Computational Aspects of Root Systems, Coxeter Groups, and Weyl Characters

John R. Stembridge

15 Aug 2001 (Final Edition?)

Contents

Partially supported by NSF Grant DMS–9700787 and RIMS, Kyoto University.

I would like to thank RIMS for their kind hospitality during the preparation of this article.

This version and previous versions dated 20 Apr 2000 and 5 Feb 2001 contain only minor revisions from the original version of 9 Nov 1998.

0. Introduction

In this article, our goal is to survey some of the fundamental computational problems that arise in working with the structures mentioned in the title. We became interested in these problems in the course of trying to gather data (and prove theorems) involving the exceptional groups and their root systems, and this in turn led us to the ongoing development of the Maple packages *coxeter* and *weyl*.

For the classical cases, especially type A, many of these problems are easy or have well-known solutions. However these solutions often do not generalize. Here our emphasis is on algorithms that are (for the most part) independent of the classification of root systems.

The canonical example we always have in mind is E_8.

We should remark that there are many researchers elsewhere who have also developed software for these and similar problems; for example, there is the *LiE* package of van Leeuwen, Cohen and Lisser (et. al.), the *CHEVIE* package for GAP and Maple by Meinolf Geck (et. al.), and the *Schur* package of Brian Wybourne. Web links to these packages can be found at the end of the article.

Throughout, Φ will denote a finite crystallographic root system of rank n embedded in a real Euclidean space V with inner product $\langle\ ,\ \rangle$. We let $\alpha_1, \ldots, \alpha_n$ denote a collection of simple roots, with Φ^+ the corresponding set of positive roots. For any root α, we write $\alpha^\vee := 2\alpha/\langle\alpha, \alpha\rangle$ for the corresponding co-root. We will assume that the reader is familiar with the basic terminology of root systems and reflection groups, as well as the classification of root systems by Dynkin diagrams.

Standard references are [**Bo**] and [**H1-2**].

The crystallographic hypothesis is unnecessary for much of what we discuss in §§1–4, however it introduces unpleasant computational details (e.g., the need for floating-point or exact number field arithmetic) that would distract us from the main issues. In §4 we will temporarily relax the assumption of finiteness.

For nonzero $\alpha \in V$, we let $\sigma_\alpha \in GL(V)$ denote the corresponding reflection; i.e.,

$$\sigma_\alpha(\lambda) = \lambda - \langle \lambda, \alpha^\vee \rangle \alpha \qquad (\lambda \in V). \tag{0.1}$$

The Weyl group W corresponding to Φ is the (finite) group generated by the reflections $\sigma_{\alpha_1}, \ldots, \sigma_{\alpha_n}$ corresponding to the simple roots. For brevity, let $s_i = \sigma_{\alpha_i}$.

It is well-known that W is a Coxeter group; i.e., the relations

$$(s_i s_j)^{m(i,j)} = 1, \tag{0.2}$$

where $m(i,j)$ denotes the order of $s_i s_j$ in W, define a presentation of W.

1. Reduced Words

In most cases, the preferred data structure we use for representing Weyl group elements are words (integer lists) that encode products of simple reflections. (An alternative is discussed in the following section.) Thus the word $(i_1, \ldots, i_l)$ encodes the group element $w = s_{i_1} \cdots s_{i_l}$. In these terms, group multiplication is concatenation, and group inversion is reversal. Of course all that this does is to move the real problem elsewhere. For this data structure, the problem is to decide when two words encode the same group element, or to produce a canonical (minimum-length) representative of a given group element.

While it possible to solve these word problems using only the Coxeter relations (0.2) (or the braid relations), there are much faster and simpler solutions available that take advantage of the geometrical tools provided by the root system. It is well-known that the hyperplanes

$$\alpha^\perp = \{\lambda \in V : \langle \lambda, \alpha \rangle = 0\} \qquad (\alpha \in \Phi)$$

are stable under the action of W and their removal from V partitions the remainder into connected components (chambers).

The action of W on chambers is simply transitive. Thus if $\lambda \in V$ is any vector in general position (i.e., not orthogonal to any root), then the words $(i_1, \ldots, i_l)$ and $(j_1, \ldots, j_m)$ encode the same group element if and only if

$$s_{i_1} \cdots s_{i_l}(\lambda) = s_{j_1} \cdots s_{j_m}(\lambda).$$

The cost of such a computation amounts to $l + m$ vector additions, scalar multiplications, and scalar products (cf. (0.1)). However, we should point out that the real cost is usually far less than would be incurred if the vectors involved were randomly distributed. Indeed, in the standard realization of every crystallographic root system, many of the roots (in some cases, all) have only one or two nonzero coordinates relative to some orthonormal basis. If the code for performing vector operations is written to take advantage of this sparsity, then the real cost of a vector operation involving a root is (often) the same as the cost of one or two scalar operations.

The minimum length among all expressions for $w \in W$ is denoted $\ell(w)$.

To determine a canonical representation for the group element indexed by the word $(i_1, \ldots, i_l)$, one may make use of the fact (e.g., see [**H2**, §5.4]) that

$$\ell(s_i w) < \ell(w) \Leftrightarrow w^{-1}\alpha_i \in -\Phi^+.$$

Indeed, it follows that if λ is any point in the fundamental chamber; i.e.,

$$\langle \lambda, \alpha_i \rangle > 0 \qquad (1 \leqslant i \leqslant n),$$

then

$$\ell(s_i w) < \ell(w) \Leftrightarrow \langle w\lambda, \alpha_i \rangle < 0. \tag{1.1}$$

In other words, w has a minimum-length expression that begins with s_i if and only if $\langle w\lambda, \alpha_i \rangle < 0$. Therefore, we can determine the lexicographically first minimal expression for w by first computing $\mu := w\lambda$ (using any representation of w as a product of simple reflections), and then starting with the empty word,

1. Find the least index i such that $\langle \mu, \alpha_i \rangle < 0$.
2. Append i to the word being constructed.
3. Replace $\mu \leftarrow s_i\mu$, and repeat.

The algorithm terminates when μ reaches the fundamental chamber.

2. Permutation Representations

In some cases, it is preferable to use permutation representations of Weyl groups, rather than reduced words. For example, this allows one to take advantage of the extensive library of group-theoretic tools (available in *GAP*, for example) that have been developed over many years by the computational group theory community.

A basic issue that arises is the problem of converting between the two ways of representing group elements. One direction is trivial. If we have permutations $\pi_1, \dots, \pi_n$ representing the action of the simple reflections, it is easy to determine the permutation that corresponds to the group element encoded by the word $(i_1, \dots, i_l)$. The inverse problem is more significant.

PROBLEM 2.1. *Given permutations $\sigma, \pi_1, \dots, \pi_n$ of some finite set, find (if possible) an expression for σ of the form*

$$\sigma = \pi_{i_1}^{\pm 1} \cdots \pi_{i_l}^{\pm 1}.$$

This is one of the fundamental problems of computational group theory[1] and fortunately there are good, polynomial-time algorithms for it that are based on building a strong generating set in the sense of Sims [**S**].

For Weyl groups, one can use the geometry of root systems to quickly build a strong generating set, much faster than is possible for general permutation groups.

First, we need to construct a permutation representation of the Weyl group W. The natural way to do this is to let W act on cosets of some subgroup. The most convenient available subgroups are the so-called parabolic subgroups, the subgroups generated by subsets of the simple reflections. Given any subset $J \subseteq \{1, \dots, n\}$, we let W_J denote the parabolic subgroup of W generated by $\{s_i : i \in J\}$.

The coset space W/W_J has a geometric representation as the orbit of a suitably chosen point $\lambda \in V$. Indeed, the stabilizer subgroup of every point in V is generated by reflections, and the stabilizer of every point in the closure of the fundamental chamber is generated by *simple* reflections (e.g., [**H2**, §1.12]). Thus by selecting λ so that

$$\langle \lambda, \alpha_i \rangle = 0 \quad (i \in J); \quad \langle \lambda, \alpha_i \rangle > 0 \quad (i \notin J), \tag{2.1}$$

we obtain a vector whose stabilizer is W_J, and the map $W/W_J \to W\lambda$ given by

$$wW_J \mapsto w\lambda$$

is an isomorphism of sets-with-W-action. It is now easy to determine the permutations $\pi_1, \dots, \pi_n$ that represent the action of the simple reflections on W/W_J. Simply construct the orbit of λ by starting with $\mathcal{O} = \{\lambda\}$ and then successively add new members to $\mathcal{O}$ until it is saturated under the action of $s_1, \dots, s_n$. The permutation π_i can then be obtained by examining the action of s_i on $\mathcal{O}$.

We now have some permutation representations available, but are they faithful?

[1] It is also the key to novelties such as Rubik's cube.

A_n	B_n, C_n	D_n	E_6	E_7	E_8	F_4	G_2
$n+1$	$2n$	$2n$	27	56	240	24	6

Table I. Degrees of permutation representations.

Proposition 2.2. *The permutation representation of W on W/W_J is faithful if and only if J omits at least one node from each connected component of the diagram of Φ. In particular, if W is irreducible, the representation is faithful if and only if W_J is a proper subgroup.*

Proof. Since W is the direct product of its irreducible components, it suffices to restrict our attention to the irreducible case. Clearly, the hypothesis that W_J is a proper subgroup is necessary. Conversely, suppose $j \notin J$ and let $w \in W$ be an element that acts trivially on W/W_J. Given λ as described in (2.1), w must stabilize every vector $x\lambda$ with $x \in W$. Hence w stabilizes $x\lambda - xs_j\lambda = \langle \lambda, \alpha_j^\vee \rangle x\alpha_j$. However $\langle \lambda, \alpha_j \rangle > 0$, so w stabilizes every $x\alpha_j$; i.e., w fixes every point in the W-orbit of α_j. Since the diagram of Φ is assumed to be connected, this means that w stabilizes every simple root. (If α_1 is in the span of $W\alpha_j$ and α_2 is adjacent to α_1 in the diagram, then $s_2\alpha_1 = \alpha_1 + c\alpha_2$ for some nonzero scalar c, so α_2 is also in the span of $W\alpha_j$.) Hence w acts as the identity map on V. □

In Table I, we list the degrees of the smallest (faithful) parabolic permutation representations in the irreducible cases.

Let us now turn to strong generating sets and the solution of Problem 2.1. For a more comprehensive account, see [**BLS**] and the references cited there.

Let $S = \{\pi_1, \dots, \pi_n\}$ be a collection of permutations of some finite set X, and let G be the permutation group generated by S. A *stabilizer chain* for G is a sequence of subgroups

$$G = G_0 \supset G_1 \supset \cdots \supset G_l = \{1\},$$

such that G_i is the stabilizer in G_{i-1} of some point $x_i \in X$. The set of points $B = \{x_1, \dots, x_l\}$ is called a *base*. For $1 \leqslant i \leqslant l$, let S_i be a set of coset representatives for G_{i-1}/G_i, so that

$$G_{i-1} = \bigcup_{\sigma \in S_i} \sigma G_i.$$

Notice that each $\pi \in G$ has a unique representation of the form

$$\pi = \sigma_1\sigma_2 \cdots \sigma_l \qquad (\sigma_i \in S_i), \tag{2.2}$$

so in particular $S_1 \cup \cdots \cup S_l$ generates G; it is called a *strong generating set*.

Once a strong generating set has been found, finding the factorization (2.2) is rapid. Indeed, since G_i is the stabilizer of x_i in G_{i-1}, there is a one-to-one correspondence between S_i and the G_{i-1}-orbit of x_i. We can even label the members of S_i by the points in this orbit so that the representative indexed by x is the unique coset representative that maps x_i to x. Hence, to determine the factorization (2.2), we first compute $x = \pi(x_1)$. The first factor σ_1 must be the coset representative in S_1 indexed by x. If there is no such representative, this constitutes proof that $\pi \notin G$. Otherwise, we replace $\pi \leftarrow \sigma_1^{-1}\pi$ and recursively determine the rest of the factorization.

In order to solve Problem 2.1 by this mechanism, we need to know how to express each of the strong generators as words built out of the original generating set S. For the first "layer" of the strong generating set S_1, this is easy. Having chosen x_1, we construct the G-orbit of x_1 by the saturation method mentioned earlier in this section, the only difference being that we also keep track of the (first) word in the generators that allows us to reach each point in the orbit. This builds S_1 as a set of reduced words over S.

In the case of a general permutation group, the next step is where the problem starts to get difficult. To continue, we need to find generators for G_1 (the stabilizer of x_1), and express them as words over S, before we can proceed to build S_2.

For a reflection group, this problem is easy to solve. In fact, we can use a stabilizer chain of parabolic subgroups, so that the generating set of G_i is merely a subset of the generating set of G_{i-1}. Indeed, supposing $G = W$, let us take W_1 to be the stabilizer of some point $\lambda \in V$. In order to minimize the index $|W|/|W_1|$, we should take λ to be of the form described in (2.1), with J maximal, so that W_1 is a maximal (proper) parabolic subgroup. Thus, the first base point of the stabilizer chain is λ. In order to ensure that the next subgroup in the stabilizer chain is again parabolic, one should choose the next base point to be the unique point in some nontrivial W_1-orbit that belongs to the closure of the fundamental chamber relative to the root system of W_1.

3. The Conjugacy Problem

For any computation involving characters of Weyl groups, fast manipulation of conjugacy class data is essential. Closely related computational problems also occur in working with characters of the corresponding Iwahori-Hecke algebras (e.g., see [**GM**], [**GP**]).

Problem 3.1.

(a) *Given group elements $w_1, w_2 \in W$ (represented as words in the simple reflections), decide whether they are conjugate in W.*

(b) *Given a group element $w \in W$, produce a canonical representative of the conjugacy class of w in W.*

Note that a solution of (b) immediately yields a solution for (a).

