certain G-variety $\Delta[\mathrm{perm}, n, m]$ associated with the permanent but not in the (dual of the)

homogeneous coordinate ring of a similar G-variety $\Delta[\det, m]$ associated with the determinant. The goal (cf. [19]) is to construct the labeling partition λ of some geometric obstruction *explicitly* in $\text{poly}(n, m)$ time when $m = \text{poly}(n)$. The reason we are going for explicit construction, when proving the existence of an obstruction even nonconstructively suffices in principle, is the Flip theorem (cf. Theorems 2.3 and 4.6 in [14]), which says that we are essentially *forced* to construct some proof certificate of hardness explicitly in the problem under consideration.

In more detail, the explicit proof strategy, called the flip [14, 19] is the following:

(1) [Verification]: Find a (strongly) polynomial-time algorithm for verifying if a given λ is a geometric obstruction label for given n and m.
(2) [Discovery]: Use this efficient verification criterion to decide if a geometric obstruction exists for given n and m in $\text{poly}(n, m)$ time. If it does, construct one such geometric obstruction label explicitly in the same time.
(3) [Program correctness]: Show that the discovery algorithm in (2) always succeeds if $m = \text{poly}(n)$.

The upper bound problems in algebraic geometry and representation theory that arise in (1) and (2) seem well beyond the reach of the existing techniques. The article [19], in conjunction with [15, 16], describes an approach to these problems based on some intermediate upper bound problems in representation theory, such as the problem of deciding nonvanishing of the Kronecker coefficients [4] in (strongly) polynomial time. Since the Littlewood–Richardson coefficient is a special case of the Kronecker coefficient, the problem of deciding its nonvanishing is a basic prototype of the far harder decision problems in algebraic geometry and representation theory that arise in GCT. This is the main motivation for studying this problem in this paper. As explained in [14], the (strongly) polynomial time algorithm for this problem in this paper serves as a basic prototype for the approach in [19] to the harder upper bound problems in algebraic geometry and representation theory that arise in GCT.

We should also explain why we are going for strongly polynomial-time algorithms instead of just polynomial-time algorithms. This too is motivated by the (strong) Flip theorem (cf. Theorem 4.6 in [14]). Assuming the permanent vs. determinant conjecture and an additional derandomization hypothesis in complexity theory, this result provides a strongly polynomial-time algorithm for generating counterexamples in the context of this conjecture. Specifically, given any $m \times m$ matrix Y, whose entries are rational affine combinations of the entries of the $n \times n$ variable matrix X, $m = \text{poly}(n)$, this algorithm generates in strongly polynomial time a set $\{X_0, \ldots, X_a\}$ of $a = O(1)$ inputs such that the value of $\text{perm}(X)$ on some input X_i in this set is different from the value of $\det(Y)$ at X_i. The strong Flip theorem (in its full stronger form) is the crux of the justification in [19] for why there should exist (strongly) polynomial-time algorithms for the various decision problems in algebraic geometry and representation theory that arise in GCT and why this existence may, in essence, be implication of the various hardness and derandomization conjectures in complexity theory. This is why we are going toward strongly polynomial-time algorithms right from the beginning.

There is also a pragmatic reason for this. For the step (3) above to succeed, it is not enough if the algorithms in the steps (1) and (2) are just efficient in theory.

They also have to be simple enough so that they can be used to carry out the step (3) successfully. The notion of a strongly polynomial-time algorithm was proposed [13] in complexity theory to develop polynomial-time algorithms that are simple in practice. As was pointed out in [13], most problems that have strongly polynomial-time algorithms eventually turn out to have strongly polynomial-time algorithms that are combinatorial in nature, simple in practice, and do not depend on complicated numerical procedures such as linear programming. In the same spirit, the strategy here is to find strongly polynomial-time algorithms for the various decision problems that arise in GCT first and eventually find simpler combinatorial algorithms (not based on linear programming) that can be used to carry out the step (3) successfully.

Since this paper gives a strongly polynomial-time algorithm for deciding nonvanishing of Littlewood–Richardson coefficient, we expect, as per this general paradigm, a simpler combinatorial strongly polynomial-time algorithm that does not use linear programming. The recent article [3] takes an important step in this direction. Motivated by the result in this article, it gives a polynomial-time combinatorial algorithm based on max-flows for deciding nonvanishing of Littlewood–Richardson coefficients in type A. But it is still not strongly polynomial.

