Branching rules for representations of simple Lie algebras through Weyl group orbit reduction

1989 J. Phys. A: Math. Gen. 22 2329

(http://iopscience.iop.org/0305-4470/22/13/027)

J. Phys. A: Math. Gen. 22 (1989) 2329-2340. Printed in the UK

Branching rules for representations of simple Lie algebras through Weyl group orbit reduction

J Patera† and R T Sharp‡

† Centre de Recherches Mathématiques, Université de Montréal, Montréal, Québec, H3C 3J7 Canada
‡ Physics Department, McGill University, Montreal, Quebec, H3A 2T8 Canada

Received 7 February 1989

Abstract. Two independent algorithms are presented, which together allow the determination of branching rules from an irreducible representation of a compact Lie algebra to those of a subalgebra (or subjoined algebra). The first gives the subalgebra Weyl orbits contained in an algebra orbit. The second gives the irreducible representations of an algebra contained in an orbit, and by inversion of a triangular matrix, the orbits contained in an irreducible representation.

1. Introduction

There are many different ways (Cummins *et al* 1989) of finding branching rules (i.e. reduction) of an irreducible representation (IR) $\phi(G)$ of a particular simple Lie algebra G over the complex number field to IR of a particular reductive Lie subalgebra or subjoined (Patera *et al* 1980) semisimple algebra. (Except where explicitly stated, the term 'subalgebra' includes subjoined algebra throughout this paper.) However, once one is interested in methods of general applicability (as to the type of semisimple Lie algebras involved) there is only one method left although its implementation may proceed in several practically very different ways. The method has three steps: (i) first one has to find the weight system $\Omega(\phi(G))$ of the initial representation $\phi(G)$, then (ii) the weights of $\Omega(\phi(G))$ have to be transformed into the weights of the subalgebra. The result of such a transformation is the weight system $\Omega(\phi(G'))$ of the reducible representation $\phi(G')$ of the subalgebra G'. Finally (iii) $\Omega(\phi(G'))$ has to be sorted out into the weight systems of irreducible components of the branching rule.

The contribution of this paper to the problem of computing the branching rules for semisimple to reductive Lie algebras over the complex number field has two distinct aspects.

(*a*) Rather than reducing the complete weight system $\Omega(\phi(G))$ to $\Omega(\phi(G'))$ we reduce the subsystems of $\Omega(\phi(G))$, called Weyl group orbits, or simply W-orbits $O(\lambda)$, $\lambda \in \Omega(\phi(G))$, to similar subsystems of $\Omega(\phi(G'))$ and subsequently reassemble the appropriate subsystems into complete weight systems of the relevant irreducible representations.

(*b*) The second contribution of this paper is showing how, in principle, all necessary operations on the weight systems and the W-orbits can be performed in terms of generating functions. General procedures for the derivation of the appropriate generating functions are formulated. The advantage offered by the generating functions comes firstly from the possibility of computing by hand particular cases of considerable size (by developing a generating function as far as needed) and secondly from the global information such a generating function provides about the whole problem (integrity bases, syzygies, etc).

The task (i) of computing the weight system $\Omega(\phi(G))$, in particular the weight multiplicities, using any of the existing recursive methods, imposed practical limits on the original implementation of the method. A relatively recent development (Moody and Patera 1982) is the possibility of computing multiplicities of dominant weights without computing the rest of them for a given representation. That opened the further possibility of making the method vastly more efficient by exploiting the decomposition of $\Omega(\phi(G))$ into W-orbits $O(\lambda)$, the multiplicity of a dominant weight λ in $\Omega(\phi(G))$ being the multiplicity of the W-orbit $O(\lambda)$ in $\Omega(\phi(G))$. The economy occurs when branching rules for orbits are calculated separately from branching rules for representations; only subsequently are those, with appropriate multiplicities, collected into branching rules for representations. A possible alternative to recursive computing of weight multiplicities is one of the results of this paper.

The second step (ii) is efficiently performed by the projection matrices introduced by Navon and Patera (1967). Such a matrix has to be calculated only once for each algebra-subalgebra pair and then used on any weight system. They are found in McKay *et al* (1977) (see also Cummins *et al* 1989).

The last step (iii) is quite easy because it is sufficient to retain and to work with the dominant weights of $\Omega(\phi(G'))$, a small fraction of the weight system, each representing precisely one W-orbit. That again hinges on the possibility to reassemble the W-orbits of G′ into whole weight systems of irreducible representations of G′.

Among the methods for branching rules computations the present approach stands in two ways: (*a*) it is the most explicit general solution available, and (*b*) it can be applied, at least in principle, to any algebra-subalgebra pair. Its practical applicability naturally has some limits. Let us point out here that, if one is interested in branching rules for only some classes of irreducible representations, one may extend the practical limits to much larger algebras-subalgebras.

