Euromath Bulletin
Vol. 2, No. 1, 1996

SCHUR, An Interactive Program for Calculating Properties of Lie Groups and Symmetric Functions

B. G. Wybourne
Instytut Fizyki, Uniwersytet Mikolaja Kopernika
ul. Grudziadzka 5/7, 87−100 Torun
Poland
bgw@phys.uni.torun.pl

Abstract

SCHUR is an interactive program for calculating properties of Lie groups and symmetric functions of relevance to a wide range of problems in physics, chemistry and mathematics. The basic operating principles of SCHUR are outlined along with examples of computations using SCHUR.

1 Introduction

SCHUR is an interactive C package for calculating properties of representations of Lie groups and yet it is much more. SCHUR has been designed to answer questi of relevance to a wide range of problems of special interest to physicists, chemists and mathematicians. These include not only problems directly relating to Lie groups but also many properties of symmetric functions. The development of SCHUR has occurred over many years and has been driven by response to specific research problems and in the need to make available to students a tool for learning about Lie groups by student creation of examples of practical calculations.

As the name of the package suggests much of the structure of SCHUR derives from the pioneering work of Issai Schur on symmetric functions and the representatio of matrix groups. Many persons find in their work they need specific knowledge relating to some aspect of Lie groups or symmetric functions and yet do not wish to be encumbered with complex algorithms or a need to be familiar with particular programming languages. The objective of SCHUR is to supply results with the complexity of the algorithms fortunately hidden from view. The user is able to obtain specific results and use these results to obtain further results in a fully interactive manner, effectively using SCHUR as a scratch pad. In using SCHUR interactively the user is able to exploit the many commands built into SCHUR and to define his or her own command structures.

The basic object in SCHUR is the partition of an integer into integers. The partitions index the Schur functions (or $S-$functions) which are a class of symmetric functions [15], [16]. The representations of the compact Lie groups [10], [1] and certain of those of the non−compact groups [11] (e.g. $Sp(2n,R)$)

may be uniquely labelled by certain constrained partitions of integers, as for tensor representations, or as half−integers for spinor representations. Such a notatio is familiar to physicists in the use of Young tableaux in describing tensors. Thus SCHUR is significantly different from programs involving weight space constructions and Dynkin diagrams though SCHUR will give translations of partition labels into Dynkin labels and vice versa.

The package SCHUR includes a manual of over 220 pages describing, usually with examples, every command as well as explanatory chapters on the mathematics behind SCHUR, a detailed explanation of notations used together with two tutorial chapters on using SCHUR. A number of examples drawn from physics, chemistry and mathematics are included. On−line help is available. Descriptions of every command, with an example, can be readily brought to screen.

2 The tasks of SCHUR

Among the many tasks amenable to attack by SCHUR are the following:

1. The calculation of Kronecker products for all compact Lie groups and for the ordinary and spin represention of the symmetric group. Kronecker products for the unitary discrete representations of the non−compact group $Sp(2n,R)$ up to a user defined weight limit. Not only individual representations but also lists of representations. List handling is a general feature of SCHUR.
2. The calculation of branching rules with the ability to successively branch through a chain of nested groups including certain non−compact groups involving subgroups of $Sp(2n,R)$ and the metaplectic groups $Mp(2n)$.
3. The calculation of properties of irreducible representations such as dimensions, second−order Casimir and Dykin invariants, the trace of the$n-$th order Casimir invariants and the conversion between partition and Dynkin labelling of representations.
4. The handling of direct products of several groups.
5. The computation of a wide range of properties related to Schur function operations such as the Littlewood−Richardson rule, inner products, skew products and plethysms as well as the inclusion of commands for generating the terms in infinite series of Schur functions up to a user defined cutoff.
6. The computation of properties of the symmetric $Q-$ functions with respect to operations such as the analogous Littlewood−Richardson rule, skew and inner products.
7. The standardisation of non−standard representations of groups by the use of modification procedures. Generation of signed−sequences of non−standard labelled representations that modify to within a sign of a chosen standard representations.
8. Transformations between various types of symmetric functions.

3 Problems involving use of SCHUR

The development of SCHUR has been research driven and as a result has involved a large range of applications using various features of SCHUR [11], [27].

