JOURNAL OF COMPUTATIONAL PHYSICS 7, 346–353 (1971)

Notes

Generation of the Weyl Group on a Computer

I. Introduction

The Weyl group of a root system in a finite-dimensional vector space is a group generated by reflections. These groups are important in a number of different applications. They are a special kind of reflection group [9], and the information gained from studying their generation can be used in studying presentations of reflection groups. The Weyl group occurs in the theories of symmetric spaces and real simple Lie algebras. It is essential in the computational aspects of the representation theory of complex simple Lie algebras. For example, the Weyl group appears in Kostant's formula [10] for computing multiplicities of weights of irreducible representations of complex simple Lie algebras; it also enters into the Kostant–Steinberg formula for obtaining the multiplicities of the irreducible components in the tensor product of such representations.

This paper describes a method for generating the Weyl group on a computer. Section II formulates the definition of the Weyl group from the Bourbaki point of view [5, 11]. For the more traditional formulation of the Weyl group the reader is referred to [1–3, 10]. Section III presents the method for generating the Weyl group on a computer. We have implemented this method on an IBM 360/75 system with a Fortran IV program.

II. The Weyl Group

We start by defining a root system. The Weyl group will act on the root system in a way that is easily identifiable and simply converted to an algorithm for the computer program.

In this paper, V denotes a finite-dimensional complex vector space. If R is a finite subset of V which generates V, then for each nonzero element α in V there exists at most one linear transformation s_α mapping V onto itself and satisfying the following properties:

(i) $s_\alpha(\alpha) = -\alpha$.

(ii) The subset of elements β of V for which $s_\alpha(\beta) = \beta$ is a hyperplane in V.

(iii) $s_\alpha(R) = R$.

A linear transformation s_α of V onto itself satisfying properties (i) and (ii) is called a *reflection associated with* α.

A subset R of V is called a *root system* in V if it satisfies the following properties:

(iv) R is finite, generates V, and does not contain 0.

(v) For each $\alpha \in R$, there exists a reflection s_α associated with α which satisfies (i), (ii), and (iii).

(vi) For each α and β in R, $s_\alpha(\beta) - \beta$ is an integer multiple of α.

The dimension of V is called the *rank* of R.

Let R be a root system in V. A subset $S = \{\alpha_1, \alpha_2, ..., \alpha\ \}$ of R is called a *simple root system* if it satisfies the following properties:

(vii) S is a basis for V.

(viii) Every element β of R can be written as $\beta = \sum_{i=1}^{n} m_i\alpha_i$, where the m_i are integers which are all nonnegative or all nonpositive. The α_i are then called *simple roots.*

Let R be a root system in V and let $S \subseteq R$ be a simple root system. The *Weyl group* $W(R)$ of R is the group generated by the reflections s_{α_i}, $\alpha_i \in S$. It contains all reflections s_β where $\beta \in R$.

It can be shown that if R is a root system in V there exists a symmetric positive definite bilinear form (,) on V which is invariant under the Weyl group $W(R)$. This inner product makes V into a complex Euclidean space, and the elements of $W(R)$ are then orthogonal linear transformations. Thus, each element of $W(R)$ has determinant ± 1. Moreover, it is not difficult to show that for each $\alpha \in R$ we have

$$s_\alpha(\beta) = \beta - \frac{2(\alpha, \beta)}{(\alpha, \alpha)}\alpha \qquad \text{for each } \beta \in V. \tag{1}$$

Let aut(R) denote the set of automorphisms of V which leave R invariant and let aut(S) denote the set of automorphisms of V which leave both R and S invariant. Note that aut(S) is a subgroup of aut(R).

THEOREM 1. *$W(R)$ is a normal subgroup of* aut(R).

Proof. We check normality on the generators of $W(R)$. Let $\alpha \in S$ and $t \in$ aut(R). Then from (1) it follows that $ts_\alpha t^{-1} = s_{t(\alpha)} \in W(R)$.