In § 2 a method of implementing step (ii) above is presented; it utilises the orbit-orbit generating function. Section 3 shows how to express a Weyl orbit as a superposition of IR; step (iii) above can thus be accomplished. If the IR, and orbits, are arranged, say, according to level (Bremner *et al* 1985), the transformation matrix from orbits to IR is triangular, and hence easily inverted to express IR in terms of orbits—step (i) above.

In the paper we assume that the reader is familiar with weight lattices of semisimple Lie algebras and with the action of the corresponding Weyl group on the weights. A generic weight is always given by integer coordinates, its 'components', relative to the basis of fundamental weights. We often use IR, or irreducible representation, to mean the weights of the IR in question.

2. Orbit–orbit generating function

We first define the orbit-orbit generating function then show how to determine it. It was first used (for subjoining Sp(6) > SU(4)) by Couture and Sharp (1989).

The orbit-orbit generating function for an algebra-subalgebra pair is a rational function $F(A, B)$ whose power expansion

$$F(A, B) = \sum_{a,b} A^a B^b C_{ab} \tag{2.1}$$

gives the multiplicity C_{ab} of the subalgebra orbit $[b]$ in the algebra orbit $[a]$. Here A^a and B^b mean $\Pi_{i=1}^{k} A_i^{a_i}$ and $\Pi_{i=1}^{k'} B_i^{b_i}$ respectively; k and k' are the respective ranks of algebra and subalgebra; A_i and B_i are dummy variables which carry as exponents the algebra orbit labels a_i and subalgebra orbit labels b_i, respectively. Orbit labels are the components of the highest weight of the orbit. The components of a weight λ are the coefficients λ_i of the fundamental weights M_i in the expansion of λ:

$$\lambda = \Sigma_i M_i \lambda_i \tag{2.2a}$$

or, alternatively,

$$\lambda_i = 2\langle\lambda|\alpha_i\rangle/\langle\alpha_i|\alpha_i\rangle \tag{2.2b}$$

where α_i are the simple roots; the orbit labels are non-negative integers. If one or more orbit labels vanish we say the orbit is degenerate.

We first give a general method of evaluating the orbit-orbit generating function (§§ 2.1 and 2.2). In § 2.4 we discuss the method of elementary orbits, which is simpler in most situations, in particular the equal-rank case.

2.1. *The generating function for weights of a Weyl group orbit*

Define the orbit-weight generating function for an algebra of rank k as

$$H(A, \Lambda) = \sum_{W'} W' \prod_{i=1}^{k} (1 - A_i\Lambda_i)^{-1} \tag{2.3}$$

where A_i are dummies which carry orbit labels as exponents and Λ_i carry weight components λ_i. The sum is over Weyl group elements W' (operating on the weight λ carried by Λ) which do not stabilise λ. The power expansion of the orbit-weight generating function gives the weights contained in each orbit

$$H(A, \Lambda) = \sum_a A^a \sum_\lambda \Lambda^\lambda C_{a\lambda}. \tag{2.4}$$

$C_{a\lambda}$ is one or zero according to whether the weight λ is or is not contained in the orbit $[a]$. The orbit-weight generating function is also the orbit-orbit generating function for the Cartan subalgebra. As examples we give the orbit-weight generating functions for SU(3), SO(5) and SU(4). For SU(3) it is

$$\begin{aligned}
&\frac{1}{(1-A_1\Lambda_1)(1-A_2\Lambda_2)} + \frac{A_1\Lambda_1^{-1}\Lambda_2}{(1-A_2\Lambda_2)(1-A_1\Lambda_1^{-1}\Lambda_2)} + \frac{A_2\Lambda_1^{-1}}{(1-A_1\Lambda_1^{-1}\Lambda_2)(1-A_2\Lambda_1^{-1})} \\
&\quad + \frac{A_1\Lambda_2^{-1}}{(1-A_2\Lambda_1^{-1})(1-A_1\Lambda_2^{-1})} + \frac{A_2\Lambda_1\Lambda_2^{-1}}{(1-A_1\Lambda_2^{-1})(1-A_2\Lambda_1\Lambda_2^{-1})} \\
&\quad + \frac{A_1A_2\Lambda_1^2\Lambda_2^{-1}}{(1-A_2\Lambda_1\Lambda_2^{-1})(1-A_1\Lambda_1)}.
\end{aligned} \tag{2.5}$$

The SO(5) orbit-weight generating function is

$$\frac{1}{(1-A_1\Lambda_1)(1-A_2\Lambda_2)}+\frac{A_1\Lambda_1^{-1}\Lambda_2^2}{(1-A_2\Lambda_2)(1-A_1\Lambda_1^{-1}\Lambda_2^2)}+\frac{A_2\Lambda_1^{-1}\Lambda_2}{(1-A_1\Lambda_1^{-1}\Lambda_2^2)(1-A_2\Lambda_1^{-1}\Lambda_2)}$$
$$+\frac{A_1\Lambda_1^{-1}}{(1-A_2\Lambda_1^{-1}\Lambda_2)(1-A_1\Lambda_1^{-1})}+\frac{A_2\Lambda_2^{-1}}{(1-A_1\Lambda_1^{-1})(1-A_2\Lambda_2^{-1})}$$
$$+\frac{A_1\Lambda_1\Lambda_2^{-2}}{(1-A_2\Lambda_2^{-1})(1-A_1\Lambda_1\Lambda_2^{-2})}+\frac{A_2\Lambda_1\Lambda_2^{-1}}{(1-A_1\Lambda_1\Lambda_2^{-2})(1-A_2\Lambda_1\Lambda_2^{-1})}$$
$$+\frac{A_1A_2\Lambda_1^2\Lambda_2^{-1}}{(1-A_2\Lambda_1\Lambda_2^{-1})(1-A_1\Lambda_1)}. \qquad (2.6)$$