In [3] a study of spectrum−generating functions for strings and superstrings required SCHUR to compute dimensions of representations of groups such as $SO(16)$, $E_8, SO(32)$ etc as well as the terms in certain generating functions. The study of spin representations of symmetric groups [18], $S(n)$, required SCHUR to form inner, skew and outer products of $Q-$ functions and to compute Kronecker products involving both ordinary and spin representations of $S(n)$.

A study of the extension of the interacting boson model to the exceptional groups [17] required SCHUR to be able to compute branching rules for group$-$subgroup combinations such as arise in the group chains

$$U(28) \to SU(7) \to SO(7) \to (2) \to SU(3) \to SO(3)$$

and

$$U(28) \to SU(27) \to E(6) \to G(2) \to SU(3) \to SO(3)$$

as well as computing Kronecker products for the various groups in the chains.

An analysis of the normal forms for tensor polynomials involving the Riemann tensor [4], [26] drew on SCHUR's ability to compute products, skews and plethysms of $S-$functions. The enumeration and counting of scalars required use of the modification rules for the full orthogonal groups. Here SCHUR's handling of lists of $S-$functions and representations of groups proved to be of particular value.

The problems associated with handling the infinite dimensional discrete unitary representations of the non$-$compact Lie groups such as $Sp(2n,R)$ and the metaplectic groups $Mp(2n)$ were considered in [11] and [5], [22], [23]. These involved the full use of SCHUR to compute $O(n) \to S(n)$ branching rules, inner and outer plethysms of $S-$functions, nested sets of commands, signed sequences of representations etc. SCHUR was also used to calculate plethysms of the basic representations of $Sp(2n,R)$ [7], [14] leading to hitherto unknown identities concerning infinite series of $S-$functions and new closed results for certain $Sp(2n,R)$ plethysms. This illustrates an important feature of SCHUR in establishing conjectures that have then inspired efforts at formal proof construction.

The three$-$fold automorphisms of the group $SO(8)$ were exploited in [25] using SCHUR to apply the the automorphism operation to lists of representations of the group and to carry out the branching decompositions for the different $SO(8) \supset SO(7) \supset G(2)$ embeddings.

The results reported in [19], [20] were stimulated by some initial studies using SCHUR which led to a number of conjectures which in turn encouraged the development of formal proofs. Conversely a conjecture [5] concerning the number of admissible Young tableaux that characterise the $S-$functions arising in the expansion of even powers of the Vandermonde determinant for N variables was disproved as a result of the discovery of a counter example using SCHUR [20]. While the conjecture had been shown to be correct for up to $N=5$ SCHUR showed that the conjecture continued to hold for $N \leq 7$ but failed for $N \geq 8$.

The study of the squares of self$-$complementary $S-$functions [27] led to a conjecture whose formal proof then lead to the determination of the number of times the adjoint representation of $SU(n)$ occurs in the Kronecker square of an arbitrary self$-$contragredient representation and the determination of the number of times the adjoint appears in the symmetric and antisymmetric parts of the square. Similar studies have recently been completed for the simple Lie groups [14].

4 Input Output issues

The input of SCHUR consists of commands and integers while the output consists of written statements and integers enclosed in a variety of brackets. SCHUR is case independent so that upper and lower case letters may be freely used in all input statements. Spaces have no significance and may be inserted anywhere for clarity. Spaces or commas are mandatory between entries of the same type as for example between two commands or two numbers.

SCHUR knows only of the existence of integers and hence all input and output of numbers is in the form of integers. Most frequently input into SCHUR involves lists of numbers. In designing SCHUR considerable attention has been given to ease of input of partitions of integers and to maximise flexibility. In reading in partitions SCHUR is unconcerned as to the order of the integers making up the partitions. In normal usage SCHUR assumes that most partitions inputted will have parts ≤ 9 in which cases the integers may be simply entered together with no spaces required. Thus the partition "9 4 3 2 1" could be simply entered as 94321. If a part is ≥ 10 it is entered prefixed with an exclamation sign and the part terminated by a space or a non−digit. The usual power notation is used for partitions having repeated parts. A partition having a negative part is entered by preceding the part by a tilde. Thus the partition "12 10 9−8754443322" may entered as `!12!10 9~8754^33^22^2` or even as `!12 !10 9 ~8 7 5 4^3 33^2 2^2`. In each case the partition is outputted by SCHUR as