THEOREM 2. aut(R) $= W(R) \cdot$ aut(S). *That is, any* $t \in$ aut(R) *can be uniquely written as a product of an element in $W(R)$ and an element in* aut(S). ($W(R) \cdot$ aut(S) *is often called the* semi-direct product *of $W(R)$ and* aut(S).)

Proof. We need the following two results whose proofs can be found in Serre [11].

LEMMA 1. *If S and S' are simple root systems of R, there is an element w of the Weyl group such that $w(S) = S'$.*

LEMMA 2. *If S is a simple root system and $w \in W(R)$ is such that $w(S) = S$, then $w =$ identity.*

Now let $t \in \text{aut}(R)$. Then $t(S)$ is a simple root system of R. Let $w \in W(R)$ be such that $w(t(S)) = S$. Then $wt \in \text{aut}(S)$ and $t = w^{-1}s$ where $s \in \text{aut}(S)$. For uniqueness, let $t = ws = w's'$ where $w, w' \in W(R)$ and $s, s' \in \text{aut}(S)$. Then

$$s's^{-1} = w(w')^{-1} \in \text{aut}(S) \cap W$$

and from Lemma 2, $s's^{-1} = \text{identity} = w(w')^{-1}$. Consequently, $s = s'$ and $w = w'$.

Let $S = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ be a simple root system of the root system R. We denote s_{α_i}, by s_i for $i = 1, ..., n$.

THEOREM 3. *A presentation of the Weyl group W of R is*

$$\text{generators: } s_1, ..., s_n \text{ and relations: } (s_i s_j)^{p_{ij}} = \text{identity},$$

where

$$p_{ij} = \begin{cases} 1 & \text{if } i = j \\ 2 & \text{if } i \neq j \quad \text{and} \quad (\alpha_i, \alpha_j) = 0 \\ 3 & \text{if } i \neq j \quad \text{and} \quad (\alpha_i, \alpha_j) = -1 \\ 4 & \text{if } i \neq j \quad \text{and} \quad (\alpha_i, \alpha_j) = -2 \\ 6 & \text{if } i \neq j \quad \text{and} \quad (\alpha_i, \alpha_j) = -3. \end{cases}$$

Proof. See Seminaire Chevalley [8].

From this presentation, it is seen that the class of Weyl groups is a subclass of the class of reflection groups [5, 9, 11].

THEOREM 4. *A presentation of the symmetric group S_n is*

$$\text{generators: } s_1, ..., s_n \text{ and relations } (s_i s_j)^{p_{ij}} = \text{identity},$$

where

$$p_{ij} = \begin{cases} 1 & \text{if } i = j \\ 3 & \text{if } |i - j| = 1 \\ 2 & \text{if } |i - j| \neq 0 \text{ or } 1. \end{cases}$$

Proof. See Coxeter and Moser [9].

If L is a complex simple Lie algebra of rank n and H is a Cartan subalgebra of L, then H is a finite-dimensional complex vector space and its dual H^* can be used as the vector space V in the above discussion. The inner product in H^* is obtained as follows: If (,) is the Killing form in L, then for each $\alpha \in H^*$ there exists a unique $h_\alpha \in H$ such that $\alpha(h) = (h_\alpha, h)$. We then define $(\alpha, \beta) = (h_\alpha, h_\beta)$ for each α and β in H^*.

The simple root systems of complex Lie algebras can be classified by their Dynkin diagrams giving the classical families: A_n, $B_n(n \geqslant 2)$, $C_n(n \geqslant 3)$, and $D_n(n \geqslant 4)$, and exceptional ones: E_6, E_7, E_8, F_4, G_2(cf. [1–3, 10]). It should be noted that the Dynkin diagram is the Coxeter diagram used to classify reflection groups but also includes information about the length of each root. Coxeter diagrams can be defined for reflection groups without reference to root systems.

A result which will be crucial for our generation of the Weyl group on a computer is the following theorem.