For SU(4) the orbit-weight generating function is

$$\frac{1}{\gamma_1\delta_1\varepsilon_1}+\frac{\gamma_2}{\gamma_2\delta_1\varepsilon_1}+\frac{\delta_2}{\gamma_1\delta_2\varepsilon_1}+\frac{\varepsilon_2}{\gamma_1\delta_1\varepsilon_2}+\frac{\gamma_3}{\gamma_3\delta_2\varepsilon_1}+\frac{\gamma_2\varepsilon_2}{\gamma_2\delta_1\varepsilon_2}+\frac{\delta_3}{\gamma_2\delta_3\varepsilon_1}$$
$$+\frac{\delta_4}{\gamma_1\delta_4\varepsilon_2}+\frac{\varepsilon_3}{\gamma_1\delta_2\varepsilon_3}+\frac{\gamma_3\delta_3}{\gamma_3\delta_3\varepsilon_1}+\frac{\gamma_4}{\gamma_4\delta_4\varepsilon_2}+\frac{\gamma_3\varepsilon_3}{\gamma_3\delta_2\varepsilon_3}+\frac{\delta_5}{\gamma_2\delta_5\varepsilon_2}+\frac{\delta_4\varepsilon_3}{\gamma_1\delta_4\varepsilon_3}$$
$$+\frac{\varepsilon_4}{\gamma_2\delta_3\varepsilon_4}+\frac{\gamma_4\delta_5}{\gamma_4\delta_5\varepsilon_2}+\frac{\gamma_4\varepsilon_3}{\gamma_4\delta_4\varepsilon_3}+\frac{\gamma_3\varepsilon_4}{\gamma_3\delta_3\varepsilon_4}+\frac{\delta_6}{\gamma_3\delta_6\varepsilon_3}+\frac{\delta_5\varepsilon_4}{\gamma_2\delta_5\varepsilon_4}+\frac{\gamma_4\delta_6}{\gamma_4\delta_6\varepsilon_3}$$
$$+\frac{\gamma_4\varepsilon_4}{\gamma_4\delta_5\varepsilon_4}+\frac{\delta_6\varepsilon_4}{\gamma_3\delta_6\varepsilon_4}+\frac{\gamma_4\delta_6\varepsilon_4}{\gamma_4\delta_6\varepsilon_4}. \qquad (2.7)$$

To save space, a symbol (γ_i, δ_i or ε_i) in the denominator of a term in (2.7) stands for one minus that symbol. The symbols in question, in turn, stand for

$$\begin{array}{lll}
\gamma_1=A_1\Lambda_1 & \gamma_2=A_1\Lambda_1^{-1}\Lambda_2 & \gamma_3=A_1\Lambda_2^{-1}\Lambda_3 \\
\gamma_4=A_1\Lambda_3^{-1} & \delta_1=A_2\Lambda_2 & \delta_2=A_2\Lambda_1\Lambda_2^{-1}\Lambda_3 \\
\delta_3=A_2\Lambda_1^{-1}\Lambda_3 & \delta_4=A_2\Lambda_1\Lambda_3^{-1} & \delta_5=A_2\Lambda_1^{-1}\Lambda_2\Lambda_3^{-1} \\
\delta_6=A_2\Lambda_2^{-1} & \varepsilon_1=A_3\Lambda_3 & \varepsilon_2=A_3\Lambda_2\Lambda_3^{-1} \\
\varepsilon_3=A_3\Lambda_1\Lambda_2^{-1} & \varepsilon_4=A_3\Lambda_1^{-1}. &
\end{array} \qquad (2.8)$$

2.2. *General method for orbit-orbit generating function*

The orbit-weight generating function for an algebra may now be converted in two steps into the orbit-orbit generating function for a subalgebra. The first step is to replace the dummies Λ which carry algebra weights in (2.3) or (2.4) with new dummies B which carry the corresponding subalgebra weights b. We give examples below. The second step is to retain that part of the generating function for which the weight components b_i (exponents of B_i) are all non-negative. They are then subalgebra orbit labels and we have the desired orbit-orbit generating function (2.1).

Let us carry out the steps just described to obtain the orbit-orbit generating functions for

$$SU(3)\supset SO(3) \qquad SO(5)\supset SU(2) \qquad SU(4)\supset SU(2)\times SU(2)$$
$$SU(4)\supset SU(2)\times SU(2)\times U(1).$$

The projection matrices of McKay *et al* (1977) may be used to convert algebra to subalgebra weights. In simple cases the prescription can be inferred from branching rules for a low IR of the algebra without turning to the projection matrices.