The solution to part (a) of this problem that we outline here is not completely satisfying, since the justification for it depends on a case-by-case analysis of the irreducible groups. On the other hand, the algorithm it provides is nearly uniform—the only exceptional cases involve a few conjugacy classes in type D.

First we mention two other approaches that are more obvious but less effective.

	D_4	D_6	E_6	E_7	E_8	F_4	G_2
$\#CC$	13	37	25	60	112	25	6
$\#CP$	9	28	25	54	106	17	5
$\#CT$	7	20	20	36	112	19	5

Table II: Conjugacy class data.

A. Characteristic polynomials.

A simple necessary condition for conjugacy of w_1 and w_2 in W is that they should be conjugate in $GL(V)$. That is, they should have the same characteristic polynomial:

$$\det(1 - qw_1) = \det(1 - qw_2).$$

It is easy to see that this criterion perfectly separates the conjugacy classes of the symmetric group, and it turns out that it also perfectly separates the conjugacy classes of E_6 and the non-crystallographic groups H_3 and H_4. However in the remaining cases, including B_n and D_n, this criterion fails significantly. In Table II, we list the number of conjugacy classes (CC) in various groups, and compare it with the number of distinct characteristic polynomials (CP) that occur. (The significance of the last line in the table will be explained in the next subsection.) For example in E_7, there are 6 pairs of distinct conjugacy classes with the same characteristic polynomial.

B. Permutation representations.

As an alternative to conjugacy in $GL(V)$, we may instead pass to a permutation representation of W, and test for conjugacy of the permutations representing w_1 and w_2 in the symmetric group. This addresses a second flaw in the characteristic polynomial approach: computing characteristic polynomials is (at least with *Maple*) more expensive than computing cycle-types of permutations.

For example, consider E_8. The length of the average element is $|\Phi^+|/2 = 60$. To determine the characteristic polynomial for an element of length 60 would involve composing 60 reflections 8 times, as well as the evaluation of an 8×8 determinant. In principle this is a symbolic determinant, although it would be enough to evaluate the characteristic polynomial at a single sufficiently generic point.[2] On the other hand, as we noted in §2, E_8 has a parabolic permutation representation of degree 240. To compute the cycle-type of a given a word of length 60 would involve composing 60 involutions on 240 points. In *Maple*, we have found that this runs about 50% faster than the corresponding characteristic polynomial calculation.

However, these permutation representations have a flaw of their own. They do a poor job of separating conjugacy classes, in some cases worse than characteristic

[2] In our experience, the fastest way to separate conjugacy classes in H_4 is to evaluate the characteristic polynomial $\det(1 - qw)$ at $q = 3$.

polynomials. For example, if we use the smallest parabolic permutation representations (see Table I), then we find that the only (irreducible) groups for which these representations perfectly separate the conjugacy classes are A_n and E_8. The last line of Table II shows the number of distinct cycle-types (CT) of permutations that occur in various cases.

Of course, we are free to use other permutation representations, and we are not limited to those arising from parabolic subgroups. For example, F_4 has a Klein group of linear characters, as well as two embeddings of B_4 as a subgroup of index 3 that allow one to separate some of the more tightly bound conjugacy classes. By combining various tricks such as these with characteristic polynomials, it is indeed possible to separate all of the conjugacy classes of every Weyl group.

C. Centrally symmetric orbits.

To explain our preferred approach to separating conjugacy classes in Weyl groups, we begin in a cryptic way by analyzing the conjugacy classes of B_n in some detail.

If we represent B_n as the group of signed permutations of $\{\pm 1, \dots, \pm n\}$ (a permutation π is "signed" if $\pi(-i) = -\pi(i)$), then the cycles of a given group element (as a permutation of $2n$ objects) can be classified into two types, according to whether the support of the cycle contains at most one from each pair $\{i, -i\}$, or is a union of such pairs. The former occur in matched pairs of the same length and are said to be *positive*; the latter are *negative.* The sign classifications of these cycles are preserved under conjugation, and two signed permutations are conjugate in B_n if and only if they have the same inventory of cycle-lengths of each sign.

As an aside, note that this sign-classification makes it clear why the permutation representation of B_n of degree $2n$ fails to separate conjugacy classes.

Continuing our crypticism, let us return to the geometric setting and consider the orbit $\mathcal{O} = W\lambda$ generated by some point $\lambda \in V$. We say that the orbit is *centrally symmetric* if $\mathcal{O} = -\mathcal{O}$. Observe that

1. Every reflection group has (nonzero) centrally symmetric orbits. For example, the root system is a union of such orbits.
2. In reflection groups that include the scalar -1 (such as B_n, D_{2n}, F_4, E_7, and E_8), *every* orbit is centrally symmetric.

Now the crucial observation:

3. If $\mathcal{O}$ is a centrally symmetric orbit, then the action of W on $\mathcal{O}$ represents W as a group of *signed* permutations.

Thus, rather than employ ordinary permutation representations and test conjugacy in the symmetric group, we should use signed permutation representations arising from centrally symmetric orbits, and test for conjugacy in B_m, where $2m$ is the size of the orbit. Not only do we gain the benefits of the finer conjugacy classes of B_m, we also get a two-fold increase in speed: products of signed permutations in B_m can be computed twice as fast as the corresponding products in S_{2m}.

For example, the ordinary permutation representation of E_8 of degree 240 arises from the (centrally symmetric) action of E_8 on its root system, so we can replace this with a more economical representation of E_8 in B_{120}.

How well do centrally symmetric actions separate conjugacy classes? Let $B(\mathcal{O})$ denote the group of signed permutations of the centrally symmetric orbit $\mathcal{O}$.

THEOREM 3.2. *Let $\Phi = \Phi_1 \cup \cdots \cup \Phi_l$ be the partition of the root system into W-orbits. If no irreducible component of Φ is of type D, then w_1 and w_2 are conjugate in W if and only if their images are conjugate in $B(\Phi_1) \times \cdots \times B(\Phi_l)$.*

The proof (which we omit) is a straightforward case-by-case analysis, with the exceptional groups being done by machine computation. The result is false if we omit the restriction on type D, although it probably remains true for D_n with n odd.

We remark that the smallest available centrally symmetric orbits are not necessarily orbits of roots. To separate the conjugacy classes of E_7, we prefer to use a signed permutation representation of degree 36 (i.e., an embedding in B_{36}), together with the sign character. For F_4, we use the action on one of the two orbits of roots (an embedding in B_{12}), together with a permutation representation of degree 3.

D. Canonical representatives.

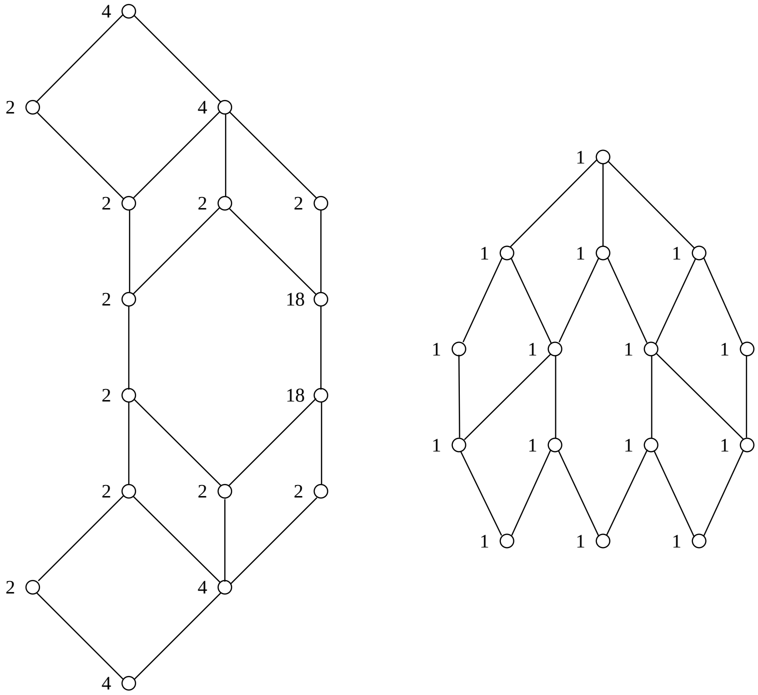
For Problem 3.1(b), the canonical representative problem, the best solution we have (speed being the primary concern) is unsatisfying from the theorist point of view. On the other hand, in the course of producing this solution, we find that there are some interesting mathematical questions that arise.

In our solution to Problem 3.1(a), we found that we can quickly assign a canonical label to the elements of a conjugacy class (in most cases, the signed cycle-type arising from some centrally symmetric action), so it is enough to select one group element for each label that arises. In the classical cases, this is fairly easy to do, and in the exceptional cases, it is enough to do this once and store the results in a table.

How does one decide which element to choose? A reasonable choice would be a representative of minimum length, say the one with the lexicographically smallest reduced expression relative to some ordering of the simple roots. Again for the classical groups this is not hard to do, but for the exceptional groups we have little choice but to resort to a search. One possibility is an exhaustive search of the entire group (e.g., following the method of §4), but another possibility is a separate search within each conjugacy class.

In order to organize such a search, let us impose some structure. For every $w \in W$ and simple reflection s_i, one of the following holds:

$$\ell(s_i w s_i) = \begin{cases} \ell(w) + 2, \\ \ell(w), \\ \ell(w) - 2. \end{cases}$$



Figures I and II: The posets corresponding to the conjugacy classes of $s_2s_3s_2s_3s_4$ in F_4 and s_1s_3 in A_4.

We define $\preccurlyeq$ to be the (implicitly reflexive) transitive closure of the relations

$$s_iws_i \preccurlyeq w \quad \text{whenever} \quad \ell(s_iws_i) \leqslant \ell(w).$$

Any transitive, reflexive relation (i.e., quasi-order) gives rise to a partial ordering on equivalence classes, two elements being equivalent if and only if $x \preccurlyeq y$ and $y \preccurlyeq x$. So in this way we have a natural partial ordering $P(w)$ associated to the conjugacy class $C(w)$ generated by any element w in a Coxeter group.

Observe that it follows directly from the definition that all elements within a given equivalence class have the same length, and the difference in length between any two covering pairs in $P(w)$ is 2. In particular, $P(w)$ is ranked. Two representative examples, one from F_4 and one from A_4, appear in Figures I and II. The vertex labels indicate the number of group elements in each equivalence class.

These posets have a number of interesting properties, and are deserving of further study. Some first results in this direction have been obtained by R. Gill [**G**]. Also, although they did not express their results in order-theoretic terms, there is some relevant work by Geck and Pfeiffer in [**GP**]. For example,

Theorem 3.3 (Geck-Pfeiffer [**GP**, Theorem 1.1(a)]). *For every $w \in W$, all minimal elements of $P(w)$ have the same rank (length).*

A_n	B_n	D_n	E_6	E_7	E_8	F_4	G_2
1	$p(n)$	$p_e(n)$	5	12	30	9	3

Table III: Inaccessible conjugacy classes.

Unfortunately the proof is case-by-case, with the exceptional cases being done by machine computation. On the other hand, once established, this proves that a minimum-length representative for the conjugacy class $C(w)$ can be found by a "greedy" method. Given w, one compares $\ell(w)$ with $\ell(s_i w s_i)$ for all i. If any shorter representative is found, we replace w and start over. Otherwise, we add all new elements of length $\ell(w)$ found in this way to a list, and saturate this list with respect to conjugation by simple reflections (keeping only the elements of length $\ell(w)$). If no shorter elements are found, then the above result guarantees that w is a minimum-length representative of $C(w)$.

It would be nice to have a uniform or conceptual proof of this result.

A variation of this method for producing minimum-length conjugacy class representatives would be to proceed by induction with respect to rank. Having identified a shortest representative for all conjugacy classes belonging to (proper) parabolic subgroups of W, it would suffice to select a shortest representative only for those conjugacy classes of W that do not meet any such subgroup.[3] We say that such conjugacy classes (and the members thereof) are *inaccessible.*

In Table III, we list the number of inaccessible conjugacy classes in each irreducible Weyl group. Here, $p(n)$ denotes the number of partitions of the integer n, and $p_e(n)$ the number of partitions with an even number of parts.

It is easy to test accessibility. Moreover, the proof of the following result shows that we can effectively find an element $x \in W$ that conjugates w to an element in some proper parabolic, if possible.

Proposition 3.4. *The element $w \in W$ is accessible if and only if w has an eigenvector in the linear span of Φ with eigenvalue 1.*

Proof. For simplicity, we may assume $V = \operatorname{Span}\Phi$. If w belongs to the (proper) parabolic subgroup W_J, then w fixes the (nonzero) intersection of the hyperplanes $\alpha_j^\perp : j \in J$, and conjugates of w must fix some transformation of this intersection. Conversely, suppose $w\lambda = \lambda$ for some nonzero $\lambda \in V$. Note that xwx^{-1} fixes $x\lambda$. Therefore, by choosing $x \in W$ so that $x\lambda$ belongs to the closure of the fundamental chamber, we force xwx^{-1} to belong to the parabolic subgroup generated by the simple reflections that fix $x\lambda$ (cf. §2). $\square$

[3] It follows from Theorem 3.3 that a minimum-length representative of a conjugacy class in a parabolic subgroup must also have minimum length within the conjugacy class of the larger group.

Conjecture 3.5. *If w is inaccessible, then the partial order $P(w)$ has a unique minimal element (i.e., all minimum-length conjugates of w belong to the same equivalence class).*

The converse of this conjecture is false; there are many examples of accessible conjugacy classes whose partial orderings have unique minimal elements.

