TOPICS IN ALGEBRA
BANACH CENTER PUBLICATIONS, VOLUME 26, PART 2
PWN POLISH SCIENTIFIC PUBLISHERS
WARSAW 1990

THE LITTLEWOOD-RICHARDSON RULE — THE CORNERSTONE FOR COMPUTING GROUP PROPERTIES

B. G. WYBOURNE

Physics Department, University of Canterbury, Christchurch, New Zealand

The role of the Littlewood–Richardson rule in the computation of the properties of Lie groups is outlined. It is shown that the modern personal computer can lead to an efficient interactive evaluation of many group properties.

Introduction

The applications of group theory and combinatorics in physics continue its unabated growth. The diversity of applications is little short of amazing, we mention but a partial list.

(a) Atomic Physics. The group chain

$$U_2{}^{14} \supset SO_{28} \supset SU_2 \times [Sp_{14} \supset SU_2 \times [SO_7 \supset G_2 \supset SO_3]]$$

has found extensive application in the analysis of rare earth spectroscopy [1–3]. The conformal group SO(4, 2) has been used as the dynamical group of the hydrogen atom [3, 4] while the group SO(2, 1) $\sim$ SU(1, 1) has been used in the analysis of Coulomb radial integrals [5, 6].

(b) Quantum Optics and Interferometry. The groups SU(1, 1) and SU(2) enter in the description of multiple interferometers [7] while the noncompact group Sp(4, R) enters in quantum optics [8].

(c) Jahn–Teller Effect. Under appropriate interaction parameters some Jahn–Teller Hamiltonians exhibit SO_5 symmetry [9] while in other cases supersymmetry has been exploited as a tool for making calculations [10].

(d) Nuclear Models. The interacting boson model (IBM) of nuclei has introduced the group chain $U_6 \supset SO_6 \supset SU_3 \supset SO_3$ [11] and many variants while supersymmetry studies have centred around the supergroups U(m/n) and OSp(m/n) [12, 13]. The noncompact group Sp$(2n, R)$ and its maximal compact subgroup is important in certain nuclear models [14].

This paper is in final form and no version of it will be submitted for publication elsewhere.

(e) Particle Physics. The particle physicists tend to be prodiguous users of group structures and their extensions. Thus in the heterotic string we find groups $E_8 \times E_8$ and SO_{32} feature prominently as well as their various subgroups [15]. Particle physicists are not adverse to making daring excursions into such topics as the Monster Group, infinite dimensional Lie algebras and Kac-Moody algebras.

The above few examples should give encouragement to mathematicians – their work is greatly valued by the physicists!

At the most mundane level physicists need to be able to readily obtain information on (1) the properties of irreps, such as dimensions and Casimir eigenvalues, (2) the resolution of Kronecker products and (3) the decomposition multiplicities for group–subgroup combinations. Over the years we have tried to formulate algorithms to make possible the interactive evaluation of group properties on personal computers.

Labelling representations

The irreps of the compact semisimple Lie groups may all be uniquely labelled by ordered partitions of integers or half-integers. Thus for the classical Lie groups we have [16]:

SU_n	$\{\lambda\}$ $p \leqslant n-1$		
SO_{2k+1}	$[\lambda]$ $p \leqslant k$	$[\Delta; \lambda]$ $p \leqslant k$	
SO_{2k}	$[\lambda]$ $p \leqslant k$	$[\lambda]_{\pm}$ $p = k$	$[\Delta; \lambda]_{\pm}$ $p \leqslant k$
Sp_{2k}	$\langle\lambda\rangle$ $p \leqslant k$		

where p is the number of parts of (λ) and the $\pm$ distinguish pairs of irreps whose highest weights are distinguished by their kth part being + or −.

The irreps of the exceptional Lie groups may be given a natural labelling in terms of the corresponding labels attributed to a chosen maximal classical Lie subgroup [16-19]. Thus $G_2 \supset SU_3$, $F_4 \supset SO_9$, $E_6 \supset SU_2 \times SU_6$, $E_7 \supset SU_8$ and $E_8 \supset SU_9$ or $E_8 \supset SO_{16}$. In each case we are led to a unique labelling scheme based on constrained partitions, as tabulated in [16]. A one-to-one correspondence between partition labels and the corresponding Dynkin labels exists [16]. Similar labelling schemes may be developed for the ordinary and spin irreps of the symmetric S_n and alternating A_n finite groups [20-24].