For SU(3) ⊃ SO(3) the necessary substitutions in (2.5) are $\Lambda_1 \to B^2$, $\Lambda_2 \to 1$. Retaining the non-negative power part in B then yields the SU(3) ⊃ SO(3) orbit-orbit generating function

$$\frac{1}{(1-A_1B^2)(1-A_2B^2)}+\frac{A_2}{(1-A_1B^2)(1-A_2)}+\frac{A_1}{(1-A_2B^2)(1-A_1)}. \tag{2.9}$$

For SO(5) ⊃ SU(2) the substitutions in (2.6) are $\Lambda_1 \to B^4$, $\Lambda_2 \to B^3$. Retaining non-negative powers of B yields the SO(5) ⊃ SU(2) orbit-orbit generating function

$$\begin{aligned}\frac{1}{(1-A_2B^3)(1-A_1B^4)}&+\frac{A_2B}{(1-A_1B^4)(1-A_2B)}+\frac{A_1A_2^2}{(1-A_2B)(1-A_1A_2^2)}\\&+\frac{A_1A_2B}{(1-A_1A_2^2)(1-A_1A_2B)}+\frac{A_1B^2}{(1-A_1A_2B)(1-A_1B^2)}\\&+\frac{A_1A_2B^5}{(1-A_1B^2)(1-A_2B^3)}+\frac{A_1A_2^2}{1-A_1A_2^2}.\end{aligned} \tag{2.10}$$

For SU(4) ⊃ SU(2) × SU(2) (Wigner supermultiplet model) the substitutions in (2.7) are $\Lambda_1 \to B_1B_2$, $\Lambda_2 \to B_2^2$, $\Lambda_3 \to B_1B_2$; the orbit-orbit generating function is

$$\begin{aligned}\frac{1}{(1-A_1B_1B_2)(1-A_2B_2^2)(1-A_3B_1B_2)}&+\frac{A_1A_3B_2^2}{(1-A_1A_3B_2^2)(1-A_2B_2^2)(1-A_3B_1B_2)}\\&+\frac{A_2B_1^2}{(1-A_1B_1B_2)(1-A_2B_1^2)(1-A_3B_1B_2)}\\&+\frac{A_1A_3B_2^2}{(1-A_1B_1B_2)(1-A_2B_2^2)(1-A_1A_3B_2^2)}\\&+\frac{A_1A_3B_1^2}{(1-A_1A_3B_1^2)(1-A_2B_1^2)(1-A_3B_1B_2)}\\&+\frac{A_2}{(1-A_1A_3B_2^2)(1-A_2)(1-A_3B_1B_2)}\\&+\frac{A_2}{(1-A_1B_1B_2)(1-A_2)(1-A_1A_3B_2^2)}\\&+\frac{A_1A_3B_1^2}{(1-A_1B_1B_2)(1-A_2B_1^2)(1-A_1A_3B_1^2)}\\&+\frac{A_1A_2A_3B_1^2}{(1-A_1A_3B_1^2)(1-A_2)(1-A_3B_1B_2)}\\&+\frac{A_1A_2A_3B_1^2}{(1-A_1B_1B_2)(1-A_2)(1-A_1A_3B_1^2)}.\end{aligned} \tag{2.11}$$

For SU(4) ⊃ SU(2) × SU(2) × U(1) the substitutions in (2.7) are $\Lambda_1 \to B_1 Z$, $\Lambda_2 \to Z^2$, $\Lambda_3 \to B_2 Z$, where Z carries the U(1) label. Keeping non-negative powers of B_1, B_2 gives the orbit-orbit generating function

$$\frac{1}{(1-A_1B_1Z)(1-A_2Z^2)(1-A_3B_2Z)}+\frac{A_2B_1B_2}{(1-A_1B_1Z)(1-A_2B_1B_2)(1-A_3B_2Z)}$$
$$+\frac{A_1B_2Z^{-1}}{(1-A_1B_2Z^{-1})(1-A_2B_1B_2)(1-A_3B_2Z)}$$
$$+\frac{A_3B_1Z^{-1}}{(1-A_1B_1Z)(1-A_2B_1B_2)(1-A_3B_1Z^{-1})}$$
$$+\frac{A_1A_3B_1B_2Z^{-2}}{(1-A_1B_2Z^{-1})(1-A_2B_1B_2)(1-A_3B_1Z^{-1})}$$
$$+\frac{A_2Z^{-2}}{(1-A_1B_2Z^{-1})(1-A_2Z^{-2})(1-A_3B_1Z^{-1})}. \tag{2.12}$$