We should point out that an assertion equivalent to Conjecture 3.5 is stated in [**GM**] (see the paragraph following Lemma 1.2) as having been proved but not published by G. Pfeiffer. In any case, we expect this should be straightforward to verify in the classical cases, and a computer search could be used for the exceptional cases. However, again we would prefer to obtain a uniform proof.

A consequence of the conjecture is that if one finds a minimum-length representative of an inaccessible conjugacy class via the greedy algorithm mentioned earlier, then at the point the algorithm halts, one will have constructed a list of *all* minimum-length representatives. It would then be easy to select the one with the lexicographically smallest reduced expression.

As the example in Figure II shows, it can happen that all of the equivalence classes are singletons; i.e., the quasi-ordering of $C(w)$ may itself be a partial order.

Theorem 3.6 (Gill). *The quasi-ordering of $C(w)$ is a partial order if and only if w is an involution.*

The following proof is adapted from [**G**].

Proof. Let C be a conjugacy class of involutions. To prove that every equivalence class in C is a singleton, it suffices to show that if $w \in C$ is an involution, then $\ell(s_iws_i) = \ell(w)$ implies $s_iws_i = w$. For this, suppose first that $\ell(s_iw) > \ell(w)$. In this case, s_iw has a reduced expression that begins with s_i. Since $\ell(s_iws_i) = \ell(w)$ it follows from the Exchange Property (e.g., [**H2**, §5.8]) that a reduced expression for s_iw can be obtained by prepending s_i to a reduced expression for w, deleting one term, and then appending s_i. If the deleted term is not the initial s_i, then we obtain $\ell(ws_i) < \ell(w)$, contradicting the fact that since w is an involution, $\ell(s_iw) = \ell(ws_i)$. So the deleted term must have been the initial s_i, hence $s_iw = ws_i$.

Similarly, if $\ell(s_iw) < \ell(w)$, then there is a reduced expression for w that begins with s_i. Since w is an involution, we also have $\ell(ws_i) < \ell(w)$. So again by the Exchange Property, there is a reduced expression for w that can be obtained by deleting a term (from the expression that begins with s_i) and appending s_i. If the deleted term is not the initial s_i, we contradict the fact that $\ell(s_iws_i) = \ell(w)$. So the deleted term must have been the initial s_i, hence $w = s_iws_i$.

For the converse, suppose that w is a minimum-length representative of a conjugacy class, and that it is the only member of its equivalence class. Choose a reduced expression $w = s_{i_1} \cdots s_{i_l}$. Since conjugation by s_{i_l} has the effect of cyclically permuting the terms of this expression, and l is minimal, it follows that every cyclic permutation of this expression is also reduced. Since the equivalence class is a singleton, these permuted words are expressions for w. It follows that $\ell(ws_j) < \ell(w)$ for every term s_j that appears in a reduced expression for w. This proves that w

is the longest element in some parabolic subgroup of W, and hence an involution (e.g., [**H2**, §1.8]). $\square$

At the opposite extreme, let us mention the following.

QUESTION 3.7. *For which conjugacy classes does the quasi-ordering consist of a single equivalence class (i.e., all conjugates have the same length)?*

Trivially, the identity element and the scalar -1 (if it occurs in W) have this property since they form singleton conjugacy classes. However, there are numerous non-trivial examples, such as the conjugacy class of $w = s_1 s_2 s_3 s_1 s_4 s_3$ in D_4 (with the diagram labeled $\genfrac{}{}{0pt}{}{1}{2}34$).

Most of the examples with this property we have examined so far are what could be described as "roots of w_0" (where w_0 denotes the longest element of W). By this we mean that there is an integer $k \geqslant 1$ such that every element w of the conjugacy class satisfies $w^k = w_0$ and $k\ell(w) = \ell(w_0)$. However, we know of no simple way to describe all conjugacy classes of this type. The only nontrivial conjugacy class we know of that has only one equivalence class but is not a root of w_0 is the conjugacy class of 3-cycles in A_2.

We remark that it is well known that if $w = s_1 \cdots s_n$ (i.e., a Coxeter element) has even order, then a reduced expression for w_0 can be obtained by taking a suitable power of w (Exercise V.6.2 of [**Bo**]). However, it is almost never the case that all conjugates of a Coxeter element have the same length. Nevertheless, roots of w_0 are often (but not always) conjugacy classes consisting of powers of Coxeter elements.

It would also be interesting to see how these results and questions extend to infinite Coxeter groups. Certainly Theorem 3.6 and its proof are valid without changes. However, very little is known about conjugacy in infinite Coxeter groups.

4. Traversal

The main issue we address in this section is the problem of efficiently touring through the elements of a Weyl group; e.g., for the purposes of searching, or for accumulating results that involve sums over Weyl groups.

PROBLEM 4.1. *Devise an efficient data structure for traversing the elements of a Weyl group. In other words, implement*

for w **in** W **do** $\ldots$ **enddo**.

Preferably, one would like to solve this in a way that runs in time proportional to $|W|$ and has bounded space requirements (or at least negligible compared to $|W|$). We can also consider this problem in infinite Coxeter groups by limiting the search to elements whose lengths are bounded by some given amount.

A variation on this problem that occurs in computations with Weyl characters is the analogue for parabolic cosets W/W_J, or equivalently, W-orbits on V.

Problem 4.2. *Devise an efficient data structure for traversing the members of a Weyl group orbit $\mathcal{O} = W\lambda$. In other words, implement*

$$\textbf{for } \mu \textbf{ in } \mathcal{O} \textbf{ do } \ldots \textbf{ enddo}.$$

We cannot afford to solve these problems simply by building the desired list of elements in memory by the saturation method. The larger groups such as E_8 are simply too large for this to be practical. Even all but the smallest E_8-orbits are too large for this.

A slight improvement on the saturation method would be to keep only a list of elements of length l (or in the orbit case, vectors reachable from the fundamental chamber via a sequence of l reflections), and use this to build the next list for the length $l+1$. However, the number of different lengths tends to grow polynomially with rank, whereas the group and orbit sizes grow exponentially. (Consider A_n.) So the number of elements of fixed length, at maximum, is also exponentially large.

A better approach, at least for the first problem, would be to choose a maximal chain of parabolic subgroups, say

$$W = W_0 \supset W_1 \supset \cdots \supset W_n = \{1\},$$

and then build coset representatives X_i for W_{i-1}/W_i ($1 \leqslant i \leqslant n$). Since each $w \in W$ has a unique factorization $w = x_1 \cdots x_n$ ($x_i \in X_i$), all that one needs is to write code for traversing the Cartesian product of $X_1, \ldots, X_n$. We must be careful to choose the subgroup chain so that each X_i is small enough to be generated by a simple method such as saturation; otherwise, we have returned to a problem equivalent to Problem 4.2. In any case, this approach is not of much use for traversing a large W-orbit, unless that orbit is compatible with a parabolic subgroup chain with steps of small index.

A. Finite automata.

By selecting the lexicographically first reduced word for each group element, one may view W as a formal language.

Theorem 4.3 (Brink and Howlett [**BH**]). *In a general Coxeter group, the language of lex-first reduced words is regular; i.e., recognizable by a finite automaton.*

This amounts[4] to the assertion that for each (possibly infinite) Coxeter group, there is a finite directed graph, with edges labeled by simple reflections s_i, so that the language of lex-first reduced words is obtained by generating all directed paths starting at some fixed vertex and ending at some fixed set of vertices.

One can also easily deduce from this an analogous result for parabolic quotients W/W_J. Indeed, if one orders the simple reflections so that the members of J precede all other generators, then it follows from [**H2**, §5.12] that the members

[4] In fact, Brink and Howlett prove more than just the regularity of the language—they prove the existence of an “automatic” structure.

A_6	A_7	A_8	E_6	E_7	E_8
22	29	37	43	80	296

Table IV: Sizes of lex-first recognizing automata.

of the language of lex-first reduced words whose first terms are not members of J are (shortest) coset representatives for $W_J\backslash W$. Thus given an automaton for generating lex-first reduced words for W, one can simply avoid taking the first step along a "J-edge," or build a slightly modified automaton with one extra vertex.

Since it is relatively easy to write efficient code for generating paths in graphs, this would appear to solve Problems 4.1 and 4.2 in arbitrary Coxeter groups. However, it still leaves aside the nontrivial issue of constructing the automaton. Furthermore, once we construct it, we may discover that it is too large to be practical.

On the other hand, given that we are primarily interested in Weyl groups, there is good news. Although Theorem 4.3 is essentially content-free in the case of a finite Coxeter group—any finite language is trivially regular—it turns out that there are indeed some very small automata for generating the lex-first reduced words in Weyl groups (both finite and affine). This is discussed in detail by Casselman [**C**], who attributes the observation (i.e., existence of small automata) to du Cloux. In Table IV, we list the minimum number of states for various cases, taken from [**C**].

It would be interesting to investigate the sensitivity of the automata to the ordering of the simple roots, since a single fixed ordering cannot accommodate traversals of arbitrary parabolic quotients. We expect that the number of states will be correlated with the indices in the corresponding chain of parabolic subgroups. Also, it would be interesting to investigate the sizes of the automata that occur in infinite Coxeter groups beyond the affine cases.

B. A low-technology solution.

Here we would like to discuss a simple approach that has minimal space requirements and comes very close to the goal of linear running time. It also requires very little in the way of special preprocessing—all that one needs is the geometric representation of the Coxeter group and the computation of a small table of roots.

For specialists in combinatorial algorithms, what we present may very well be a standard form of exhaustive search, but it is nevertheless interesting how features of the geometric representation play an essential role in the construction.

We begin by describing a general method for traversing virtually any set of combinatorial objects with a "sufficiently nice" structure; e.g., permutations, multisets, compositions, number partitions, set partitions, tableaux, and so on. It would not be illuminating at this point to attempt a description of the precise features we require, other than to say that a basic prerequisite is that there should be a "natural" partial ordering on the set of objects, and this partial ordering should have a unique minimal element, a root object.[5]

[5] In fact, transitivity is not strictly necessary. Any acyclic digraph with a unique sink will do.

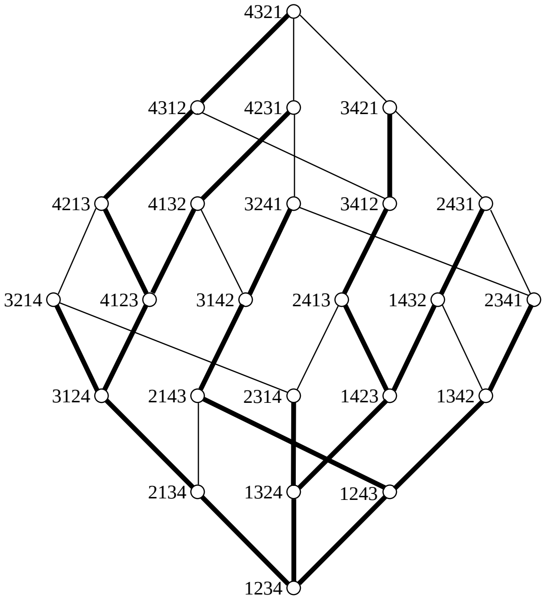


FIGURE III: The weak ordering of A_3.

As an illustration, in Figure III we have selected the set of permutations of 4 objects, partially ordered by the so-called "weak" ordering. (The significance of the thick and thin lines will be explained below.) In any Coxeter group, such as in this case, the symmetric group of degree 4, the (left) weak ordering can be defined as the transitive closure of the relations

$$s_i w < w \quad \text{whenever} \quad \ell(s_i w) < \ell(w).$$

Here, the simple reflections are the adjacent transpositions, and the length of the permutation π is the inversion number; i.e., the number of pairs $i < j$ such that $\pi(i) > \pi(j)$.

Having settled on a partial ordering of the set of objects, the next step is to choose, for each non-root object x, a canonical "ancestral" object $y < x$. In the case of a Coxeter group, it is natural to determine the least i such that $\ell(s_i x) < \ell(x)$, and select $y = s_i x$. Since each non-root object has exactly one canonical ancestor, these choices have the effect of selecting a (rooted) spanning tree on the set of objects.

Applying this scheme to the particular case of the symmetric group, one finds that the canonical ancestor of the permutation π is obtained by transposing i and $i+1$, where i is the least index such that $i+1$ precedes i in $(\pi(1), \ldots, \pi(n))$. This is illustrated in Figure III, where we have used thick lines to mark the tree edges that indicate canonical ancestry.

The traversal scheme we propose amounts to a depth-first search of the chosen spanning tree. While depth-first search is a standard graph algorithm, what makes the problem difficult for our application is that the number of objects is so large that the tree structure itself cannot be stored in memory—we have to be able to generate the local structure from local data. In particular, the crucial problem to be solved, the one that determines whether the partial ordering and spanning tree are "sufficiently nice," is the Canonical Descendant Problem:

Given an object y, determine all x such that y is the canonical ancestor of x.

Of course, we have to be able to solve this problem using data localized at y. Furthermore, if we seek to traverse the set of objects in linear time, we need to be able to solve this problem in time proportional to the number of objects x that are canonical descendants of y, plus some constant amount of overhead. (Since we are searching a tree, the number of edges and the number of objects are roughly the same.)

It is remarkable that virtually all of the standard combinatorial structures have natural partial orderings whose Canonical Descendant Problem admits a good solution.

C. Implementation.

Before solving this problem in the case of Coxeter groups, we first briefly discuss how one uses the solution to implement a traversal of the set of objects.

During the tour, we maintain two stacks: one is a stack of objects that records the path along canonical (i.e., tree) edges from the current object y to the root object. Note that in the Coxeter group case, the simple reflections corresponding to this sequence of objects form the lexicographically first reduced word for y. The second stack consists of lists of objects that assist in directing the tour.