Finally we note that the infinite dimensional irreps of the positive discrete series D^+ and the harmonic series unitary irrep of the noncompact groups $U(p, q)$, $Sp(2n, R)$ and $SO^*(2n)$ may likewise be labelled in terms of partitions of integers based on the corresponding labelling adopted for their maximal compact subgroup [25, 26]. Thus for $Sp(2n, R)$ the discrete series irrep becomes labelled $\langle\{\lambda\}\rangle$ and those of the harmonic series by $\langle \frac{1}{2} k (\lambda)\rangle$ where k is an integer.

S-Function series

Professor King has outlined the role of special S-function series in giving a concise one symbol universal characterisation of the Kronecker products and branching rules for the classical Lie groups. Thus the Kronecker products of the tensor irreps of Sp_{2k} may be concisely written as

$$\langle\lambda\rangle . \langle\mu\rangle = \sum_{\zeta}\langle\lambda/\zeta . \mu/\zeta\rangle \tag{1}$$

which involve simply the evaluation of the S-function skews and outer S-function products using the Littlewood–Richardson rule with the summation being over all compatible S-functions ζ. Similar results may be readily developed for other products [16], [27].

Branching rules may be developed in a similar fashion [16], [28], [29]. Thus the $U_n \downarrow O_n$ decomposition for the covariant irreps $\{\lambda\}$ of U_n may be concisely written as

$$\{\lambda\} \downarrow [\lambda/D] \tag{2}$$

where D is the S-function series characterised by partitions whose parts are all even.

The use of particular S-function series leads to a process of symbolic manipulation of S-functions that is well adapted to computer implementation. Indeed most of the relevant series can be generated by a single piece of PASCAL code. We note that the branching rules for the aforementioned noncompact groups can also be described concisely in terms of S-function series. Thus for $Sp(2n, R) \downarrow U(n)$ we have for the positive discrete irreps [26], [31]

$$\langle\{\lambda\}\rangle \downarrow \{\lambda\}.D \tag{3}$$

Note that in (2) the D series appears as a skew whereas in (3) it appears as an outer product. Equations (2) and (3) show clearly the significance of the Littlewood–Richardson rule in computing group properties.

Modification rules

The evaluation of Kronecker products and branching rules may lead to the appearance of non-standard characters which must be modified to produce either a null result or a signed standard character. A complete set of modification rules for the exceptional and classical Lie groups is given in [16].

Modification rules are symptomatic of a measure of overcounting. S-function procedures produce universal results and the modification rules only arise when specialisation is made to specific cases involving low rank groups. In practice modification rules are extremely simple to programme and a string of standard and non-standard characters can be reduced to a string of standard characters with negligible time or memory penalty.

Kronecker products for exceptional groups

King [19] has shown that if a group G has a maximal classical Lie subgroup H and if the decomposition

$$\mu_G \to \sum_{\nu_H} g_{\mu_G}^{\nu_H} \nu_H \tag{4}$$

is known then the Kronecker product $(\lambda \times \mu)_G$ where λ is an arbitrary irrep of G and can be evaluated by what amounts to essentially doing the Kronecker products $(\lambda \times \nu)_H$ in the subgroup H standardizing the result in H and then modifying in G. Explicit details have been given in [16]. Again the Littlewood–Richardson rule plays the key role.

By way of example we note that the fundamental irrep (21^7) of E_8 is of dimension 248 and the Kronecker seventh power of (21^7), which is of course of degree $(248)^7$ can be evaluated in ~ 4 minutes using an XT IBM clone operating at 8 MHz. In this case the sum of the multiplicities exceeds 260,000. The key to such speed is a highly efficient implementation of the Littlewood–Richardson rule coupled with an efficient opportunistic sort strategy.

Computing S-function operations

Our implementation of the Littlewood–Richardson rule is such that an XT clone at 8 MHz will determine as a sorted list the 930 terms in $\{4321\}.\{4321\}$ in 4.7 seconds. On a SUN 360/4 workstation the result appears almost instantly. It is this high speed that permits the computer to be used as an interactive device for computing group properties.

Inner products are computer making use of the reduced notation [23], [32] for S_n where the partition $(\lambda) \equiv (n-m, \mu)$ with $\mu = \lambda_2 \lambda_3 \ldots \lambda_p$ of weight m is written in reduced notation as $\langle \mu \rangle$. Explicit details are given in [32].

S-function plethysms $\{\lambda\} \otimes \{\mu\}$ are evaluated using an extension of a result due to Butler and King [33]. The computation of S-function plethysms is a complex problem and a truly efficient method remains to be discovered. It seems almost inevitable that each method proposed necessarily involves overcounting which becomes excessive as the size of the problem increases.

Computing with Schur 4.1

Over the years we have been trying to refine the algorithms for calculating group properties and the Littlewood–Richardson rule forms the cornerstone of the package SCHUR 4.1 [34]. This package consists of $\sim$ 174 K of compiled PASCAL code and runs on any IBM compatible personal computer (XT, AT or 286) with 512 K or greater usable RAM. It is designed to carry out the following operations.