It is useful to interpret an orbit-orbit generating function in terms of an integrity basis, consisting of a finite number of 'elementary orbits'; an elementary orbit (a, b) is a subalgebra orbit $[b]$ belonging to an algebra orbit $[a]$ and cannot be written as a stretched product of lower orbits (a', b') ('stretched product' means that algebra and subalgebra labels are componentwise additive). Any subalgebra orbit belonging to any algebra orbit can be written as a stretched product of elementary orbits. Some pairs of elementary orbits are incompatible—they must not appear in the same product. There exists a polynomial identity (syzygy) relating them. The elementary orbits and the compatibility rules may be read from the orbit-orbit generating function. Thus from (2.9) we read the elementary orbits $\alpha = (10, 2)$, $\beta = (10, 0)$, $\gamma = (01, 2)$, $\delta = (01, 0)$, with compatibility table

	β	γ	δ
α	×	√	√
β		√	×
γ			×

(a cross means incompatible). In (2.10) the elementary orbits are $\alpha = (10, 4)$, $\beta = (10, 2)$, $\gamma = (01, 3)$, $\delta = (01, 1)$, $\varepsilon = (11, 1)$, $\zeta = (12, 0)$, $\zeta' = (12, 0)$. The compatibility table is

	β	γ	δ	ε	ζ	ζ'
α	×	√	√	×	×	×
β		√	×	√	×	×
γ			×	×	×	×
δ				×	√	×
ε					√	×
ζ						×

The elementary orbits and compatibility rules for the orbit-orbit generating functions (2.11) and (2.12) are easily written down—we omit them here.

2.3. *Relative position of Weyl chambers of algebra and subalgebra*

A few general statements can be made about orbit-orbit generating functions, based on the way the Weyl chambers for the subalgebra line up with those of the algebra.

When we compare regions of algebra and subalgebra weight space, a region of subalgebra weight space, say a Weyl chamber, means the region of algebra weight space which projects into the subalgebra region in question.

The simplest situation is that in which Weyl group chambers of algebra and subalgebra 'line up', i.e. each subalgebra Weyl chamber contains only complete algebra Weyl chambers (N/N' of them, where N and N' are, respectively, the orders of the algebra and subalgebra Weyl groups); each boundary of a subalgebra Weyl chamber is also a boundary of an algebra chamber.

In this paragraph algebra and subalgebra Weyl chambers are assumed to line up. Let W be one of the N/N' elements of the algebra Weyl group which carry the dominant algebra chamber within the dominant subalgebra chamber. Then an algebra-subalgebra orbit pair (a, b) corresponding to the term A^aB^b in the orbit-orbit generating function can always be written (a, PWa) corresponding $\Pi_i(A_i \Pi_j B_j^{(PWM_i)_j})^{a_i}$, where P is the projection onto subalgebra weight space. Thus the elementary orbits correspond to $A_i \Pi_j B_j^{(PWM_i)_j}$, i.e. they are the subalgebra orbits contained in the fundamental orbits of the algebra. The compatibility rules for elementary orbits can then be stated as follows: two elementary orbits are compatible if and only if the two weights WM_i and WM_k can be obtained by the same Weyl element W; in particular, two subalgebra orbits belonging to the same fundamental algebra orbit are incompatible. SU(3) ⊃ SO(3) and SU(4) ⊃ SU(2) × SU(2) × U(1) above are examples of algebra and subalgebra Weyl chambers' lining up. A sufficient but not necessary condition for the lining up is that algebra and subalgebra have equal rank; for all known maximal subjoint algebras that is always the case. For a regular subalgebra, the Weyl chambers line up.

Examples of cases where algebra and subalgebra sectors do not line up are SO(5) ⊃ SU(2) and SU(4) ⊃ SU(2) × SU(2) above. When a dominant subalgebra weight lies inside a chamber of algebra weight space that is only partly in the dominant subalgebra sector, it cannot be compounded from elementary orbits belonging to fundamental algebra orbits; hence composite elementary orbits (more than one algebra label non-zero) arise, as in the generating functions (2.10) and (2.11).

2.4. *Method of elementary orbits*

Usually it is easier to find the orbit-orbit generating function heuristically by examining low algebra orbits to find the elementary orbits and their incompatibilities, rather than using the orbit-weight generating function (2.3). With the help of existing tables for branching rules (McKay and Patera 1981) and orbit multiplicities (Bremner *et al* 1985), one proceeds by increasing level within each algebra congruence class as follows:

(i) from known branching rules write the algebra IR corresponding to the orbit to be decomposed as a superposition of subalgebra IR;

(ii) write the subalgebra IR as a sum of subalgebra orbits, and the algebra orbits in the IR in question other than the orbit to be decomposed, as a sum of subalgebra orbits. The result is the algebra orbit in question decomposed into subalgebra orbits. Alternatively, if tables are unavailable, use projection matrices to convert the weights of the algebra orbit to be decomposed into subalgebra weights and retain those in the dominant subalgebra sector.