$$\{12\,10\,9-874\hat{}3\,3\hat{}2\,2\hat{}2\,\}$$

In the case of group representations labelled by partitions SCHUR will automatically encase the partitions with brackets $\{,\}$ if the group is a unitary, or special unitary, or symmetric group, [,] for orthogonal groups, $<,>$ for symplectic groups or (,) for the exceptional Lie groups. For products of Lie groups the appropriate brackets are supplied as appropriate to each of the groups in the product.

SCHUR can produce lists in response to the input of commands. Commands can also act on lists to produce new lists. A user may nest commands inside commands with only the final output being displayed with all the intermediate output (which may be massive) being suppressed.

5 Help files

SCHUR contains 200 help files which may be brought to screen at any time by entering ?filename. Thus issuing the command ?help produces a list of all the help files. Issuing the command ?o produces the following output

```
O_sfnProduct
    Format:- O EXPR1,EXPR2
     Modes:-SFN
Definition:-Calculates  the  S-function  outer  product  using  the
            Littlewood-Richardson rule.
Example:-
         SFN>
       ->o 21,21
         {42} + {41^2} + {3^2} + 2{321} + {31^3} + {2^3} + {2^2 1^2}
         SFN>
```

6 Examples of using SCHUR

Let us now illustrate a few of the features of SCHUR in action. In each case we have marked input with an arrow $->$. Issuing the command "schur" we obtain, apart from installation specifics,

```
(If you wish to EXIT, enter END)
(If you wish to obtain HELP, enter ?help)
DPrep Mode (with function)
DP>
```

To calculate the outer product of two lists of Schur functions we switch to the SFN mode as instructed above to give

```
->sfn
```

```
Schur Function Mode
SFN>
->o21+1,21+11
       {42} + {41^2 } + {3^2 } + 2{321} + {32} + {31^3 } + {31^2 }
       + {31} + {2^3 } + {2^2 1^2 } + {2^2 1} + {2^2 } + {21^3 }
       + {21^2 } + {21} + {1^3 }
SFN>
->yhook last

                                        2
     5  4  2  1    6  3  2  1    4  3  2    5  3  1    4  3  1

     2  1          2             3  2  1    3  1       2  1

                   1                        1

     6  2  1    5  2  1    4  2  1    4  3    5  2    4  2    3  2

     3          2          1          3  2    4  1    3  1    2  1

     2          1                     2  1    2       1

     1                                1

     5  1    4  1    3  1    3

     3       2       1       2

     2       1               1

     1
 SFN>
```

The instruction “yhook last” has taken the partitions obtained in the previous output “last” and computed the hooklengths for every cell of the corresponding Young frame. We could have arrived at the same final output by inputting the command sequence “yhook o21+1,21+11”. Issuing the command sequence “yhook wt5 o21+1,21+11” results in just the hooklengths for the partitions of weight ≤ 5 being output while “yhook wt−5 o21+1,21+11” would give the output for partitions having exactly 5 parts.

SCHUR can calculate dimensions and other properties of irreducible representations of groups as illustrated below for the staircase representation of the symmetric group $S(105)$

```
->rep
REP mode
REP>
->gr s105
Group is S(105)
REP> ->!14!13!12!11!10 987654321
           {14 13 12 11 10 987654321}
REP>
->dim last
dimension=5137825685807319573670197678030853203966327760999759
         18380865685412418054992691200
REP>
```

SCHUR does not construct representation matrices. While the complete construction for every element of the representation matrices for any irreducible representation of any $S(n)$ group is known to calculate just the diagonal elements of the matrix for the above case for an element of $S(105)$ would take a supercomputer many trillions of times longer than the age of the universe.