THEOREM 5. *If R is a root system of a complex simple Lie algebra of rank n, then $\mathbf{S}_n$ is a subgroup of $W(R)$.*

Proof. This follows by inspecting the root system of each complex simple Lie algebra in turn.

III. Generation of the Weyl Group

To classify root systems of simple Lie algebras by Dynkin diagrams one must construct models of each root system. In these models the root system is a subset of Euclidean space and the Weyl group can, in the case of the classical Lie algebras, be identified as permutations of indices or sign changes of the standard basis for the Euclidean space.

An identification similar to the following example can be made for each of the exceptional algebras.

EXAMPLE. For the root system of the simple Lie algebra F_4, we have $\mathrm{aut}(F_4) = W(F_4)$ since there are no nontrivial automorphisms of the simple root system of F_4. Also, it can be shown that $W(F_4) = \mathrm{aut}(D_4)$ [5]. Then by using Theorem 2, we have $W(F_4) = W(D_4) \cdot \mathbf{S}_3$.

The presentation of $\mathbf{S}_n$ leads directly to the following recursion method for generating it. It is evident that $\mathbf{S}_{n-1}$ is a subgroup of $\mathbf{S}_n$ and is generated by $s_1, \ldots, s_{n-2}$. Let $\mathbf{T}_1 = \mathbf{S}_{n-1}$ and $\mathbf{T}_i = \mathbf{T}_{i-1} s_{n+1-i}$, $i = 2, \ldots, n$. Then $\mathbf{S}_n = \bigcup_{i=1}^{n} \mathbf{T}_i$. This algorithm is very easily programmed and is the basis of the Weyl group algorithm for each of the types of complex simple Lie algebras.

Let $\epsilon_1, \epsilon_2, ..., \epsilon_n$ be an orthonormal basis for an n-dimensional complex Euclidean space. For each simple Lie algebra, a root system and a simple root system can be described in terms of the ϵ_i. For the classical Lie algebras, the Weyl group is given as a group of permutations on the indices of $\epsilon_1, ..., \epsilon_n$ and sign changes on the ϵ_i. For the exceptional ones, another description is provided. Using these descriptions we generate the Weyl group. For the classical ones we first generate $\mathbf{S}_n$ and then construct a representative of each of the cosets in $W/\mathbf{S}_n$; each coset can then be identified with a certain number of sign changes on the ϵ_i.

We list below the specific technique used to generate the Weyl group of each type of Lie algebra.

A_n: The recursion algorithm for the symmetric group is used to generate $\mathbf{S}_{n+1}$.

B_n and C_n: Each coset of $\mathbf{S}_n$ is determined by the element of the Weyl group which changes the sign of each of $\epsilon_{i_1}, ..., \epsilon_{i_k}$ where $i_1 < \cdots < i_k$ and $1 \leqslant k \leqslant n$ or by the identity element, which changes no signs. The element which changes the sign on ϵ_i is $w_i' = s_i s_{i+1} \cdots s_n s_{n-1} \cdots s_i$. However, because of the following two lemmas, the algorithm represents the sign change on ϵ_i by $w_i = s_n s_{n-1} \cdots s_i$ and the sign changes on both ϵ_i and ϵ_j, $i < j$, by $w_{ij} = s_n s_{n-1} \cdots s_i s_n s_{n-1} \cdots s_j$.

LEMMA 3. *If $\mathbf{S}_n$ is presented as in Theorem* 4 *then $\mathbf{S}_n s_i = \mathbf{S}_n$ for any $i = 1, ..., n-1$.*

LEMMA 3B. *Using the presentation for $W(B_n)$ given in Theorem* 3, *we have $w_i' w_j' = w w_{ij}$, where $w \in \mathbf{S}_n$ and $i < j$.*

Proof. We use induction on $n - j$. The following calculations are necessary for the induction step from $n - j = k - 1$ to $n - j = k$.