The rest of this article is organized as follows. Theorem 1.1 is proved in Sect. 2, Theorem 1.2 in Sect. 3 and Theorem 1.5 in Sect. 4.

2 Littlewood–Richardson coefficient of type A

Here we prove Theorem 1.1. The proof follows easily from the following three results:

1. Littlewood–Richardson rule: specifically, a polyhedral interpretation of the Littlewood–Richardson coefficients based on the Hive polytope [10]; one could also use the BZ-polytope [1] instead.
2. Saturation Theorem [10].
3. Polynomial-time algorithm for linear programming, e.g., the ellipsoid or the interior point method and the related strongly polynomial-time algorithm for combinatorial linear programming in [13].

As per the polyhedral interpretation of the Littlewood–Richardson rule in [10], the Littlewood–Richardson coefficient $C^{\gamma}_{\alpha,\beta}$ can be expressed as the number of integer points in a hive polytope $P = P^{\gamma}_{\alpha,\beta}$ which can be specified by an explicit linear program of the form

$$Ax \leq b \tag{2}$$

such that: (1) Given α, β, and γ, this specification can be computed in strongly polynomial time. (2) The linear program is combinatorial in the terminology of [22]. This means the entries of A have constant bit lengths (in fact, they are just $0, 1$, or -1). (3) The entries of b are homogeneous integral linear forms in α_i, β_j, and γ_k's.

Claim 2.1 *The polytope P contains an integer point iff it is nonempty.*

Proof One direction is trivial.

Suppose P is nonempty. Since b is homogeneous in α, β, and γ, it follows that, for any positive integer q, $C^{q\gamma}_{q\alpha,q\beta}$ is the number of integer points in the scaled polytope qP. All vertices of P have rational coefficients. Hence, for some positive integer q, the scaled polytope qP has an integer point. It follows that, for this q, $C^{q\gamma}_{q\alpha,q\beta}$ is positive. Saturation Theorem [10] says that, in this case, $C^{\gamma}_{\alpha,\beta}$ is positive. Hence, P contains an integer point. □

Whether P is nonempty can be determined in polynomial time using either the ellipsoid or the interior point algorithm for linear programming. Since the linear program (2) is combinatorial, this can also be done in strongly polynomial time using the combinatorial linear programming algorithm in [22]. This proves Theorem 1.1.

3 Generalized Littlewood–Richardson coefficients

In this section we prove Theorem 1.2.

Let $P = P^{\nu}_{\lambda,\mu}$ denote the BZ-polytope [1] whose Ehrhart quasi-polynomial coincides with $\tilde{C}^{\nu}_{\lambda,\mu}(n)$.

Definition 3.1 For any subset B of $\mathbb{Q}^n$, its affine span over rationals, $\mathrm{Aff}(B)$, is

$$\left\{ v \,\middle|\, \exists\big(\{v_i\} \subseteq B, \{\alpha_i\} \subseteq \mathbb{Q}\big) \text{ such that } \sum_{i=1}^{n} \alpha_i = 1 \text{ and } v = \sum_{i=1}^{n} \alpha_i v_i \right\}.$$

Let $\mathbb{Z}_{\langle 2\rangle}$ denote the subring of $\mathbb{Q}$ obtained by localizing $\mathbb{Z}$ at 2, i.e., the subring of fractions with odd denominators. We will call a point in $\mathbb{R}^d$ rational if all its coordinates are rational.

Lemma 3.2 *Assume that $\mathcal{G}$ is simple of type B, C, or D. If the positivity conjecture is true, the following are equivalent*:

(1) $C^{\nu}_{\lambda\mu} \geq 1$.
(2) *There exists an odd integer n such that* $C^{n\nu}_{n\lambda\, n\mu} \geq 1$.
(3) *P contains a point in* $\mathbb{Z}^d_{\langle 2\rangle}$.
(4) $\mathrm{Aff}(P)$ *contains a point in* $\mathbb{Z}^d_{\langle 2\rangle}$.