When algebra and subalgebra ranks are equal the procedure is particularly simple. Again, using branching rule tables, write the algebra IR, corresponding to the desired algebra orbit, in terms of subalgebra IR. Now note the distance squared from the origin of the algebra orbit in question (the distance squared (or 'scalar product' (SP))) is tabulated by Bremner *et al* (1985). Ignore smaller orbits in the algebra IR; on the other side of this equation, retain only subalgebra orbits with the same distance squared as the algebra orbit—that means keeping only the outside orbit of each IR, and that only when it has the correct distance squared. The scale of subalgebra weights may have to be adjusted to match that for algebra weights. According to the discussion of § 2.3 the elementary orbits are all found in the fundamental algebra orbits.

We illustrate the procedure for $F_4 \supset B_4$. From branching rule tables (McKay and Patera 1981) we find

$$\underset{24}{\overset{2}{(0001)}} \supset \underset{16}{\overset{2}{(0001)}} + \underset{8}{\overset{2}{(1000)}} + \overset{0}{(0000)}$$

$$\underset{24}{\overset{4}{(1000)}} \supset \underset{24}{\overset{4}{(0100)}} + \overset{2}{(0001)}$$

$$\underset{96}{\overset{6}{(0010)}} \supset \underset{64}{\overset{6}{(1001)}} + \underset{32}{\overset{6}{(0010)}} + \overset{4}{(0100)} + \overset{2}{(0001)} + \overset{2}{(1000)}$$

$$\underset{96}{\overset{12}{(0100)}} \supset \underset{96}{\overset{12}{(1010)}} + \overset{10}{(0101)} + \overset{6}{(1001)} + \overset{6}{(0010)} + \overset{4}{(0100)}.$$

The number below an IR is the size (number of weights) of the corresponding orbit; the number above is the distance squared of the corresponding orbit, divided by 2 for B_4 to make the scale agree with that of F_4. Following the discussion above we find the elementary orbits to be

$$a = (1000, 0100) \qquad b = (0100, 1010)$$
$$c = (0010, 1001) \qquad d = (0010, 0010)$$
$$e = (0001, 1000) \qquad f = (0001, 0001).$$

Since a and b are the only orbits in their respective fundamental algebra orbits, they are compatible with each other and with all the rest. In the algebra orbit (0011), whose distance squared and orbit size are respectively 14 and 192, the candidate orbits are $ce = (2001)$, with distance squared 14 and orbit size 64, $cf = (1002)$ with distance squared 14 and orbit size 64, $de = (1010)$ with distance squared 12, orbit size 96, and $df = (0011)$ with distance squared 14, orbit size 64. Only the pair de are incompatible, with distance squared too small; as a check, one notices that the dimensions of the subalgebra orbits add up to the dimension of the algebra orbit. The $F_4 \supset B_4$ orbit-orbit generating function can now be written down:

$$\frac{1}{(1-A_1B_2)(1-A_2B_1B_3)}\left(\frac{1}{(1-A_3B_3)(1-A_4B_4)} + \frac{A_3B_1B_4}{(1-A_4B_4)(1-A_3B_1B_4)} + \frac{A_4B_1}{(1-A_3B_1B_4)(1-A_4B_1)}\right). \tag{2.13}$$

The correspondence with the elementary orbits above is

$$A_1B_2 \sim a \qquad A_2B_1B_3 \sim b \qquad A_3B_3 \sim d$$
$$A_4B_4 \sim f \qquad A_3B_1B_4 \sim c \qquad A_4B_1 \sim e.$$

3. Orbits and IR

We show how a set of weights with Weyl symmetry can be represented as a superposition of IR; the weights are then taken to be those of a Weyl group orbit.

Let λ_n be a set of weights of a semisimple algebra and c_n is the multiplicity of the weight λ_n. We suppose the weights have Weyl symmetry. Then the weights can be written as a superposition of the weights of IR:

$$\sum_n \Lambda^{\lambda_n} c_n = \sum_a \chi_a g_a. \tag{3.1}$$

χ_a is the character of the IR (a) and g_a is an integer which we continue to call the multiplicity of (a) even if now it can take negative integer values as well as positive ones; as usual the dummies Λ carry weight components as exponents.

To find g_a we use Weyl's character formula

$$\chi_a = \xi_a / \xi_0 \tag{3.2}$$

where ξ_a is the Weyl characteristic function:

$$\xi_a = \sum_W (-1)^W \Lambda^{W(a+R)}. \tag{3.3}$$

The sum is over Weyl reflections W, $(-1)^W$ is the determinant of the matrix of W, i.e. ± 1 according to whether W is a product of an even or odd number of reflections, and R is half the sum of the positive roots, or the highest weight of the IR with all Dynkin labels unity. ξ_0 is the characteristic function of the scalar IR (with $a = 0$). Then

$$\sum_n c_n \Lambda^{\lambda_n} \xi_0 = \sum_a \xi_a g_a. \tag{3.4}$$

Now ξ_a has just one term, Λ^{a+R}, in the dominant Weyl sector, so g_a is the coefficient of Λ^a in

$$\sum_n c_n \Lambda^{\lambda_n - R} \xi_0. \tag{3.5}$$

We will take λ_n in (3.1) to be the weights of the Weyl orbit $[\lambda]$. Then g_a is the multiplicity of the IR (a) in the expansion of the orbit $[\lambda]$.