$$\begin{aligned} s_n s_{n-1} \cdots s_i s_{n-k} s_{n-k+1} \cdots s_n &= s_n s_{n-1} \cdots s_{n-k} s_{n-k-1} s_{n-k} \cdots s_i s_{n-k+1} \cdots s_n \\ &= s_n s_{n-1} \cdots s_{n-k-1} s_{n-k} s_{n-k-1} \cdots s_i s_{n-k+1} \cdots s_n \\ &= s_{n-k-1} s_n s_{n-1} \cdots s_{n-k} s_{n-k-1} \cdots s_i s_{n-k+1} \cdots s_n . \end{aligned}$$

D_n: Each coset of $\mathbf{S}_n$ is determined by the element of the Weyl group which changes the sign of each of $\epsilon_{i_1}, ..., \epsilon_{i_{2k}}$ where $i_1 < \cdots < i_{2k}$ and $1 \leqslant k \leqslant [n/2]$ or by the identity element. The element which changes the signs on both ϵ_i and ϵ_j, $i < j$, is $v'_{ij} = s_j s_{j+1} \cdots s_{n-1} s_i s_{i+1} \cdots s_{n-2} s_n s_{n-2} s_{n-3} \cdots s_i s_{n-1} s_{n-2} \cdots s_j$. Because of Lemma 3, the algorithm represents this element as

$$v_{ij} = s_n s_{n-2} s_{n-3} \cdots s_i s_{n-1} s_{n-2} \cdots s_j .$$

A proof similar to that of Lemma 3B will show

LEMMA 3D. *Using the presentation for $W(D_n)$ given in Theorem 3, we have $v'_{ij}v'_{kl} = vv_{ijkl}$, where $v \in \mathbf{S}_n$, $i < j < k < l$, and*

$$v_{ijkl} = s_n s_{n-2} s_{n-3} \cdots s_i s_{n-1} s_{n-2} \cdots s_j s_n s_{n-2} s_{n-3} \cdots s_k s_{n-1} s_{n-2} \cdots s_l .$$

Consequently, the algorithm can represent $v'_{ij}v'_{kl}$as v_{ijkl}.

E_n: Each of these Weyl groups is too large for the present computer implementation. A possible approach is to generate the cosets of $W(D_5)$ in $W(E_6)$, the cosets of $W(E_6)$ in $W(E_7)$, and the cosets of $W(E_7)$ in $W(E_8)$.

F_4: Each element of $W(F_4)$ is the unique product of an element of $\mathbf{S}_3$ and one of $W(D_4)$. The algorithm first computes $\mathbf{S}_3$ using the symmetric group algorithm. However, instead of denoting the generators by 1 and 2, they are denoted by 3 and 4 so that the resulting words will be written in terms of the four generators for $W(F_4)$ defined by its Coxeter diagram. Next, $W(D_4)$ is generated by its algorithm. Then the generators for $W(D_4)$, s_1, s_2, s_3, s_4, are written in terms of the generators of $W(F_4)$ as: $s_1 = r_3r_4r_3r_2r_3r_4r_3$, $s_2 = r_1$, $s_3 = r_2$, $s_4 = r_3r_2r_3$, where $r_1 ,..., r_4$ are the generators for $W(F_4)$ defined by its Coxeter diagram. Finally, each word in $\mathbf{S}_3$ is adjoined to every word in a copy of $W(D_4)$.

G_2: The simplest way to generate this group, the dihedral group of order 12, is by table look-up.