Proof Clearly, the first three statements are equivalent, and (3) implies (4). It remains to show that (4) implies (3). Let $z \in \mathbb{Z}^d_{\langle 2\rangle} \cap \mathrm{Aff}(P)$.

The 0-dimensional case is trivial since $\{z\} = P$. Suppose that the dimension of P is greater than or equal to 1. Since z has rational coordinates and is contained in $\mathrm{Aff}(P)$, $z = ax + (1-a)y$ for some distinct rational points $x, y \in P$, and $a \in \mathbb{Q}$. Let q be a positive integer such that $2^q(x-y) \in \mathbb{Z}^d_{\langle 2\rangle}$.

The set $\{z + \lambda 2^q(x-y) \mid \lambda \in \mathbb{Z}_{\langle 2\rangle}\}$ is a dense subset of $\mathrm{Aff}(\{x, y\})$ in the topology induced by the standard topology of $\mathbb{R}^n$ and is therefore nonempty. Thus, $P \cap \mathbb{Z}^d_{\langle 2\rangle} \cap \{z + \lambda 2^q(x-y) \mid \lambda \in \mathbb{Z}_{\langle 2\rangle}\} \neq \emptyset$. □

Now we turn to the proof of Theorem 1.2. First, let us assume that $\mathcal{G}$ is simple of type B, C, or D.

The specification of an explicit linear program of the form $Ax \leq b$ defining the BZ-polytope $P = P^{\nu}_{\lambda,\mu}$ can be computed in strongly polynomial time using its description in [1]. It is also clear from [1] that the entries of A here have constant bitlengths. In the terminology of [22], this linear program is combinatorial. Hence, we can determine if P is nonempty in strongly polynomial time by the combinatorial linear programming algorithm in [22]. If P is nonempty, this algorithm can also be extended to find an integral matrix C and an integral vector D so that $\mathrm{Aff}(P)$ is defined by the linear system $Cx = D$. One way of achieving this is the following. Find, for every constraint hyperplane h of P, a vertex v_h of P that is the farthest to h. The affine span is the intersection of all constraint hyperplanes h such that $v_h \in h$. Usual linear programming algorithms [7, 8] here, in place of the algorithm in [22], will yield a polynomial-time algorithm, instead of a strongly polynomial-time algorithm.

By Lemma 3.2 (4), it remains to check if $\mathrm{Aff}(P)$ contains a point in $\mathbb{Z}^d_{\langle 2\rangle}$. This can be done as follows. By padding, if necessary, we can assume that C is square. Using [5], we find the Smith normal form S of C and unimodular matrices U and V such that $C = USV$; here S is a diagonal integer matrix whose ith diagonal entry divides the $(i+1)$st diagonal entry. Since the entries of C have constant bitlengths, the algorithm in [5] works in strongly polynomial time. The question now reduces to checking if $USVx = D$ has a solution $x \in \mathbb{Z}^d_{\langle 2\rangle}$. This is so iff $Sy = U^{-1}D$ has a solution $y \in \mathbb{Z}^d_{\langle 2\rangle}$. Since S is diagonal, this can be verified in (strongly) polynomial time by checking each coordinate.

This proves Theorem 1.2 for types B, C, D.

Now let $\mathcal{G}$ be any semisimple algebra. A generalized Littlewood–Richardson coefficient for $\mathcal{G}$ is the product of corresponding generalized Littlewood–Richardson coefficients for each of its simple factors. Hence, without loss of generality, we can assume that $\mathcal{G}$ is simple. If it is of type A, then Theorem 1.2 holds unconditionally by Theorem 1.1. If it is an exceptional simple Lie algebra, then a Littlewood–Richardson coefficient can be computed in $O(1)$ arithmetic steps. This is because, when the rank of $\mathcal{G}$ is constant, the chambers of quasi-polynomiality [21] of the generalized Littlewood–Richardson coefficient, considered as a vector partition function, are generated by $O(1)$ constraints.

This proves Theorem 1.2.