The sum in (3.5) can be visualised graphically in the spirit of a Speiser (1962) diagram. Plot the weights of the orbit and reflect each if necessary so that it falls in the dominant sector, reversing the sign with each reflection. The reflection hyperplanes are $\lambda_i = -1$, $i = 1, 2, \ldots, l$ (and their reflections in each other). Ignore a weight that lies on a reflection hyperplane. The resulting weights, interpreted as IR, are the ones which appear in the expansion of $[\lambda]$. Figure 1 shows the diagram for the orbit $[2, 1]$ of G_2. It decomposes into the IR $(21)+(03)+(02)-(04)-(12)-(01)$.

A second graphical method is to plot the dominant weight λ of the orbit and superpose ξ_0 on it so that its dominant weight $(1, 1, \ldots, 1)$ falls on λ. Ignore a weight that lies on a reflection hyperplane, and reflect each of the other weights until it lies in the dominant sector, reversing the sign with each reflection. The reflection planes are the same as in the preceding paragraph. Again the resulting weights, interpreted as IR, are those which appear in the expansion of the orbit $[\lambda]$. However, in the present case, each IR appears with multiplicity equal to the order of the Weyl group divided by the orbit size (number of weights of the orbit) of λ.

Generating function methods can be used to treat the orbit-IR problem. One has only to multiply the orbit-weight generating function $H(A, \Lambda)$, (2.3), by $\xi_0(\Lambda) R^{-1}$ and

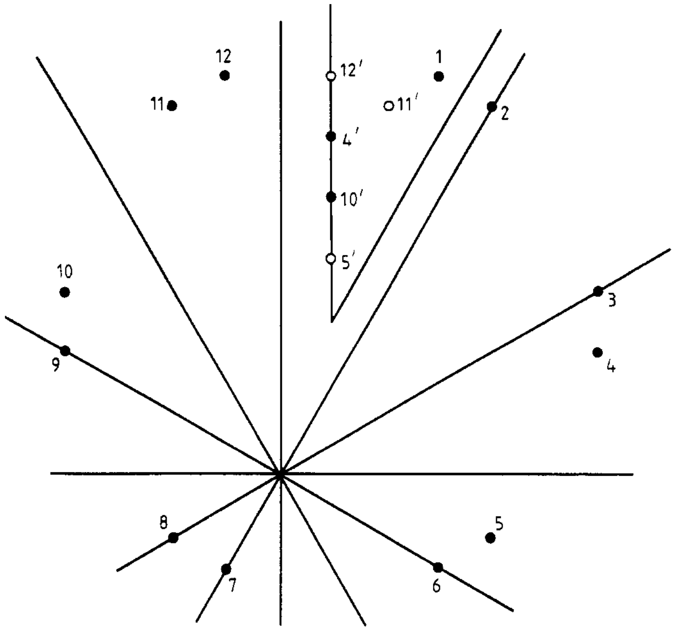

Figure 1. Decomposition into IR of the G_2 orbit [2, 1]. The weights of the orbit are labelled 1, 2, . . . , 12. The reflection lines intersect at (−1, −1). The weights 4, 5, 10, 11, 12 are reflected into the dominant sector to the corresponding primed weights where they, and 1, are interpreted as IR. A closed circle means plus (even number of reflections), an open circle minus (odd number of reflections).

retain non-negative powers of Λ in the result. Then one gets the orbit-IR generating function $J(A, \Lambda)$ whose power expansion

$$J(A, \Lambda) = H(A, \Lambda)\xi_0(\Lambda)R^{-1}|_{\Lambda^+} = \sum_{a,\lambda} A^a\Lambda^\lambda c_{a,\lambda} \tag{3.6}$$

gives the multiplicity $c_{a,\lambda}$ of the IR (λ) in the expansion of the orbit $[a]$.

For example, the SU(3) orbit-IR generating function turns out to be

$$J(A, \Lambda) = \frac{1}{(1-A_1\Lambda_1)(1-A_2\Lambda_2)}(1 - A_2^2\Lambda_1 + A_2^3 - A_1^2A_2^2 + A_1^3 - A_1^2\Lambda_2) - \frac{A_1A_2}{1-A_2\Lambda_2} - \frac{A_1A_2}{1-A_1\Lambda_1} \tag{3.7}$$

and the SO(5) orbit-IR generating function is

$$\frac{1}{(1-A_1\Lambda_1)(1-A_2\Lambda_2)}(1 - A_2^2\Lambda_1 + A_2^4\Lambda_1 - A_2^4 + A_1^2A_2^2 - A_1^3 + A_1^3\Lambda_2^2 - A_1^2\Lambda_2^2) + \frac{A_1A_2^2 - A_1}{1-A_2\Lambda_2} + \frac{A_1A_2^2 + A_1^2A_2\Lambda_2 - A_1A_2^2\Lambda_1 - A_1A_2\Lambda_2}{1-A_1\Lambda_1} - A_2^2. \tag{3.8}$$

The orbit $[a]$ contains only IR of equal or lower level; hence the matrix from the orbit basis to the IR basis is triangular. It is therefore easy to invert it and obtain the orbit content of an IR. Thus the problem of weight multiplicities of an IR, usually solved by Freudenthal's recursive technique, or else by use of Weyl's or Demazure's character formula, is rather easily solved, working upwards by level within each congruence class. A similar approach has been applied to the Kac-Moody algebra $A_1^{(1)}$ by Kass (1986).