We next describe the representation of an element in the Weyl group used in our program. Each element of the Weyl group is a finite product of the generators $s_{\alpha_1} ,..., s_{\alpha_n}$ and it is represented in the computer by a string of integers $i_1, i_2 ,..., i_k$ where i_j corresponds to $s_{\alpha_{i_j}}$. Thus, $s_{\alpha_1}s_{\alpha_2}s_{\alpha_3}s_{\alpha_1}s_{\alpha_3}$ is represented by the string 12313. This representation makes it particularly simple to compute the action of an element of the Weyl group on a vector in H^* which is written either in terms of the basis of simple roots $\{\alpha_1, \alpha_2 ,..., \alpha_n\}$ or in terms of the basis of fundamental weights $\{\lambda_1, \lambda_2 ,..., \lambda_n\}$. Thus, let

$$\omega = \sum_{j=1}^{n} m_j a_j = \sum_{j=1}^{n} k_j \lambda_j .$$

Then $s_{\alpha_i}(\omega)$ is computed as

$$i(m_1, m_2 ,..., m_n) = \left(m_1, m_2 ,..., m_i - \sum_{j=1}^{n} a_{ij} m_j ,..., m_n\right),$$

or

$$i(k_1, k_2, \ldots, k_n) = (k_1 - k_i a_{1i}, k_2 - k_i a_{2i}, \ldots, k_n - k_i a_{ni}),$$

where

$$a_{ij} = \frac{2(\alpha_i, \alpha_j)}{(\alpha_i, \alpha_j)}, \qquad \text{the } (i, j) \text{ entry in the Cartan matrix.}$$

To handle large Weyl groups the string could be packed five digits per two bytes instead of the present one digit per two bytes. An alternative approach would be to use the more sophisticated methods of representation discussed by Cannon [6, 7].

The method of storing the Weyl group in the computer is strongly dependent upon its intended use. For example, the entire Weyl group must be available in random access memory to evaluate multiplicities of the weights of an irreducible representation of a complex simple Lie algebra using Konstant's formula. However, to find all the simple root systems of a complex simple Lie algebra each element need only be applied to the given simple root system as it is generated.

As a pilot project, we have developed a program, written in FORTRAN IV for the IBM 360/75 system, to generate the Weyl group of the complex simple Lie algebras of rank at most 5. This program puts the entire Weyl group into the random access core memory.

References

1. J. G. Belinfante, B. Kolman and H. A. Smith, An introduction to Lie groups and Lie algebras, with applications, *SIAM Rev.*, **8** (1966), 11–46.
2. J. G. Belinfante, B. Kolman and H. A. Smith, An introduction to Lie groups and Lie algebras, with applications. II. The basic methods and results of representation theory, *SIAM Rev.* **10**, 1968, 60–95.
3. J. G. Belinfante and B. Kolman, An introduction to Lie groups and Lie algebras, with applications. III. Computational methods and applications of representation theory, *SIAM Rev.* **11**, 1969, 510–543.
4. N. Bourbaki, "Groupes et Algèbres de Lie," Chap. 1, Hermann, Paris, 1960.
5. N. Bourbaki, "Groupes et Algèbres de Lie," Chaps. 4–6, Hermann, Paris, 1968.
6. J. J. Cannon, "Computation in Finite Algebraic Structures," Ph.D. Thesis submitted to Department of Pure Mathematics, University of Sydney, 1969.
7. J. J. Cannon, Some combinatorial and symbol manipulation programs in group theory, *in* "Computational Problems in Abstract Algebra," (J. Leech, Ed.), pp. 199–203. Pergamon Press, Oxford, 1970.
8. Classification de groupes de Lie algébriques *in* "Seminaire Chevalley," Vols. I and II, École Norm. Sup., Paris, 1956–1958.

9. H. M. S. Coxeter and W. O. J. Moser, "Generators and Relations for Discrete Groups," Springer-Verlag, Berlin, 1965.
10. N. Jacobson, "Lie algebras," John Wiley & Sons, New York, 1962.
11. J. P. Serre, "Algèbres de Lie Semi-Simple Complexes," Benjamin, New York, 1966.

Received: June 4, 1970

Robert E. Beck
Department of Mathematics,
Villanova University, Villanova Pennsylvania

Bernard Kolman*
Department of Mathematics,
Drexel University, Philadelphia, Pennsylvania

* The work of Bernard Kolman was partially supported by the Air Force Office of Scientific Research, Office of Aerospace Research, United States Air Force, under AFOSR Grant No. 69-1797.