The search begins at the root object, with both stacks empty.

Once we arrive at object y, we compute the list of objects x that have chosen y as their canonical ancestor (here is where the Canonical Descendant Problem occurs). If the list is non-empty, we select the first object x on the list, push y onto the top of the object stack, push the list (with x deleted) onto the top of the list stack, and travel from y to x.

Otherwise, if the list is empty, we backtrack: pop the items at the top of the object and list stacks, and repeat until the top list is nonempty. (If there is no such list, the tour is complete.) This takes us back to the object y' at the top of the object stack. We now delete the first object x from the top list, travel from y' to x, and continue the tour.

Note that we backtrack along each tree edge exactly once, so the total cost of backtracking is proportional to the number of objects.

The amount of space used by this scheme is controlled roughly by the height of the tree and the maximum number of canonical descendants possessed by an object. In the Coxeter group case, the former is proportional to the length of an average group element, and the latter is n, the rank. In practice, this amount of space is negligible.

D. Canonical descendants in Coxeter groups.

Finally, to solve the Canonical Descendant Problem in an arbitrary Coxeter group, we begin by choosing a point $\lambda \in V$ in the fundamental chamber. It will simplify matters to regard the objects as ordered pairs (w, μ), where $\mu = w\lambda$.

For Problem 4.2, the orbit problem, we would allow λ to be on the boundary of the fundamental chamber. However in the following, we will restrict our attention to the case of a generic orbit, and leave to the reader the task of adjusting for the general case.

Note that the canonical ancestor of object (w, μ) is $(s_i w, s_i \mu)$, where i is the least index such that $\langle \mu, \alpha_i \rangle < 0$. Let us call i the *ancestral index* of object (w, μ). This index can be recorded when we arrive at (w, μ) for the first time (as part of the object stack), and hence is available to assist in computing the canonical descendants of (w, μ).

If i is the ancestral index of (w, μ), we say dually that i is a *descendant index* for the object $(s_i w, s_i \mu)$. In these terms, the objective is to determine all descendant indices for the object (w, μ).

LEMMA 4.4. *If i is the ancestral index of (w, μ), then $1, 2, \ldots, i-1$ are descendant indices for (w, μ).*

Proof. Consider any $j < i$. Since i is the ancestral index, it must be the case that $\langle \mu, \alpha_j \rangle > 0$, or equivalently, $\langle s_j \mu, \alpha_j \rangle < 0$. If there were any index $k < j$ such that $\langle s_j \mu, \alpha_k \rangle < 0$, then we would also have $\langle s_j \mu, s_j \alpha_k \rangle < 0$, since $s_j(\alpha_k)$ is in the nonnegative linear span of α_j and α_k. Hence $\langle \mu, \alpha_k \rangle < 0$, contradicting the fact that i is the ancestral index and $k < i$. It therefore must be the case that $\langle s_j \mu, \alpha_k \rangle > 0$ for all $k < j$, and therefore j is the ancestral index of $(s_j w, s_j \mu)$. □

LEMMA 4.5. *Assume $\langle \alpha_i, \alpha_j \rangle = 0$. If i is the ancestral index of (w, μ) and $j > i$, then j is not a descendant index for (w, μ).*

Proof. Since i is the ancestral index, we must have $\langle \mu, \alpha_i \rangle < 0$. However, $\langle \alpha_i, \alpha_j \rangle = 0$ implies $s_j \alpha_i = \alpha_i$, hence $\langle s_j \mu, \alpha_i \rangle < 0$. In other words, the smallest index k such that $\langle s_j \mu, \alpha_k \rangle < 0$ is $\leqslant i$, so j cannot be the ancestral index for $(s_j w, s_j \mu)$. □

LEMMA 4.6. *Assume $\langle \alpha_i, \alpha_j \rangle \neq 0$. If i is the ancestral index of (w, μ) and $j > i$, then j is a descendant index for (w, μ) if and only if $\langle \mu, s_j \alpha_k \rangle > 0$ for all $i \leqslant k < j$.*

Proof. Since $\langle s_j \mu, \alpha_k \rangle = \langle \mu, s_j \alpha_k \rangle$, the stated conditions are clearly necessary for j to be the ancestral index of $(s_j w, s_j \mu)$. Conversely, given these conditions, we have in particular that $\langle \mu, s_j \alpha_i \rangle > 0$. We also have $\langle \mu, \alpha_i \rangle < 0$, since i is the ancestral index. Now since we are given $\langle \alpha_i, \alpha_j \rangle \neq 0$, it follows that α_j is a positive multiple of $s_j \alpha_i - \alpha_i$, so we may deduce $\langle \mu, \alpha_j \rangle > 0$, or equivalently $\langle s_j \mu, \alpha_j \rangle < 0$.

To prove that j is the ancestral index of $(s_j w, s_j \mu)$, it now suffices to show that $\langle s_j \mu, \alpha_k \rangle > 0$ for $k < i$. However, we know that $\langle \mu, \alpha_k \rangle > 0$ (since i is ancestral for (w, μ)) and have established $\langle \mu, \alpha_j \rangle > 0$, so the desired result follows from the fact that $s_j \alpha_k$ is in the nonnegative linear span of α_j and α_k. □

Combining Lemmas 4.4–6, we obtain the following description of $\mathrm{DesInd}(w, \mu)$, the set of descendant indices for (w, μ).

Theorem 4.7. *If i is the ancestral index of (w, μ), then*

$$\mathrm{DesInd}(w, \mu) = \{1, \dots, i-1\} \cup \{j > i : \langle \alpha_i, \alpha_j \rangle \neq 0,\ \langle \mu, s_j \alpha_k \rangle > 0,\ i \leqslant k < j\}\,.$$

In particular, if the diagram of W is linear (i.e., $\langle \alpha_i, \alpha_j \rangle \neq 0$ iff $|i-j| \leqslant 1$), then

$$\mathrm{DesInd}(w, \mu) = \begin{cases} \{1, \dots, i-1\} & \text{if } \langle \mu, s_{i+1} \alpha_i \rangle < 0, \\ \{1, \dots, i-1, i+1\} & \text{otherwise.} \end{cases}$$

An optimization of the computation of these sets of descendant indices requires a small amount of preprocessing. First we should compute, for each simple root α_i, the list of indices $j > i$ that are adjacent to i in the Coxeter graph. In most cases of interest (e.g., finite or affine Weyl groups), there is usually at most one index of this type; namely, $j = i + 1$. Furthermore, we can usually order the indices so that the few cases where this is violated involve the last few indices. The group elements that have these last few indices as their ancestral index will be extremely rare.

Second, for each adjacent pair of nodes $i < j$ in the diagram, we should precompute the list of roots $s_j \alpha_k$ $(i \leqslant k < j)$. In most cases, this list will consist of a single root. If W has a linear diagram, this will be true in every case.

In the case of finite Weyl groups, we have noted previously that most roots α can be represented in the form $\pm\varepsilon_i$ or $\pm\varepsilon_i \pm \varepsilon_j$ (where ε_i denotes an orthonormal basis of V). In these cases, queries of the form $\langle \mu, \alpha \rangle > 0$ require no arithmetic operations—just a single comparison or signed comparison.

For example, consider the "worst" case of E_8. If the simple roots are ordered

$$8 - 7 - \overset{\overset{\displaystyle 6}{|}}{5} - 4 - 3 - 2 - 1,$$

then in the standard realization, the only simple root not of the form $\pm\varepsilon_i \pm \varepsilon_j$ is α_8, which has 8 nonzero coordinates. Furthermore, the proportion of elements whose ancestral index is $\geqslant 5$ is $1/|A_4| = 1/120$. Thus, more than 99% of the time, we can determine the set of descendant indices for a member of E_8 by making a single comparison. Out of the remaining 1%, elements with ancestral index 5 require up to 3 comparisons, 6 and 8 require no computation at all, and ancestral index 7 requires one scalar product with a root having 8 nonzero coordinates.

5. The Weight System of a Weyl Character

We now turn our attention to computational issues more directly related to Weyl characters. In particular, all root systems and Coxeter groups will henceforth be finite and crystallographic. For simplicity, we also assume $V = \mathrm{Span}\,\Phi$.

A. Weyl characters.

Let $\Lambda = \{\lambda \in V : \alpha \in \Phi \Rightarrow \langle \lambda, \alpha^\vee \rangle \in \mathbf{Z}\}$ denote the lattice of integral weights, and define $\omega_i \in \Lambda$ to be the ith fundamental weight; i.e.,

$$\langle \omega_i, \alpha_j^\vee \rangle = \delta_{ij}.$$

The weight lattice is generated by $\omega_1, \ldots, \omega_n$.

In the following, it will be convenient to work in the group ring R of Λ, with the basis element corresponding to λ written as a formal exponential e^λ. Since W permutes Λ, the ring R also carries a W-action: $w.e^\lambda = e^{w\lambda}$.

For each $\lambda \in \Lambda$, we define

$$\Delta(\lambda) = \sum_{w \in W} \operatorname{sgn}(w) e^{w\lambda},$$

where $\operatorname{sgn}(w) = (-1)^{\ell(w)} = \det(w)$. Note that Δ is a skew-symmetric operator in the sense that $\Delta(w\lambda) = \operatorname{sgn}(w)\Delta(\lambda)$. In particular, $\Delta(\lambda) = 0$ if λ is orthogonal to any root. The Weyl Denominator Formula (e.g., [**H1**, §24.3]) asserts that

$$\Delta(\rho) = e^\rho \prod_{\alpha \in \Phi^+} (1 - e^{-\alpha}) = \prod_{\alpha \in \Phi^+} (e^{\alpha/2} - e^{-\alpha/2}), \tag{5.1}$$

where

$$\rho = \sum_{\alpha \in \Phi^+} \alpha/2 \ = \ \omega_1 + \cdots + \omega_n.$$

Let Λ^+ denote the set of *dominant* integral weights; i.e., integral weights λ satisfying $\langle \lambda, \alpha_i \rangle \geqslant 0$ for $1 \leqslant i \leqslant n$. We say that $\lambda \in \Lambda$ is *strongly* dominant if $\langle \lambda, \alpha_i \rangle > 0$ for $1 \leqslant i \leqslant n$, or equivalently, $\lambda = \mu + \rho$, where $\mu \in \Lambda^+$. Note that $\Delta(\lambda + \rho) : \lambda \in \Lambda^+$ forms a $\mathbf{Z}$-basis for the skew-symmetric members of R, whereas the orbit sums

$$M(\lambda) = \sum_{\mu \in W\lambda} e^\mu \qquad (\lambda \in \Lambda^+)$$

form a $\mathbf{Z}$-basis for R^W, the subring of W-invariants in R.

For each $\lambda \in \Lambda^+$, the *Weyl character* $\chi(\lambda)$ may be defined by setting

$$\chi(\lambda) = \Delta(\lambda + \rho)/\Delta(\rho). \tag{5.2}$$

It is not hard to show that $\Delta(\rho)$ divides any skew-symmetric member of R, so $\chi(\lambda)$ is a well-defined member of R^W. In particular, there exist integers $K_{\lambda,\mu}$ such that

$$\chi(\lambda) = \sum_{\mu \in \Lambda^+} K_{\lambda,\mu} M(\mu). \tag{5.3}$$

Furthermore, since the skew-sums $\Delta(\lambda+\rho)$ are a $\mathbf{Z}$-basis for the skew-symmetric elements, it follows that the Weyl characters are a $\mathbf{Z}$-basis for the invariants. In particular, there exist integers $c^{\lambda}_{\mu,\nu}$ such that

$$\chi(\mu)\chi(\nu) = \sum_{\lambda\in\Lambda^+} c^{\lambda}_{\mu,\nu}\chi(\lambda). \tag{5.4}$$

Of course, Weyl characters are of interest not merely as formal objects. If $\mathfrak{g}$ is a (complex, semi-simple) Lie algebra with root system Φ, and G is a (complex, connected, simply connected, semi-simple) Lie group with Lie algebra $\mathfrak{g}$, then the irreducible finite-dimensional representations of G (and $\mathfrak{g}$) are indexed by $\lambda \in \Lambda^+$, and the Weyl character $\chi(\lambda)$ encodes the trace function for the representation U_λ indexed by λ.

In particular, the coefficient of e^{μ} in $\chi(\lambda)$ is the dimension of the μ-weight space of the representation U_λ, so the coefficients $K_{\lambda,\mu}$ in (5.3) are nonnegative. Likewise, the coefficient $c^{\lambda}_{\mu,\nu}$ in (5.4) is also nonnegative, being the multiplicity of U_λ in $U_\mu \otimes U_\nu$.

B. The partial ordering of integral weights.

Using (5.1), we can rewrite the definition of $\chi(\lambda)$ in the form

$$\chi(\lambda) = \sum_{w\in W} \operatorname{sgn}(w) e^{w(\lambda+\rho)-\rho} \prod_{\alpha\in\Phi^+} \frac{1}{1-e^{-\alpha}}, \tag{5.5}$$

interpreting the right side as a formal power series in $R\cdot\mathbf{Z}[[e^{-\alpha_1},\ldots,e^{-\alpha_n}]]$. From this point of view, it is clear that every term e^{μ} appearing in the expansion of $\chi(\lambda)$ satisfies

$$\mu = w(\lambda+\rho) - \rho - \gamma,$$

where γ is a sum of positive roots and $w \in W$. In fact, we claim that $(\lambda+\rho) - w(\lambda+\rho)$ is also a sum of positive roots. This follows easily by induction with respect to $\ell(w)$. For if $\ell(s_i w) > \ell(w)$, then (1.1) implies $\langle w(\lambda+\rho), \alpha_i\rangle > 0$, whence $w(\lambda+\rho) - s_i w(\lambda+\rho)$ is a positive (integer) multiple of α_i.