Sometimes the orbit-IR generating function can be inverted to obtain the IR-orbit generating function whose power expansion gives the multiplicities of orbits in IR. If $J(A, \Lambda)$ and $K(\Lambda, A)$ are the orbit-IR and IR-orbit generating functions, they satisfy the equation

$$J(A, \Lambda)K(\Lambda^{-1}, A')|_{\Lambda^0} = \prod_i^l (1 - A_iA_i')^{-1}. \tag{3.9}$$

We illustrate the procedure with SU(2), whose orbit-IR generating function is $(1 - A^2) \times (1 - A\Lambda)^{-1}$. Then (3.9) for $K(\Lambda, A)$ becomes

$$(1 - A^2)K(A, A') = (1 - AA')^{-1}$$

from which we find

$$K(\Lambda, A) = [(1 - \Lambda^2)(1 - \Lambda A)]^{-1}.$$

IR-orbit generating functions for SU(3) and SO(5) are given by Michel *et al* (1988).

4. Concluding remarks

The idea that the Weyl group is a useful tool in applications of representation theory of semisimple Lie algebras is not new, but until recently it might have been difficult to make a convincing case. That was due partly to the fact that, in most applications, only representations of relatively small dimension were of importance and that the ranks of relevant semisimple Lie algebras were modest so that most problems could be solved without much sophistication. By now, of course, the situation is quite different in that respect.

A truly systematic exploitation of the Weyl group in applications became possible when it became possible in practice to calculate the multiplicity of any W-orbit in the weight system of any representation, at least for the representations which may be of interest in the forseeable future. Indeed, until Moody and Patera (1982), the practical limit of the existing recursive methods were representations of several thousands for ranks, say 8, even for the largest mainframe computers.

In this paper the task of computing the branching rules makes use of the fact that the multiplicities can be calculated or looked up (Bremner *et al* 1985) separately whenever needed. It further splits the problem into two smaller ones: computation of branching rules for the W-orbits of a given simple Lie algebra to W-orbits of a maximal subalgebra (subjoined algebras are included as well), and subsequent collecting of the appropriate numbers of appropriate W-orbits into complete weight systems of representations of the subalgebra according to the known multiplicities. We give an explicit prescription for writing a Weyl orbit as a superposition of irreducible representations and, by inversion of a triangular matrix, writing an irreducible representation as a superposition of Weyl group orbits.

Exploitation of generating functions for both steps of our computations adds a further efficiency to the procedure, especially for the calculation of algebra W-orbit to subalgebra W-orbit branching rules. For the equal-rank case we give explicit easy-to-implement instructions for the construction of the complete orbit-orbit generating function. Even when the complete orbit-orbit generating function is too complicated to write down, it may be possible to convey the same information by listing the members of the integrity basis (elementary orbits) and their syzygies (incompatibilities). In still more complicated cases one is typically interested in branching rules for only highly degenerate representations of an algebra; then only a small subset of the integrity basis and the corresponding syzygies are required.

Acknowledgment

We thank Dr C J Cummins for helpful discussions.

References

Bremner M R, Moody R V and Patera J 1985 *Tables of Dominant Weight Multiplicities for Representations of Simple Lie Algebras* (New York: Dekker)

Couture M and Sharp R T 1989 *J. Phys. A: Math. Gen.* **22** 1525-42

Cummins C, Patera J, Rodriguez M and Zaoui M 1989 *Branching Rules for Representations of Simple Lie Algebras* to appear

Kass S N 1986 *Infinite Dimensional Lie Algebras and Their Applications* ed S N Kass (Singapore: World Scientific)

McKay W and Patera J 1981 *Tables of dimensions, Indices, and Branching Rules for Representations of Simple Lie Algebras* (New York: Dekker)

McKay W, Patera J and Sankoff D 1977 *Computers in Nonassociative Rings and Algebras* ed J Beck and B Kolman (New York: Academic) p 235-77

Michel L, Patera J and Sharp R T 1988 *J. Math. Phys.* **29** 777-96

Moody R V and Patera J 1982 *Bull. Am. Math. Soc.* **7** 237-42

Navon A and Patera J 1967 *J. Math. Phys.* **8** 489-93

Patera J, Sharp R T and Slansky R 1980 *J. Math. Phys.* **21** 2335-41

Speiser D 1962 *Lectures of the Istanbul Summer School of Theoretical Physics* vol 1 (New York: Gordon and Breach)

Weyl H 1925 *Math. Z.* **23** 271