4 Proof of Theorem 1.5

Suppose first that $\mathcal{G}$ is of type B, C, or D. By [2] and [20], it follows that if there exists an integer n such that $C^{n\nu}_{n\lambda,n\mu} \geq 1$, then $C^{2\nu}_{2\lambda,2\mu} \geq 1$. A weaker form of this result (with 4 in place of 2) was proven in [6]. By the argument in Sect. 3, $C^{2\nu}_{2\lambda,2\mu} \geq 1$ if and only if the BZ-polytope $P^{\nu}_{\lambda,\mu}$ is nonempty, which can be checked in strongly polynomial time. The argument toward the end of Sect. 3 allows the algorithm to be extended to arbitrary semisimple groups.

Acknowledgements We are grateful to T. McAllister for bringing the positivity conjecture in [12] to our attention and to Apoorva Khare for helpful discussions.

K.D. Mulmuley was supported by NSF grant CCF-1017760.

References

1. Berenstein, A.D., Zelevinsky, A.V.: Tensor product multiplicities, canonical bases and totally positive varieties. Invent. Math. **143**(1), 77–128 (2001)
2. Belkale, P., Kumar, S.: Eigencone, saturation and Horn problems for symplectic and odd orthogonal groups. J. Algebr. Geom. **19**, 199–242 (2010)
3. Bürgisser, P., Ikenmeyer, C.: A max-flow algorithm for positivity of Littlewood–Richardson coefficients. In: FPSAC (2009)
4. Fulton, W., Harris, J.: Representation Theory. Springer, Berlin (1991)
5. Kannan, R., Bachem, A.: Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. SIAM J. Comput. **8**(4), 499–507 (1979)
6. Kapovich, M., Millson, J.: A path model for geodesics in Euclidean buildings and its applications to representation theory. Groups, Geom. Dyn. **2**, 405–480 (2008)
7. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4), 373–395 (1984)
8. Khachian, L.G.: A polynomial algorithm for linear programming. Dokl. Akad. Nauk SSSR **244**, 1093–1096 (1979). (In Russian)
9. King, R.C., Tollu, C., Toumazet, F.: Stretched Littlewood–Richardson polynomials and Kostka coefficients. In: CRM Proceedings and Lecture Notes, vol. 34 (2003)
10. Knutson, A., Tao, T.: The honeycomb model of $GL_n(C)$ tensor products I: proof of the saturation conjecture. J. Am. Math. Soc. **12**, 1055–1090 (1999)
11. Knutson, A., Tao, T.: Honeycombs and sums of Hermitian matrices. Not. Am. Math. Soc. **48**(2), 175–186 (2001)
12. De Loera, J., McAllister, T.: On the computation of Clebsch–Gordan coefficients and the dilation effect. Exp. Math. **15**(1), 7–20 (2006)
13. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1993)
14. Mulmuley, K.: On P vs. *NP* and geometric complexity theory. JACM **58**(2) (2011)
15. Mulmuley, K., Sohoni, M.: Geometric complexity theory: an approach to the P vs. *NP* and related problems. SIAM J. Comput. **31**(2), 496–526 (2001)
16. Mulmuley, K., Sohoni, M.: Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties. SIAM J. Comput. **38**(3) (2008)
17. Mulmuley, K., Sohoni, M.: Geometric complexity theory III, on deciding positivity of Littlewood–Richardson coefficients. Preprint arXiv:cs.CC/0501076v1 (26 January 2005)
18. Mulmuley, K., Narayanan, H.: Geometric complexity theory V, on deciding non-vanishing of a generalized Littlewood–Richardson coefficients. Technical Report TR-2007-05, Computer Science Department, University of Chicago (April 2007)
19. Mulmuley, K.: Geometric complexity theory VI: the flip via positivity. Technical Report, Computer Science Department, The University of Chicago (January 2011). Available at: http://ramakrishnadas.cs.uchicago.edu
20. Sam, S.: Symmetric quivers, invariant theory, and saturation theorems for the classical groups. arXiv:1009.3040 (September, 2010)
21. Sturmfels, B.: On vector partition functions. J. Comb. Theory Ser. A **72**(2) (1995)
22. Tardos, É.: A strongly polynomial algorithm to solve combinatorial linear programs. Oper. Res. **34**, 250–256 (1986)
23. Valiant, L.: The complexity of computing the permanent. Theor. Comput. Sci. **8**, 189–201 (1979)
24. Zelevinsky, A.: Littlewood–Richardson semigroups, new perspectives in geometric combinatorics. MSRI Publ. **38**, 337–345 (1999)