Thus every term e^{μ} appearing in $\chi(\lambda)$ has the form $\mu = \lambda - \gamma$, where γ is a sum of positive roots. This serves as motivation for the definition of the usual partial ordering of the weight lattice as the transitive closure of the relations

$$\lambda > \mu \quad \text{whenever} \quad \lambda - \mu \in \Phi^+.$$

Note that since the root lattice $\mathbf{Z}\Phi$ is a sublattice of Λ, this partial ordering breaks into connected components, one for each coset of $\Lambda/\mathbf{Z}\Phi$. Each of these components is isomorphic (as a partial order) to the product of n copies of $\mathbf{Z}$, each equipped with the usual ordering. We let $f = |\Lambda/\mathbf{Z}\Phi|$ denote the number of components, the so-called *index of connection.*

C. The complexity of a representation.

Given the preceding analysis, it follows that for all $\lambda, \mu \in \Lambda^+$, $K_{\lambda,\mu}$ is nonzero only if $\lambda \geqslant \mu$. In other words, (5.3) may be rewritten in the form

$$\chi(\lambda) = \sum_{\mu \in \Lambda^+ :\, \lambda \geqslant \mu} K_{\lambda,\mu} M(\mu).$$

Furthermore, one can see that there is a unique term in (5.5) that contributes to e^λ, yielding $K_{\lambda,\lambda} = 1$. Hence the transition matrix between Weyl characters and orbit sums is unitriangular with respect to $<$. This provides another proof of the fact that the Weyl characters are a $\mathbf{Z}$-basis for R^W.

We remark that by representation-theoretic methods (e.g., [**H1**, §21.3]), it is known that

$$\lambda, \mu \in \Lambda^+,\ \lambda \geqslant \mu \ \Rightarrow\ K_{\lambda,\mu} \neq 0. \tag{5.6}$$

It would be interesting to find an elementary proof of this fact, starting from (5.2). So far as we are aware, this has been done only for type A.

Given $\lambda \in \Lambda^+$, we define

$$\Lambda^+(\lambda) = \{\mu \in \Lambda^+ : \lambda \geqslant \mu\}.$$

Since dominant integral weights are maximums with respect to $\geqslant$ within their W-orbits, it follows that $\Lambda^+(\lambda)$ is a set of orbit representatives for

$$\Lambda(\lambda) = \{\mu \in \Lambda : w \in W \Rightarrow \lambda \geqslant w\mu\}.$$

Bearing in mind (5.6), $\Lambda(\lambda)$ is the *weight system* of U_λ, the set of weights that occur with positive multiplicity in U_λ.

In computations with Weyl characters, our experience has been that a good measure of the "complexity" of a representation U_λ is the size of $\Lambda^+(\lambda)$; i.e., the number of orbits of weights. For example, by this measure, the nontrivial representations with the least complexity (i.e., $|\Lambda^+(\lambda)| = 1$) are the so-called *minuscule* representations. As a second example, consider the complexity of the representation U_ρ, the smallest representation whose highest weight does not lie on the boundary of the fundamental chamber. The reader whose experience is mainly concerned with type A may think of this as a smallish representation—the corresponding Young diagram is the "staircase" shape. On the other hand, consider the data provided in Table V. In our experience, U_ρ is a rather complex representation.

D. Generating the weight system.

The following problem is at the core of many Weyl character computations.

Problem 5.1. *Given a dominant integral weight λ, determine $\Lambda^+(\lambda)$.*

For example, we invite the reader looking for a computational challenge simply to determine the size of $\Lambda^+(2\rho)$ (i.e., the complexity of $U_{2\rho}$) in the case of E_8. It is a conjecture of Kostant that the multiplicity of U_λ in $U_\rho \otimes U_\rho$ is nonzero if and only if $\lambda \in \Lambda^+(2\rho)$.

It is natural to approach Problem 5.1 recursively, which leads to the following refinement.

A_1	1								
A_2	2	B_2	2					G_2	5
A_3	4	B_3	6	C_3	8				
A_4	9	B_4	19	C_4	26	D_4	14	F_4	58
A_5	22	B_5	64	C_5	81	D_5	44		
A_6	59	B_6	223	C_6	277	D_6	147	E_6	226
A_7	167	B_7	796	C_7	996	D_7	537	E_7	1464
A_8	490	B_8	2887	C_8	3584	D_8	1976	E_8	14869

Table V: The complexity of U_ρ.

Problem 5.1$'$. *Given $\lambda \in \Lambda^+$, determine all $\mu \in \Lambda^+$ that are covered by λ in $(\Lambda^+, <)$ (i.e., $\lambda > \mu$, but there is no $\nu \in \Lambda^+$ such that $\lambda > \nu > \mu$).*

Although the poset $(\Lambda, <)$ has a simple structure, one should realize that the subposet formed by Λ^+ is rather subtle. For example, in Figure IV is displayed the portion of $(\Lambda^+, <)$ below ρ in the case of F_4. The vertex labels indicate the coordinates of the corresponding weights; i.e., the weight $m_1\omega_1 + \cdots + m_4\omega_4$ is labeled $m_1 \cdots m_4$.

For a detailed study of the posets $(\Lambda^+, <)$, see [**St**].

Let us define $\mathcal{E}(\Phi) \subset \mathbf{Z}_{\geqslant 0}\Phi$ to be the set of *elementary moves*; i.e., the set of differences $\lambda - \mu$ that occur as (λ, μ) vary over all covering pairs in $(\Lambda^+, <)$.

In type A, the partial ordering of Λ^+ is closely related to the dominance ordering on partitions of an integer, and it is a familiar combinatorial fact that a covering pair in dominance order must differ by an increment of the ith part and decrement of the jth part, for some $i < j$. Translating this into the present context, every elementary move in type A is a root. Conversely, it is not hard to show that every positive root in type A does arise as the difference between the members of a covering pair.

It is natural to guess that in the general case the elementary moves are the positive roots. Indeed, it is true that $\mathcal{E}(\Phi) \subseteq \Phi^+$; i.e.,

$$\lambda \text{ covers } \mu \text{ in } (\Lambda^+, <) \;\Rightarrow\; \lambda - \mu \in \Phi^+, \tag{5.7}$$

and the earliest reference for this fact we have found is a 1982 paper of Moody and Patera [**MP**]. However, it is *only* in the case $\Phi = A_n$ that $\mathcal{E}(\Phi) = \Phi^+$. For example, in the case of E_8, only 44 of the 120 positive roots are elementary moves.

To describe what happens in the general case, let us recall that in an irreducible root system, there are either one or two orbits of roots. In the latter case, roots in the two orbits are of different lengths (“long” and “short”). In case there is only one orbit, we can agree to call the roots long or short, whichever happens to be

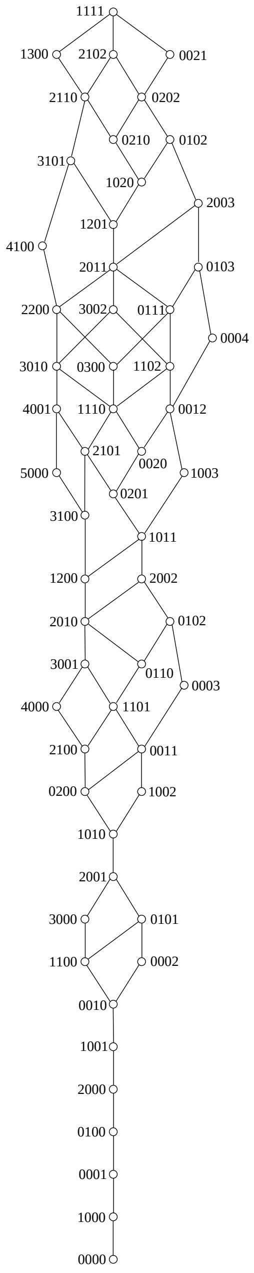


FIGURE IV: The partial ordering of $\Lambda^+(\rho)$ in F_4.

more convenient in a given context. With this convention, every irreducible root system has a unique short root that is dominant.

Given $J \subseteq \{1, \dots, n\}$, let Φ_J denote the root subsystem of Φ generated by $\{\alpha_j : j \in J\}$. Assuming that Φ_J is irreducible, we say that the short dominant root of Φ_J is a *locally short dominant* (LSD) root for Φ, even though it may be "long" as a member of Φ. Note that the number of LSD roots in Φ is simply the number of connected subgraphs of the Dynkin diagram. In the case of a linear diagram, this is $\binom{n+1}{2}$. In the case of a forked diagram, it can be shown that the number is $\binom{n+2}{2} - f$, where f denotes the index of connection as in §5B. (See Corollary 2.11 of [**St**].)

THEOREM 5.2 [**St**]. *Assume that Φ is irreducible.*

(a) *Every LSD root in Φ is an elementary move.*
(b) *Conversely, every elementary move is an LSD root unless Φ is of type G_2, in which case $\alpha_1 + \alpha_2$ is also an elementary move.*

For the proof, we refer the reader to [**St**]. However we cannot resist including the following argument, due to R. Steinberg, showing that all elementary moves are roots.

Proof of (5.7). Among all expressions $\lambda - \mu = \beta_1 + \cdots + \beta_l$ with $\beta_i \in \Phi^+$, choose one that maximizes the sum of the simple root coordinates of β_1. If $\mu + \beta_1$ were not dominant, say $\langle \mu + \beta_1, \alpha_i^\vee \rangle < 0$, then we would have $\langle \beta_1, \alpha_i^\vee \rangle < 0$, so $\beta_1 + \alpha_i$ would be a (positive) root. Moreover, since $\lambda = (\mu + \beta_1) + (\beta_2 + \cdots + \beta_l)$ is dominant, we must also have $\langle \beta_2 + \cdots + \beta_l, \alpha_i^\vee \rangle > 0$. Reordering indices if necessary, we may assume that $\langle \beta_2, \alpha_i^\vee \rangle > 0$. But then $\beta_2 - \alpha_i$ is a positive root or zero, and the expression

$$\lambda - \mu = (\beta_1 + \alpha_i) + (\beta_2 - \alpha_i) + \beta_3 + \cdots + \beta_l$$

contradicts the choice of β_1. Therefore $\mu + \beta_1$ is dominant and $\mu < \mu + \beta_1 \leqslant \lambda$. Given that λ covers μ, this implies $\lambda - \mu = \beta_1$. □

In order to efficiently determine the set of dominant integral weights covered by λ (thereby solving Problem 5.1′), we may proceed as follows.

First, we prepare in advance the list of elementary moves and their coordinates with respect to the fundamental weights. For simplicity, we will assume $\Phi \neq G_2$, so that these are merely the LSD roots.

Let α_J denote the LSD root corresponding to the irreducible subsystem Φ_J. It is easy to show that the support of α_J, as a linear combination of simple roots, is precisely J. A stronger observation is that

$$I \subset J \;\Leftrightarrow\; \alpha_I < \alpha_J.$$

In case Φ_J has only one orbit of roots, this follows from the fact that dominant roots are maximal in their orbits. Otherwise, this requires a bit more work (e.g., α_I could be long in Φ_J), but is simply a matter of checking a few cases.

Consequently, λ covers μ in $(\Lambda^+, <)$ if and only if $\mu = \lambda - \alpha_J$, where J is a (nonempty) connected subset of the diagram of Φ that is inclusion-minimal with respect to the property that $\lambda - \alpha_J$ is dominant.

In deciding whether $\lambda - \alpha_J$ is dominant, notice that the only information one needs from α_J are its positive weight coordinates. For example, if $J = \{j\}$ (i.e., α_J is the simple root α_j), the coefficient of ω_j is 2, and this is the only positive coordinate. Thus λ covers $\lambda - \alpha_j$ if and only if the coefficient of ω_j in λ is $\geqslant 2$. In that case, there can be no other covering relations involving $\lambda - \alpha_I$ with $j \in I$.

Of course, if we are not concerned with generating only *covering* pairs, one could simply compare weight coordinates to determine all LSD roots α_J such that $\lambda - \alpha_J \in \Lambda^+$. Otherwise, one could organize a search through the partial order of LSD roots, backtracking whenever one finds a root α_J such that $\lambda - \alpha_J$ is dominant.

6. Weight Multiplicities

Consider the problem of computing the weight multiplicities of the irreducible representation U_λ, or equivalently, the coefficients $K_{\lambda,\mu}$ appearing in (5.3). Although we defined $K_{\lambda,\mu}$ only for dominant μ, it is convenient more generally to interpret $K_{\lambda,\mu}$ as the coefficient of e^μ in $\chi(\lambda)$ for any $\mu \in \Lambda$.

A. Freudenthal's formula.

The most commonly used algorithms for computing weight multiplicity are based on the following.

Theorem 6.1 (Freudenthal). *For all $\lambda \in \Lambda^+, \mu \in \Lambda$, we have*

$$(c_\lambda - c_\mu)K_{\lambda,\mu} = 2 \sum_{\alpha \in \Phi^+} \sum_{i \geqslant 1} \langle \mu + i\alpha, \alpha \rangle K_{\lambda,\mu+i\alpha}, \tag{6.1}$$

where $c_\nu = \langle \nu + \rho, \nu + \rho \rangle$.

A proof can be found in [**H1**, §22.3], for example. It would be interesting if (6.1) could be understood at a combinatorial level. So far as we know, this is an open problem even in type A.