(a) Kronecker products for all compact Lie groups including direct product groups and the ordinary irreps of the symmetric group. Products may be evaluated for lists or strings of irreps.

(b) Branching rules such as

$$U_n \to O_n, \quad U_{2n} \to Sp_{2n}, \quad SO_{m+n} \to SO_m {*} SO_n, \quad SU_{m+n} \to U_1 {*} SU_m {*} SU_n$$

$$SO_n \to U_1 {*} SU_k, \quad U_{mn} \to U_m {*} U_n, \quad U_n \to SO_3, \quad SU_{m/n} \to U_1 {*} SU_m {*} SU_n,$$

$$OSp_{m/n} \to O_m {*} Sp_n, \quad U_{m+n/p+q} \to U_1 {*} SU_{m/p} {*} SU_{n/q},$$

$$U_{mn+pq/mp+nq} \to U_{m/p} {*} U_{n/q},$$

$$G_2 \to SU_3, \quad F_4 \to SO_9, \quad E_6 \to SU_2 {*} SU_6, \quad E_6 \to U_1 {*} SO_{10}, \quad E_7 \to SU_8,$$

$$E_8 \to SU_9, \quad E_8 \to SU_2 {*} E_7, \quad E_8 \to SO_{16}, \quad SO_7 \to G_2, \quad G_2 \to SO_3.$$

Other branching rules may be built up using SCHUR 4.1 interactively.

(c) The Schur function operations Outer, Inner, Skew and Plethysm are available along with commands generating the infinite Schur function series up to a user defined cutoff. The Young tableaux or frames of these operations may be displayed directly on the screen.

(d) Modification procedures for standardising non-standard representations are automatically invoked as required.

(e) Properties such as dimensions, Casimir eigenvalues and conversion between partition labels and the Dynkin notation are available.

(f) Sequences of instructions may be set as functions allowing the user to implement user defined rules.

(g) Helpfiles can be brought to the screen at any time giving a complete description of the over 100 commands available.

Examples

Consider the S-function content up to terms of weight $\leqslant 8$ of the generating function

$$\prod_i (1-x_i) \prod_{i \leqslant j} (1-x_i x_j). \tag{5}$$

We first recognise that (5) is the product of two simpler series [31] L and C. The terms of the L and C series up to weight 8 are evaluated by the commands

trwt 8, L and **trwt** 8, C

We then need to determine the outer S-function product and then bring to the screen terms of weight $\leqslant 8$ of the resultant to give:

SFN >
weight8, outer, trwt8, L, trwt8, C

$-\{521\}+\{511\}-\{44\}+\{43\}-\{422\}+\{42\}-\{41\}-\{3311\}+\{331\}-\{33\}$
$+\{32111\}-\{3211\}+\{321\}-\{32\}+\{3\}-\{2111111\}+\{211111\}-\{21111\}$
$+\{2111\}-\{211\}+\{21\}+\{11111111\}-\{1111111\}+\{111111\}-\{11111\}$
$+\{1111\}-\{111\}+\{11\}-\{1\}+\{0\}$

As a second example consider the branching rule for $O_N \downarrow S_N$ where in reduced notation [23]

$$[1] \downarrow \langle 0 \rangle + \langle 1 \rangle \tag{6}$$

Following Luan Dehuai and Wybourne [23] we may write the decomposition rule for an arbitrary tensor irrep of O_N as

$$\langle 1 \rangle \otimes \{\lambda/MC\} = \langle 1 \rangle \otimes \{\lambda/G\} \tag{7}$$

where we note that $MC = G$. The reduced notation plethysm is evaluated in SCHUR by the command **plinner.** Thus for the decomposition of the irrep [321] of O_N the command sequence

plinner skew 321, G

would yield the irreps of S_N in reduced notation. To obtain the results for a specific value of N it is necessary to prefix each partition in the reduced notation with a part sufficient to make the partition up to weight N and then to standardise any non-standard irreps of S_N. This is accomplished by the command sequence, for say $O_9 \downarrow S_9$,

mkweight 9, plinner, skew 321, G

In practice it is more generally useful to make a user defined function such as:

group O_9
enter $sv1$
dim[convert $sv1$]
group S_9
[convert mkweight9, plinner, skew, $sv1$, G]
dim last
last
stop

The first statement sets the group as O(9) and then the succeeding statement asks the user to enter a partition ($sv1$). The third statement converts ($sv1$) into a standard irrep of O_9 and computes its dimension. The set group is now changed to $S(9)$ and the sequence of S-function operators computed and then converted into standard labelled irreps of $S(9)$. The dimension of the result is computed so that we can check that it agrees with that calculated for $O(9)$. Finally, the $S(9)$ content of the decomposition is displayed.