The group $SO(8)$ has tensor representations that may be labelled by partitions of not more than 4 parts and in the case of 4 part partitions pairs of inequivalent representations occur that are distinguished by an additional label $\pm$. The spin representations are designated by a spin index s and a partition into at most 4 parts. Spin representations occur in inequivalent pairs distinguished again by a label $\pm$. Some features of SCHUR for the group $SO(8)$ are now shown by the results given below

```
REP>
->gr so8
Group is SO(8)
REP>
->s321+
      [s;321]+
REP>
->prop last
<dynkin label>(1121)
dimension=12936   48*2nd-casimir=220   2nd-dynkin=12705
REP>
->prop11
<dynkin label>(0100)
dimension=28   48*2nd-casimir=48   2nd-dynkin=6
REP>
->prod 11,s321+
   [s;431]+ + [s;42^2]+ + [s;421^2]+ + [s;421]- + [s;42]+
 + [s;41^2]+ + [s;3^22]+ + [s;3^21^2]+ + [s;3^21]- + [s;3^2]+
 + [s;32^21]+ + [s;32^2]- + 4[s;321]+ +  [s;32]- + [s;31^3]+
 + [s;31^2]- + [s;31]+ + [s;2^3]+ + [s;2^21^2]+ + [s;2^21]-
 + [s;2^2]+ + [s;21^2]+
 REP>
->prop last
<dynkin label>(1221) + (2032)+ (2130) + (2112) + (2210) + (3021)
+ (0132) + (0230) + (0212) + (0310) + (1041) + (1023) + 4(1121)
+ (1201) + (2030) + (2012) + (2110) + (0032) + (0130) + (0112)
+ (0210) + (1021)
dimension=362208   2nd-dynkin=433356
REP>
```

In the above example, we first illustrate the calculation of some of the properties of irreducible representations of $SO(8)$ and then compute the Kronecker product of two irreducible representations and then compute the corresponding dynkin labels, dimension and second order dynkin index of the result.

The group $SO(8)$ has a three−fold automorphism and continuing with the above output we have the automorphisms of the previous result given below

```
->au so8,last
```

```
Group is SO(8)
   [531] +  [52^2] +  [521^2]+ +  [521^2]- +  [52] +  [51^2]
+  [432] +  [431^2]+ +  [431^2]- +  [43] +  [42^21]+ +  [42^21]-
+  4[421] +  [41^3]+ +  [41^3]- +  [41] +  [3^21] +  [32^2]
+  [321^2]+ +  [321^2]- +  [32] +  [31^2]
REP>
−>au so8,last
Group is SO(8)
   [s;431]- +  [s;42^2]- +  [s;421^2]- +  [s;421]+ +  [s;42]-
+  [s;41^2]- +  [s;3^22]- +  [s;3^21^2]- +  [s;3^21]+ +  [s;3^2]-
+  [s;32^21]- +  [s;32^2]+ +  4[s;321]- +  [s;32]+ +  [s;31^3]-
+  [s;31^2]+ +  [s;31]- +  [s;2^3]- +  [s;2^21^2]- +  [s;2^21]+
+  [s;2^2]- +  [s;21^2]-
REP>
−>au so8,last
Group is SO(8)
   [s;431]+ +  [s;42^2]+ +  [s;421^2]+ +  [s;421]- +  [s;42]+
+  [s;41^2]+ +  [s;3^22]+ +  [s;3^21^2]+ +  [s;3^21]- +  [s;3^2]+
+  [s;32^21]+ +  [s;32^2]- +  4[s;321]+ +  [s;32]- +  [s;31^3]+
+  [s;31^2]- +  [s;31]+ +  [s;2^3]+ +  [s;2^21^2]+ +  [s;2^21]-
+  [s;2^2]+ +  [s;21^2]+
REP>
```

Note that the third application of the automorphism returns us to the original Kronecker product output. The above examples are, for SCHUR, very simple and fast. There is no difficulty in computing very large Kronecker products. For example SCHUR will rapidly produce the 292 distinct representations whose sum of multiplicities total 4508, with a maximum multiplicity of 80, for the Kronecker product $[s;4321]+\times[432]$ which is of total dimension 2,825,222,400. Likewise, for example, the $E(8)$ product $(21)\times(15\,76\,\hat{}5\,5)$ $((00000010)\times(5100010)$ in Dynkin notation) of dimension $51,677,377,007,616,000,000$ may likewise be rapidly resolved. Throughout this article we have limited the size of our examples for reasons of space rather than computational capability.