To use Freudenthal's formula as a recursion, one starts with $K_{\lambda,\lambda} = 1$. In subsequent computations, given that $K_{\lambda,\nu}$ has been determined for all $\lambda \geqslant \nu > \mu$, one can then use the formula to compute $K_{\lambda,\mu}$. Of course, there would be obvious difficulties if it happened that $c_\lambda = c_\mu$. However, it is easily shown that $c_\lambda > c_\mu$ if $\mu < \lambda$ is dominant, and replacing μ with a non-dominant member of its orbit can only decrease the value of c_μ [**H1**, §13].

Without further refinements, what we have just described is too unwieldy to be useful for computing weight multiplicities in all but the smallest cases. Indeed, if $\lambda - \mu = c_1\alpha_1 + \cdots + c_n\alpha_n$, then the above scheme would require $(c_1+1)\cdots(c_n+1)$ iterations of (6.1). In the case of the adjoint representation of E_8, a representation of complexity two (i.e., two orbits of weights), it would take 151200 iterations to compute the dimension of the zero weight space.

In order to increase the useability of (6.1), one should take advantage of the Weyl group symmetry $K_{\lambda,\nu} = K_{\lambda,w\nu}$ and confine all computations to the dominant chamber. Indeed, if $\nu = \mu + i\alpha$ fails to be dominant, then we can replace ν with the maximal (hence dominant) weight ν' in the same W-orbit. Also, one should recognize that as soon as $\nu \not\leqslant \lambda$, then the same will be true for all larger values of i and the weight string in the direction of α can be terminated.

Thus in order to compute the full set of dominant weight multiplicities for U_λ, one should only apply (6.1) once for each dominant weight $\mu < \lambda$. In fact, to simplify bookkeeping for the recurrence, one could first generate $\Lambda^+(\lambda)$ following the method of §5D and order the applications of (6.1) in a way consistent with $>$; e.g., sorted by decreasing values of $\langle \mu, \rho \rangle$ or c_μ.

B. The Moody-Patera refinement.

Bearing in mind the data in Table V, the weight systems of the representations most amenable to computation are largely confined near the walls of the fundamental chamber, and hence the relevant weights tend to have nontrivial stabilizers. This leads to the possibility of further uses of Weyl group symmetry to improve on the basic Freudenthal algorithm, an idea due to Moody and Patera [**MP**].

Fix dominant integral weights $\lambda > \mu$, and consider the terms appearing in (6.1). The stabilizer of μ is the parabolic subgroup W_J, where $J = \{j : \langle \mu, \alpha_j \rangle = 0\}$. If $\alpha, \beta \in \Phi^+$ belong to the same W_J-orbit, say $\alpha = w\beta$, then we have

$$K_{\lambda,\mu+i\beta} = K_{\lambda,w(\mu+i\beta)} = K_{\lambda,\mu+i\alpha},$$
$$\langle \mu + i\beta, \beta \rangle = \langle w(\mu + i\beta), w\beta \rangle = \langle \mu + i\alpha, \alpha \rangle,$$

so the contributions to (6.1) indexed by α and β are identical. Hence we should partition Φ into W_J-orbits, and collect separately the contributions to (6.1) from each orbit.

The collecting is complicated by the fact that a given W_J-orbit in Φ may not have an equal number of positive and negative roots. However, the longest element x_0 of W_J interchanges the positive and negative roots in Φ_J and must permute the remaining positive roots in Φ [**H2**, §1.8]. Furthermore, x_0 acts as an order-reversing involution on each W_J-orbit. Thus if a W_J-orbit has both positive and negative roots, then x_0 maps the maximal root (which must be positive) to the minimal root (which must be negative), so by the previous remark, these roots must belong to Φ_J. The remaining orbits occur in pairs, one consisting of positive roots; the other, their negatives.

Let $\beta_1, \ldots, \beta_l \in \Phi^+$ denote the positive roots that are dominant relative to Φ_J. We may compute the sizes of the corresponding W_J-orbits, say $b_1, \ldots, b_l$, using the formula $b_j = |W_J|/|W_I|$, where $I = \{i \in J : \langle \beta_j, \alpha_i \rangle = 0\}$. By the preceding discussion, it follows that (6.1) may be rewritten as

$$(c_\lambda - c_\mu) K_{\lambda,\mu} = \sum_j \kappa_j b_j \sum_{i \geqslant 1} \langle \mu + i\beta_j, \beta_j \rangle K_{\lambda,\mu+i\beta_j},$$

where $\kappa_j = 1$ if $\beta_j \in \Phi_J$, and $\kappa_j = 2$ otherwise.

To accelerate a series of weight multiplicity computations, one could store in advance the orbit representatives β_j and the quantities $\kappa_j b_j$ for each parabolic subgroup W_J.

C. The q-analogue of weight multiplicity.

By extracting the coefficient of e^μ from (5.5), one obtains

$$K_{\lambda,\mu} = \sum_{w\in W} \operatorname{sgn}(w)[e^{w(\lambda+\rho)-(\mu+\rho)}] \prod_{\alpha\in\Phi^+} \frac{1}{1-e^\alpha},$$

where $[e^\gamma]$ denotes the coefficient-of-e^γ-operator. This is Kostant's weight multiplicity formula [**H1**, §24.2].

Lusztig [**Lu**] introduced a q-analogue of the above formula by setting

$$K_{\lambda,\mu}(q) = \sum_{w\in W} \operatorname{sgn}(w)[e^{w(\lambda+\rho)-(\mu+\rho)}] \prod_{\alpha\in\Phi^+} \frac{1}{1-qe^\alpha}. \tag{6.2}$$

He conjectured that for dominant $\mu \leqslant \lambda$, the polynomial $q^{h(\lambda-\mu)}K_{\lambda,\mu}(q^{-1})$ is a particular Kazhdan-Lusztig polynomial for the corresponding affine Weyl group. (Here we are using $h(\gamma)$ to denote the sum of the simple root coordinates of γ.) This conjecture was proved by Kato [**Ka**] using p-adic groups, and an elementary proof was later given by R. Gupta (Brylinski) [**Gu**]. Thus these polynomials have nonnegative (integer) coefficients.

We must caution the reader that although (6.2) makes sense for any weight μ, these q-analogues of weight multiplicity are *not* constant on W-orbits, and may have negative coefficients if μ is not dominant. For example, it is easy to see that

$$K_{\lambda,\mu}(0) = \begin{cases} \operatorname{sgn}(w) & \text{if } \lambda+\rho = w(\mu+\rho), \\ 0 & \text{otherwise,} \end{cases} \tag{6.3}$$

which confirms both of these non-properties.

The following identity involving $K_{\lambda,\mu}(q)$ is due to A. Broer [**Br**, §4]. His proof is very short; we include an even shorter one.

Theorem 6.2 (Broer). *For all $\lambda, \mu \in \Lambda$ with λ dominant, we have*

$$\frac{d}{dq}K_{\lambda,\mu}(q) = \sum_{\alpha\in\Phi^+}\sum_{i\geqslant 1} q^{i-1}K_{\lambda,\mu+i\alpha}(q).$$

Proof. Differentiating (6.2) with respect to q yields

$$\begin{aligned}\frac{d}{dq}K_{\lambda,\mu}(q) &= \sum_{w\in W} \operatorname{sgn}(w)[e^{w(\lambda+\rho)-(\mu+\rho)}] \sum_{\alpha\in\Phi^+} \frac{e^\alpha}{1-qe^\alpha} \prod_{\beta\in\Phi^+} \frac{1}{1-qe^\beta} \\ &= \sum_{\alpha\in\Phi^+}\sum_{i\geqslant 1} q^{i-1} \sum_{w\in W} \operatorname{sgn}(w)[e^{w(\lambda+\rho)-(\mu+i\alpha+\rho)}] \prod_{\beta\in\Phi^+} \frac{1}{1-qe^\beta}.\end{aligned}$$

Now reapply (6.2). □

We can use this identity to compute $K_{\lambda,\mu}(q)$ in a manner similar to the one we described for Freudenthal's formula, using (6.3) to recover the constant of integration killed by the derivative. However, as noted above, there is no Weyl group invariance available for us to exploit, so we are limited to the "slow" algorithm described at the beginning of §6A.

7. Tensor Product Multiplicities

Our final topic is the problem of computing the multiplicities of the irreducible representations in $U_\mu \otimes U_\nu$, or equivalently, the coefficients $c^\lambda_{\mu,\nu}$ appearing in (5.4).

It will be convenient to extend (5.2), the definition of $\chi(\lambda)$, by allowing λ to be non-dominant. With this convention, we have either $\chi(\lambda) = 0$ (if $\lambda+\rho$ is orthogonal to one of the roots), or else $\lambda + \rho$ has a W-orbit that includes a strongly dominant weight $\lambda' + \rho$, in which case $\chi(\lambda) = \mathrm{sgn}(w)\chi(\lambda')$, where $w(\lambda + \rho) = \lambda' + \rho$.

A. The Brauer-Klimyk formula.

While it would be interesting to see algorithms based on Kashiwara's crystal bases [**K**] or Littelmann's path model [**L1-2**], it seems that a widely used strategy for computing tensor product multiplicities is based on the following result. It is often attributed to Klimyk (see [**Kl**]), although in the notes for Chapter 24 in [**H1**], Humphreys traces it back to a 1937 paper of R. Brauer [**B**].

Theorem 7.1 (Brauer-Klimyk). *For all $\mu, \nu \in \Lambda^+$, we have*

$$\chi(\mu)\chi(\nu) = \sum_{\xi\in\Lambda} K_{\mu,\xi}\, \chi(\nu + \xi). \tag{7.1}$$

Proof. Since $\chi(\lambda)\Delta(\rho) = \Delta(\lambda + \rho)$, we find

$$\begin{aligned}\chi(\mu)\chi(\nu)\Delta(\rho) = \chi(\mu)\Delta(\nu + \rho) &= \sum_{\xi\in\Lambda} K_{\mu,\xi}\, e^\xi \sum_{w\in W} \mathrm{sgn}(w)e^{w(\nu+\rho)} \\ &= \sum_{\xi\in\Lambda,\, w\in W} \mathrm{sgn}(w)K_{\mu,\xi}\, e^{w(\nu+\xi+\rho)} = \sum_{\xi\in\Lambda} K_{\mu,\xi}\Delta(\nu + \xi + \rho),\end{aligned}$$

using the W-invariance of weight multiplicities. □

One may use this as the basis of a tensor product algorithm as follows. First, determine the dominant weight multiplicities for $\chi(\mu)$ by the method described in §6. Second, for each dominant $\xi \leqslant \mu$, one needs to traverse the full W-orbit of ξ: for each orbit member $w\xi$, one finds the dominant member of the orbit of $\nu + w\xi + \rho$, keeping track of the parity p of the number of reflections used during the calculation. If the result is strongly dominant, say $\lambda + \rho$, one adds $(-1)^p K_{\mu,\xi}$ to the multiplicity of $\chi(\lambda)$.

This algorithm has one major flaw in that it requires traversal of W-orbits. As we have noted previously, most of the Weyl groups tend to have very few small orbits. For example, after the adjoint representation of E_8, the next smallest representation has complexity three, and these three orbits of weights have sizes 1, 240, and 2160. To use the algorithm effectively, one has to develop an efficient means of orbit traversal; e.g., following the methods outlined in §4.

In any case, the Brauer-Klimyk formula is most effective if one of the representations is "small." In the particular case of a minuscule representation, one in which there is just a single orbit of weights, the resulting decomposition has a rather simple form.

Corollary 7.2. *For all $\mu, \nu \in \Lambda^+$ with μ minuscule, we have*

$$\chi(\mu)\chi(\nu) \;=\; \sum_{\xi \in W\mu:\, \nu+\xi \in \Lambda^+} \chi(\nu+\xi).$$

Proof. Consider a typical term $K_{\mu,\xi}\,\chi(\nu+\xi)$ in (7.1). Since there is only one orbit of weights, we may assume $w\xi = \mu$, say. Hence $\langle \xi, \alpha_i^\vee \rangle = \langle \mu, (w\alpha_i)^\vee \rangle$ and $K_{\mu,\xi} = 1$. However, minuscule weights have the (characterizing) property that $\langle \mu, \alpha^\vee \rangle \in \{0, \pm 1\}$ for all $\alpha \in \Phi$ (e.g., Exercise VI.1.24 in [**Bo**]), so if $\nu + \xi$ fails to be dominant, we must have $\langle \nu+\xi, \alpha_i^\vee \rangle = -1$ for some i, whence $\langle \nu+\xi+\rho, \alpha_i^\vee \rangle = 0$ and $\chi(\nu+\xi) = 0$. □

The next simplest cases are the quasi-minuscule representations. Assuming that Φ is irreducible, the highest weight of a quasi-minuscule representation is the short dominant root $\bar{\alpha}$. There are just two orbits of weights: the short roots and zero, the latter having multiplicity equal to the number of short simple roots. In case Φ has only one W-orbit, this is the adjoint representation.

Corollary 7.3. *Assume Φ is irreducible. For all $\nu \in \Lambda^+$, we have*

$$\chi(\bar{\alpha})\chi(\nu) \;=\; m\chi(\nu) \;+ \sum_{\alpha \in \Phi_s:\, \nu+\alpha \in \Lambda^+} \chi(\nu+\alpha),$$

where Φ_s denotes the set of short roots and m is the number of short simple roots α_i such that $\langle \nu, \alpha_i^\vee \rangle > 0$.