The function is invoked by a single command *fn*1 to produce the output given below:

*fn*1
Group is O(9)
enter *sv*1
321
Dimension = 9009
Group is $S(9)$
Dimension = 9009

$$2\{81\}+6\{72\}+6\{711\}+6\{63\}+15\{621\}+6\{6111\}+2\{54\}+9\{531\}$$
$$+7\{522\}+9\{5211\}+2\{51111\}+\{441\}+3\{432\}+3\{4311\}+3\{4221\}$$
$$+\{42111\}+\{3321\}$$

The above two examples give some indication of how the computer can assist in evaluating the properties of Lie groups and of the symmetric groups [34].

Concluding remarks

I hope we have demonstrated the manner in which the Littlewood–Richardson rule can form the cornerstone for computing many properties of Lie groups. I must take the opportunity of thanking the organisers of this Seminar at the Stefan Banach International Mathematical Center for a stimulating week of great interest and true international spirit.

References

[1] B. G. Wybourne, *Spectroscopic Properties of Rare Earths*, J. Wiley & Sons, New York 1966.
[2] —, *Symmetry Principles in Atomic Spectroscopy*, ibid. 1970.
[3] —, *Classical Groups for Physicists*, ibid. 1974.
[4] A. O. Barut, *Dynamical Groups and Generalized Symmetries in Quantum Theory*, University of Canterbury Press, Christchurch, New Zealand 1972.
[5] L. L. Armstrong Jr., Phys. Rev. A(3), 1546 (1970).
[6] M. J. Cunningham, J. Math. Phys. 13, 33 (1972).
[7] B. Yurke, S. L. McCall and J. R. Skauder, Phys. Rev. A(33), 4033 (1986).
[8] G. J. Milburn, J. Phys. A 17, 737 (1984).
[9] B. R. Judd, Canad. J. Phys. 52, 999 (1974).
[10] P. D. Jarvis and G. E. Stedman, J. Phys. A 17, 757 (1984).
[11] A. Arima and Iachello, *The Interacting Boson Model I*, Cambridge University Press, Cambridge 1988.
[12] A. B. Balantekin and I. Bars, J. Math. Phys. 22, 1810 (1981).
[13] P. D. Jarvis, M. Yang and B. G. Wybourne, ibid. 28, 1192 (1987).
[14] D. J. Rowe, Rep. Progr. Phys. 48, 1419 (1985).

[15] M. B. Green, J. Schwartz and E. Witten, *Superstring Theory*, Vols. 1 & 2. Cambridge University Press, Cambridge, 1987.
[16] G. R. E. Black, R. C. King and B. G. Wybourne, J. Phys. A 16, 1555 (1983).
[17] B. G. Wybourne and M. J. Bowick, Austral. J. Phys. 30, 259 (1977).
[18] R. C. King and A. H. A. Al-Qubanchi, J. Phys. A 14, 15 (1981); 14, 51 (1981).
[19] R. C. King, ibid. 14, 77 (1981).
[20] G. Frobenius, S. B. Preuss. Akad. Wiss. Lit. Mainz. Abh. Math.-Naturwiss. Kl. p. 303 (1901).
[21] I. Schur, J. Reine Angew. Math. 139, 155 (1911).
[22] A. O. Morris, Proc. London Math. Soc. (3) 12, 55 (1962).
[23] Luan Dehuai and B. G. Wybourne, J. Phys. A 14, 327 (1981).
[24] –, –, ibid. 14, 1835 (1981).
[25] D. J. Rowe, B. G. Wybourne and P. H. Butler, ibid. 18, 939 (1985).
[26] R. C. King and B. G. Wybourne, ibid. 18, 3113 (1985).
[27] G. R. E. Black and B. G. Wybourne, ibid. 16, 2405 (1983).
[28] R. C. King, ibid. 8, 429 (1975).
[29] M. Yang and B. G. Wybourne, ibid. 19, 2003 (1986).
[30] R. C. King and B. G. Wybourne, ibid. 15, 1137 (1982).
[31] M. Yang and B. G. Wybourne, ibid. 19, 3513 (1986).
[32] P. H. Butler and R. C. King, J. Math. Phys. 14, 1176 (1973).
[33] –, –, ibid. 14, 741 (1973).
[34] SCHUR 4.1 is distributed by SCHUR Software Associates, School Road, Yaldhurst, No. 6 R. D., Christchurch, New Zealand.