Proof. The term indexed by $\xi = 0$ contributes m_0 copies of $\chi(\nu)$ to (7.1), where m_0 denotes the number of short simple roots. The remaining contributions arise from choosing $\xi = \alpha \in \Phi_s$, in which case $K_{\bar{\alpha},\alpha} = 1$. If $\nu + \alpha$ is dominant, we obtain a contribution of $\chi(\nu+\alpha)$. If $\nu+\alpha$ fails to be dominant, then we must have $\langle \nu+\alpha, \alpha_i^\vee \rangle \leqslant -1$. If equality occurs, then the reasoning in the proof of the previous corollary shows that the corresponding term contributes nothing to (7.1). Hence $\langle \nu+\alpha, \alpha_i^\vee \rangle \leqslant -2$, and in particular $\langle \alpha, \alpha_i^\vee \rangle \leqslant -2$. However since α is short, this occurs if and only if $\alpha = -\alpha_i$ and $\langle \nu, \alpha_i^\vee \rangle = 0$ (e.g., Lemma 2.3 of [**St**]). In that case, we have $\langle \nu+\alpha+\rho, \alpha_i^\vee \rangle = -1$ and $s_i(\nu+\alpha+\rho) = \nu+\rho$. It follows that $\chi(\nu+\alpha) = -\chi(\nu)$, so we must decrement m_0 by the number of short simple roots α_i such that $\langle \nu, \alpha_i^\vee \rangle = 0$. □

B. Double specialization of Weyl characters.

Let $\Lambda^\vee = \{\lambda \in V : \alpha \in \Phi \Rightarrow \langle \lambda, \alpha \rangle \in \mathbf{Z}\}$ denote the co-weight lattice and $\theta_1, \ldots, \theta_n$ the fundamental co-weights. Since $\langle \omega_i, \theta_j \rangle$ is the inverse of the Cartan matrix, it follows that $\langle \lambda, \mu \rangle \in (1/f)\mathbf{Z}$ for all $\lambda \in \Lambda$ and $\mu \in \Lambda^\vee$.

For each $\mu \in \Lambda^\vee$, let us define a ring homomorphism from R (the group ring of Λ as in §5) to $\mathbf{Z}[q^{\pm 1/f}]$ by assigning

$$e^\lambda \mapsto q^{\langle \lambda, \mu \rangle} \qquad (\lambda \in \Lambda).$$

We will use the notations $\chi(\lambda; \mu)$, $\Delta(\lambda; \mu)$, and $M(\lambda; \mu)$ for the respective images of $\chi(\lambda)$, $\Delta(\lambda)$, and $M(\lambda)$ under this "specialization" map.

For $\mu \in \Lambda^\vee$, let $\chi^\vee(\mu) = \Delta(\mu + \rho^\vee)/\Delta(\rho^\vee)$ denote the corresponding Weyl character for the co-root system $\Phi^\vee$, where

$$\rho^\vee = \sum_{\alpha \in \Phi^+} \alpha^\vee/2 = \theta_1 + \cdots + \theta_n.$$

Theorem 7.4 (Double Specialization). *For all $\lambda \in \Lambda$ and $\mu \in \Lambda^\vee$, we have*

$$\chi(\lambda; \mu + \rho^\vee) = q^{\langle \mu, \rho \rangle - \langle \lambda, \rho^\vee \rangle} \chi^\vee(\mu; \lambda + \rho) \cdot \prod_{\alpha \in \Phi^+} \frac{q^{\langle \lambda + \rho, \alpha^\vee \rangle} - 1}{q^{\langle \mu + \rho^\vee, \alpha \rangle} - 1}. \tag{7.2}$$

Proof. From (5.1), it is clear that

$$\begin{aligned} \Delta(\rho; \mu + \rho^\vee) &= q^{-\langle \mu + \rho^\vee, \rho \rangle} \prod_{\alpha \in \Phi^+} \bigl(q^{\langle \mu + \rho^\vee, \alpha \rangle} - 1\bigr), \\ \Delta(\rho^\vee; \lambda + \rho) &= q^{-\langle \lambda + \rho, \rho^\vee \rangle} \prod_{\alpha \in \Phi^+} \bigl(q^{\langle \lambda + \rho, \alpha^\vee \rangle} - 1\bigr). \end{aligned}$$

Furthermore,

$$\begin{aligned} \Delta(\lambda + \rho; \mu + \rho^\vee) &= \sum_{w \in W} \operatorname{sgn}(w) q^{\langle w(\lambda + \rho), \mu + \rho^\vee \rangle} \\ &= \sum_{w \in W} \operatorname{sgn}(w) q^{\langle \lambda + \rho, w^{-1}(\mu + \rho^\vee) \rangle} = \Delta(\mu + \rho^\vee; \lambda + \rho). \end{aligned}$$

Hence

$$\chi(\lambda; \mu + \rho^\vee) = \frac{\Delta(\lambda + \rho; \mu + \rho^\vee)}{\Delta(\rho; \mu + \rho^\vee)} = \frac{\Delta(\mu + \rho^\vee; \lambda + \rho)}{\Delta(\rho; \mu + \rho^\vee)} = \chi^\vee(\mu; \lambda + \rho)) \cdot \frac{\Delta(\rho^\vee; \lambda + \rho)}{\Delta(\rho; \mu + \rho^\vee)},$$

and the result follows. □

We remark that Cherednik [**Ch**] has proved a generalization of the Double Specialization formula for Macdonald's (q, t)-deformation of Weyl characters.[6]

In the special case $\mu = 0$, we obtain the well-known Principal Specialization of $\chi(\lambda)$; the coefficients of $\chi(\lambda; \rho^\vee)$ (as a Laurent polynomial in $q^{1/2}$) are the dimensions of the weight spaces of U_λ relative to the principal embedding of sl_2.

[6] I would like to thank I. G. Macdonald for bringing this to my attention.

Corollary 7.5. *For all $\lambda \in \Lambda$, we have*

$$\chi(\lambda;\rho^\vee) = q^{-\langle \lambda,\rho^\vee\rangle} \prod_{\alpha\in\Phi^+} \frac{q^{\langle \lambda+\rho,\alpha^\vee\rangle}-1}{q^{\langle \rho,\alpha^\vee\rangle}-1}.$$

The appearance of the expression $\langle \rho,\alpha^\vee\rangle$ here, rather than $\langle \rho^\vee,\alpha\rangle$, is justifiable in view of the fact that $\chi(0;\rho^\vee)=1$.

In case Φ is irreducible (for simplicity) and has two orbits of roots, then ρ and $\rho^\vee$ are linearly independent, and the specialization $\chi(\lambda;\rho)$ also factors in closed form. However, the expression $\chi(\lambda;\rho)$ is "unclean" in the sense that ρ need not be an integral co-weight, and $\chi(\lambda;\rho)$ depends on the choice of coordinates (e.g., the lengths of roots).

To present this second specialization in a clean way, we can choose an integral co-weight proportional to ρ as follows. Let $\Phi = \Phi_l \cup \Phi_s$ denote the partition of Φ into orbits of long and short roots, and define

$$\rho_s = \sum_{\alpha\in\Phi_s^+} \alpha/2, \quad \rho_s^\vee = \sum_{\alpha\in\Phi_s^+} \alpha^\vee/2.$$

Similarly define $\rho_l, \rho_l^\vee$ as sums over Φ_l^+. It is not hard to show that ρ_s and $\rho_s^\vee$ are the sums of the fundamental weights ω_i and co-weights θ_i such that α_i is short, hence both are dominant. Furthermore, if r denotes the squared length-ratio of long roots versus short roots, then $r=2$ or 3, and

$$(1/2)c^2\rho^\vee = r\rho_s + \rho_l = (r-1)\rho_s + \rho \in \Lambda,$$

where c denotes the length of a long root. Dually, $r\rho_l^\vee + \rho_s^\vee$ is proportional to ρ.

Given that $\rho^\vee$ is proportional to $r\rho_s + \rho_l$, a straightforward application of (5.1) yields that

$$\chi((r-1)\rho_s) = \prod_{\alpha\in\Phi_s^+} \frac{e^{r\alpha/2}-e^{-r\alpha/2}}{e^{\alpha/2}-e^{-\alpha/2}},$$

and there is a dual version for $\chi^\vee((r-1)\rho_l^\vee)$.

By setting $\mu = (r-1)\rho_l^\vee$ in (7.2), we obtain

Corollary 7.6. *Assume Φ is irreducible and let $r \geqslant 1$ denote the squared length-ratio of long and short roots. For all $\lambda\in\Lambda$, we have*

$$\chi(\lambda;\rho_s^\vee + r\rho_l^\vee) = q^{-\langle \lambda,\rho_s^\vee + r\rho_l^\vee\rangle} \prod_{\alpha\in\Phi_s^+} \frac{q^{\langle \lambda+\rho,\alpha^\vee\rangle}-1}{q^{\langle \rho,\alpha^\vee\rangle}-1} \prod_{\alpha\in\Phi_l^+} \frac{q^{r\langle \lambda+\rho,\alpha^\vee\rangle}-1}{q^{r\langle \rho,\alpha^\vee\rangle}-1}.$$

C. The support of a tensor product.

Fix $\mu, \nu \in \Lambda^+$. Since the partial ordering of Λ respects addition, every term e^λ appearing in $\chi(\mu)\chi(\nu)$ satisfies $\lambda \leqslant \mu + \nu$. It follows that for dominant λ, the multiplicity of $\chi(\lambda)$ in $\chi(\mu)\chi(\nu)$ is nonzero only if $\lambda \leqslant \mu + \nu$. If equality occurs, the multiplicity is 1.

Some further restrictions on λ can be obtained through the use of duality.

For all $\lambda \in \Lambda$, let $\lambda^* = -w_0(\lambda)$, where w_0 denotes the longest element in W. Since w_0 and -1 both act as order-reversing involutions on $(\Lambda, <)$, it follows that $\lambda \mapsto \lambda^*$ is an automorphism, and λ^* is dominant if λ is dominant. Furthermore, $\chi(\lambda^*)$ is the image of $\chi(\lambda)$ under the automorphism of R defined by $e^\mu \mapsto e^{-\mu}$. This reflects the fact that U_{λ^*} is isomorphic to the dual representation U_λ^*.

Since the multiplicity of U_λ in $U_\mu \otimes U_\nu$ is the dimension of the subspace of invariants in $U_\lambda^* \otimes U_\mu \otimes U_\nu$, it follows that $c_{\mu,\nu}^\lambda$ is a symmetric function of λ^*, μ, ν. This symmetry can also be explained by a direct calculation as follows. For any $g \in R^W$, the multiplicity of $\chi(\lambda)$ in g is expressible as

$$\begin{aligned}[e^{\lambda+\rho}]\Delta(\rho)g &= \operatorname{sgn}(w)[e^{w(\lambda+\rho)}]\Delta(\rho)g = \operatorname{sgn}(w)[e^0]e^{-w(\lambda+\rho)}\Delta(\rho)g \\ &= \frac{1}{|W|}[e^0]\Delta(-\lambda-\rho)\Delta(\rho)g = \frac{1}{|W|}[e^0]\Delta(\rho)\Delta(-\rho)\chi(\lambda^*)g,\end{aligned}$$

where in the first equality, w denotes an arbitrary member of W, and the third equality is obtained by averaging over $w \in W$. In particular, we obtain

$$c_{\mu,\nu}^\lambda = \frac{1}{|W|}[e^0]\Delta(\rho)\Delta(-\rho)\chi(\lambda^*)\chi(\mu)\chi(\nu), \tag{7.3}$$

confirming the symmetry.

Applying these symmetries to the inequality $\lambda \leqslant \mu + \nu$, we obtain $\mu^* \leqslant \lambda^* + \nu$ and $\nu^* \leqslant \mu + \lambda^*$, or equivalently $\mu - \nu^* \leqslant \lambda$ and $\nu - \mu^* \leqslant \lambda$. Hence,

$$c_{\mu,\nu}^\lambda \neq 0 \;\Rightarrow\; \mu + \nu \geqslant \lambda \geqslant \mu - \nu^*, \nu - \mu^* \quad (\lambda, \mu, \nu \in \Lambda^+).$$

For any $\xi \in \Lambda$, let ξ^+ denote the unique dominant member of the W-orbit of ξ. The following result provides an even sharper set of bounds for the support of a tensor product.

Proposition 7.7. *If $\lambda, \mu, \nu \in \Lambda^+$ and $c_{\mu,\nu}^\lambda \neq 0$, then $\lambda \in \Pi(\mu,\nu)$, where*

$$\Pi(\mu,\nu) = \{\lambda \in \Lambda^+ : \mu + \nu \geqslant \lambda \geqslant (\mu - \nu^*)^+,\ \mu \geqslant (\lambda - \nu)^+,\ \nu \geqslant (\lambda - \mu)^+\}.$$

The proof relies on the following well-known result (e.g., see Exercise 24.12 of [**H1**]). We include a proof for the sake of completeness.

Lemma 7.8. *If $\lambda, \mu, \nu \in \Lambda^+$ and $c^\lambda_{\mu,\nu} \neq 0$, then $K_{\mu,\lambda-\nu} \neq 0$.*

Proof. By Theorem 7.1, we have

$$c^\lambda_{\mu,\nu} = \sum_{w \in W} \operatorname{sgn}(w) K_{\mu,\delta(w)},$$

where $\delta(w) = w(\lambda + \rho) - (\nu + \rho)$. Hence $c^\lambda_{\mu,\nu} \neq 0$ implies $K_{\mu,\delta(w)} \neq 0$ for some element $w \in W$. Among all w with this property, choose one that minimizes length. To complete the proof, it suffices to show that $w = 1$; i.e., $\delta(w) = \lambda - \nu$.

If $w \neq 1$, there is a simple reflection s_i such that $\ell(s_i w) < \ell(w)$. Since weight multiplicities are W-invariant, it follows that both $\delta(w)$ and

$$s_i \delta(w) = \delta(w) + (m + n)\alpha_i$$

are weights of U_μ, where $m = -\langle w(\lambda + \rho), \alpha_i^\vee \rangle$ and $n = \langle \nu + \rho, \alpha_i^\vee \rangle$. However, both m and n are positive (the former by (1.1); the latter since ν is dominant), so $\delta(s_i w) = \delta(w) + m\alpha_i$ is in the α_i-weight string from $\delta(w)$ to $s_i(\delta(w))$, and hence must also be a weight of U_μ (e.g., [**H1**, §21.3]), contrary to the choice of w. □

Proof of Proposition 7.7. Since weight multiplicities are W-invariant and μ is the highest weight of U_μ, Lemma 7.8 implies $\mu \geqslant (\lambda - \nu)^+$, along with $\nu \geqslant (\lambda - \mu)^+$ by symmetry. Also, by replacing (λ, μ, ν) with (μ, λ, ν^*) (which is justified by (7.3)), we obtain $\lambda \geqslant (\mu - \nu^*)^+$. □

We remark that by the PRV "Conjecture" (for a recent proof see [**L1**]), one knows that $\chi(\lambda)$ occurs with positive multiplicity in $\chi(\mu)\chi(\nu)$ when $\lambda = (\mu - \nu^*)^+$. In view of Proposition 7.7, if follows that this is the unique minimal constituent of $\chi(\mu)\chi(\nu)$ with respect to $<$.

On the other hand, the converse of Proposition 7.7 is far from true; there are easy examples with $\lambda \in \Pi(\mu, \nu)$ and $c^\lambda_{\mu,\nu} = 0$. In fact, determining necessary and sufficient conditions for the nonvanishing of $c^\lambda_{\mu,\nu}$ is a notoriously difficult problem. Even in type A, there are no *simple* conditions, but with the recent solution of the Saturation Conjecture by Knutson and Tao [**KT**], there is now a remarkable recursive set of conditions. For example, see the survey article by Fulton [**F**].

D. The qtensor *algorithm.*

We are now ready to describe a new algorithm we call *qtensor*[7] for computing tensor product multiplicities. Unlike algorithms based on the Brauer-Klimyk formula, it does not require one of the factors to be "small." Instead, the algorithm makes use of a third representation as a kind of "catalyst," and this catalyst must be small. In the case of E_8, even the trivial representation turns out to be a surprisingly powerful catalyst.

[7] The "q" stands for both q and "quick-and-dirty."

Given $\mu, \nu \in \Lambda^+$, the *qtensor* algorithm starts by constructing the set $\Pi(\mu, \nu)$. This can be done by modifying the algorithm in §5D for computing $\Lambda^+(\mu + \nu)$, discarding all $\lambda \notin \Pi(\mu, \nu)$, and stopping as soon as there is no remaining way to apply an "elementary move" and remain above $(\mu - \nu^*)^+$.

Now consider the finite expansion

$$\chi(\mu)\chi(\nu) = \sum_{\lambda \in \Pi(\mu,\nu)} c^{\lambda}_{\mu,\nu} \chi(\lambda).$$

Since the specialization maps are ring homomorphisms, it follows that

$$\chi(\mu; \xi + \rho^\vee)\chi(\nu; \xi + \rho^\vee) = \sum_{\lambda \in \Pi(\mu,\nu)} c^{\lambda}_{\mu,\nu}\, \chi(\lambda; \xi + \rho^\vee), \tag{7.4}$$

for all co-weights ξ.

Now the Double Specialization Formula (Theorem 7.4) comes into play. Suppose that the catalyst ξ is the dominant co-weight for a (dual) Weyl character that has a "sufficiently small" weight system. In that case, we can compute any specialization of the form $\chi^\vee(\xi; \lambda + \rho)$ with $\lambda \in \Lambda^+$ reasonably fast. Hence by the Double Specialization Formula, we can compute every specialization of the form $\chi(\lambda; \xi + \rho^\vee)$ reasonably fast, including all of the terms appearing in (7.4). In particular, note that a fast way to simplify the product over Φ^+ appearing in (7.2) is to recognize it as a quotient of cyclotomic polynomials $\phi_d(q)$. Rewrite each factor $q^m - 1$ symbolically as the product of $\phi_d(q) : d \mid m$, and then cancel common factors syntactically.

Bearing in mind that specializations of Weyl characters are Laurent polynomials, equation (7.4) amounts to a system of linear equations for the unknown multiplicities $c^{\lambda}_{\mu,\nu}$. Furthermore, it is not hard to show that these multiplicities are uniquely determined by the equations corresponding to sufficiently many choices for ξ. Indeed, this is equivalent to the assertion that a Laurent polynomial g in n variables is determined by sufficiently many specializations $g(q^{a_1}, \ldots, q^{a_n})$.

The main question, one we have only started investigating, is whether the number of specializations needed, and their cost, makes this algorithm practical for large problems. While a simple counting argument shows that it is impossible (outside of A_1) for the system of equations corresponding to a single catalyst to solve every instance of the problem, in practice we have found that these systems of equations are amazingly powerful.

In Table VI, we illustrate the results of using *only* the trivial catalyst $\xi = 0$; i.e., the principal specialization $\chi(\lambda; \rho^\vee)$ of Corollary 7.5. For each root system, we generated pairs $\mu, \nu \in \Lambda^+$ in order of increasing complexity. We then found the first instance for which the decomposition of $\chi(\mu)\chi(\nu)$ was not uniquely determined by the principal specialization. By "first," we mean that among all examples, we minimized the complexity of the larger of the two factors, and then among these, we minimized the complexity of the smaller factor. In some cases there is more than one such choice.

	μ	$\lvert\Lambda^+(\mu)\rvert$	ν	$\lvert\Lambda^+(\nu)\rvert$	$\lvert\Pi(\mu,\nu)\rvert$
A_4	1001	2	1001	2	6
B_5	$1\langle 0001$	2	$1\langle 0001$	2	20
C_5	$0\rangle 0011$	3	$0\rangle 0011$	3	16
D_6	$\substack{0\\0}0100$	2	$\substack{0\\0}0100$	2	11
E_6	$01\overset{0}{0}00$	2	$00\overset{0}{0}10$	2	9
E_7	$01\overset{0}{0}001$	7	$01\overset{0}{0}001$	7	71
E_8	$01\overset{0}{0}0001$	19	$00\overset{0}{0}0101$	17	164
F_4	$10\langle 01$	6	$10\langle 01$	6	25
G_2	$1\langle 1$	5	$1\langle 1$	5	14

TABLE VI: Failures of the catalyst $\xi = 0$.

It should be emphasized that these small failures do not represent the upper bound of what can be done using $\xi = 0$ alone. In most cases, there are much larger tensor products that can be successfully decomposed in this way. In type A however, it is not hard to show that the catalyst $\xi = 0$ is powerful enough to decompose the tensor products of the fundamental representations, but not much more. Of course in type A, there are numerous small representations available for use as catalysts.

The impressive results for E_8 can be explained heuristically as follows. The degree of $\chi(\lambda;\rho^\vee)$ as a Laurent polynomial in $q^{1/2}$ is $\langle\lambda, 2\rho^\vee\rangle$. A necessary condition for $\xi = 0$ to fail as a catalyst is that there must exist a distinct pair $\lambda, \lambda' \in \Pi(\mu,\nu)$ whose corresponding specializations have the same degree; i.e., $\langle\lambda,\rho^\vee\rangle = \langle\lambda',\rho^\vee\rangle$. However in the case of E_8, if $\lambda = m_1\omega_1 + \cdots + m_8\omega_8$, then

$$\langle\lambda, 2\rho^\vee\rangle = 92m_1 + 136m_2 + 182m_3 + 270m_4 + 220m_5 + 168m_6 + 114m_7 + 58m_8.$$

The lowest dominant weights yielding the same degree are quite high.

On the other hand, the "disappointing" results reported in Table VI that occur in types A, D, and E_6 can be attributed to the existence of non-trivial Dynkin diagram automorphisms. Any pair of weights that are related by such an automorphism will have the same specialization at $\rho^\vee$. Therefore if $\Pi(\mu,\nu)$ contains such a pair, $\xi = 0$ will fail. To break the symmetry, one needs to include catalysts ξ that are not fixed by automorphisms. Fortunately in each of these cases, there are minuscule weights with this property.

In Table VII, we report the smallest unsolvable problems obtained after adding the catalyst $\xi = (r-1)\rho_l^\vee$ (see Corollary 7.6) to $\xi = 0$, for the multiply-laced root systems. Note the dramatic improvement that occurs in comparison with $\xi = 0$ alone, particularly in the exceptional cases.

	μ	$\lvert\Lambda^+(\mu)\rvert$	ν	$\lvert\Lambda^+(\nu)\rvert$	$\lvert\Pi(\mu,\nu)\rvert$
B_5	$1\langle 0011$	6	$1\langle 0002$	4	64
C_5	$0\rangle 0102$	8	$0\rangle 0110$	6	42
F_4	$10\langle 02$	14	$10\langle 02$	14	76
G_2	$4\langle 2$	23	$4\langle 2$	23	75

TABLE VII: Failures of the combination $\xi = 0$ and $\xi = (r-1)\rho_l^\vee$.

Software

[**1**] coxeter & weyl: `http://www.math.lsa.umich.edu/~jrs/maple.html`

[**2**] Maple: `http://www.maplesoft.com/`

[**3**] LiE: `http://wallis.univ-poitiers.fr/~maavl/LiE/`

[**4**] CHEVIE: `http://www.math.rwth-aachen.de/~CHEVIE/`

[**5**] GAP: `http://www.math.rwth-aachen.de/GAP/`

[**6**] Schur: `http://smc.vnet.net/Schur.html`

References

[**BLS**] L. Babai, E. M. Luks, and Á. Seress, Fast management of permutation groups I, *SIAM J. Comput.* **26** (1997), 1310–1342.

[**Bo**] N. Bourbaki, “Groupes et Algèbres de Lie, Chp. IV–VI,” Hermann, Paris, 1968.

[**BH**] B. Brink and R. Howlett, A finiteness property and an automatic structure for Coxeter groups, *Math. Ann.* **296** (1993), 179–190.

[**B**] R. Brauer, Sur la multiplication des charactéristiques des groupes continus et semi-simples, *C. R. Acad. Sci. Paris* **204** (1937), 1784–1786.

[**Br**] A. Broer, The sum of generalized exponents and Chevalley's restriction theorem for modules of covariants, *Indag. Math.* **6** (1995), 385–396.

[**C**] W. A. Casselman, Machine calculations in Weyl groups, *Invent. Math.* **116** (1994), 95–108.

[**Ch**] I. Cherednik, Macdonald's evaluation conjectures and difference Fourier transform, *Invent. Math.* **122** (1995), 119–145.

[**F**] W. Fulton, Eigenvalues, invariant factors, highest weights, and Schubert calculus, *Bull. Amer. Math. Soc.* **37** (2000), 209–249.

[**GM**] M. Geck and J. Michel, “Good” elements of finite Coxeter groups and representations of Iwahori-Hecke algebras *Proc. London Math. Soc. (3)* **74** (1997), 275–305.

[**GP**] M. Geck and G. Pfeiffer, On the irreducible characters of Hecke algebras, *Advances in Math.* **102** (1993), 79–94.

[**G**] R. Gill, On posets from conjugacy classes of Coxeter groups, *Discrete Math.* **216** (2000), 139–152.

[**Gu**] R. K. Gupta, Characters and the q-analog of weight multiplicity, *J. London Math. Soc.* **36** (1987), 68–76.

[**H1**] J. E. Humphreys, "Introduction to Lie Algebras and Representation Theory," Springer Verlag, Berlin-New York, 1972.

[**H2**] J. E. Humphreys, "Reflection groups and Coxeter groups," Cambridge Univ. Press, Cambridge, 1990.

[**K**] M. Kashiwara, Crystalizing the q-analogue of universal enveloping algebras, *Comm. Math. Phys.* **133** (1990), 249–260.

[**Ka**] S. Kato, Spherical functions and a q-analogue of Kostant's weight multiplicity formula, *Invent. Math.* **66** (1982), 461–468.

[**Kl**] A. U. Klimyk, Decomposition of a direct product of irreducible representations of a semisimple Lie algebra into a direct sum of irreducible representations, *Amer. Math. Soc. Transl., Series 2* **76** (1968), 63–73.

[**KT**] A. Knutson and T. Tao, The honeycomb model of $GL_n(C)$ tensor products I: proof of the saturation conjecture, *J. Amer. Math. Soc.* **12** (1999), 1055–1090.

[**L1**] P. Littelmann, A Littlewood-Richardson rule for symmetrizable Kac-Moody algebras, *Invent. Math.* **116** (1994), 329–346.

[**L2**] P. Littelmann, Paths and root operators in representation theory, *Ann. of Math.* **142** (1995), 499–525.

[**Lu**] G. Lusztig, Singularities, character formulas, and a q-analog of weight multiplicities, *in* "Analysis and topology on singular spaces II–III," *Astérisque* **101–102** (1983), 208–229.

[**MP**] R. V. Moody and J. Patera, Fast recursion formula for weight multiplicities, *Bull. Amer. Math. Soc.* **7** (1982), 237–242.

[**S**] C. C. Sims, Computational methods in the study of permutation groups, *in* "Computational Problems in Abstract Algebra," (Proc. Conf., Oxford, 1967) Pergamon, Oxford, 169–183.

[**St**] J. R. Stembridge, The partial order of dominant weights, *Advances in Math.* **136** (1998), 340–364.

Department of Mathematics, University of Michigan,
Ann Arbor, Michigan 48109–1109 USA
E-mail address: jrs@umich.edu