SCHUR can also compute Kronecker products, plethysms and branching rules for the unitary harmonic representations of the non−compact Lie group $Sp(2N,R)$ up to a user defined cutoff. These computations find important applications in physical problems related to quantum dots and to symplectic models of nuclei. The labelling of these infinite dimensional irreducible representations is in terms of that of the lowest weight irreducible representation of $U(1)\times U(N)$ arising in the decomposition of the irreducible representation of $Sp(2N,R)$ under the restriction $Sp(2N,R)$ 7 $U(1)\times U(N)$.

```
REP>
−>gr spr8
Group is Sp(8,R)
REP>
−>s1;2
  <s1;(2)>
REP>
−>prod s1;2,1;11
   <s2;(91)> + <s2;(82)> + <s2;(81^2 )> + 2<s2;(73)> + <s2;(721)>
```

```
    + <s2;(71)> + <s2;(64)> + <s2;(631)> + <s2;(62)> + <s2;(61^2 )>
    + <s2;(5^2)> + <s2;(541)> + 2<s2;(53)> + <s2;(521)> + <s2;(51)>
    + <s2;(431)> + <s2;(42)> + <s2;(41^2)> + <s2;(3^2)> +<s2;(321)>
    + <s2;(31)> + <s2;(21^2 )>
REP>
->sb_tex true
REP>
->last
  \+$<2;(91)>$&$ +\ <2;(82)>$&$+ \ <2;(81^2)>$&$ + \2<2;(73)>$\cr
  \+$ + \<2;(721)>$&$ +\<2;(71)>$&$ +\<2;(64)>$&$ +\<2;(631)>$\cr
  \+$ +\ <2;(62)>$&$ + \ <2;(61^2)>$&$ + \ <2;(5^2)>$&$ + \ <2;(541)>$\cr
  \+$ + \ 2<2;(53)>$&$ + \ <2;(521)>$&$ + \ <2;(51)>$&$ + \ <2;(431)>$\cr
  \+$ + \ <2;(42)>$&$ + \ <2;(41^2)>$&$ + \ <2;(3^2)>$&$ + \ <2;(321)>$\cr
  \+$ + \ <2;(31)>$&$ + \ <2;(21^2)>$\cr
REP>
```

The above verbatim output illustrates that setting the Boolean true gives TeX output in a form suitable for setting a TeX box. Indeed we need simply to prefix the last output with the TeX statement `\setbox1 = \vbox{\settabs4\columns` and add a closing parenthesis and we can print the box with `$$\box1$$` to produce the display given below

$$\begin{array}{llll}
<2;(91)> & +<2;(82)> & +<2;\,(81)^2)> & +2<2;(73)> \\
+<2;(721)> & +<2;(71)> & +<2;(64)> & +<2;(631)> \\
+<2;(62)> & +<2;\,(61)^2)> & +<2;\,(5)^2)> & +<2;(541)> \\
+2<2;(53)> & +<2;(521)> & +<2;(51)> & +<2;(431)> \\
+<2;(42)> & +<2;\,(41)^2)> & +<2;\,(3)^2)> & +<2;(321)> \\
+<2;(31)> & +<2;\,(21)^2)> & &
\end{array}$$

The following example illustrates the way in which SCHUR handles direct product groups, sets the groups and encases the inputted partitions with brackets appropriate to the various groups.

```
DP>
->gr5su5sp8so8g2e8
Groups are   SU(5) ⋆ Sp(8) ⋆ SO(8) ⋆ G(2) ⋆ E(8)
DP>
->[4321⋆4321⋆s4321−⋆42⋆21^7]
      {4321}<4321>[s;4321]-(42)(21^7 )
DP>
->dim last
Dimension = 92361051202387968
DP>
```

Below is an example of the calculation of a Kronecker product for a direct product group

```
DP>
->gr2su5so8
Groups are   SU(5) * SO(8)
DP>
```

```
->prod[21*s0-],[3*1^4-]
  {51}[s;1^4 ]- + {51}[s;1^2 ]- + {51}[s;0]- + {42}[s;1^4 ]-
+ {42}[s;1^2]- + {42}[s;0]- + {41^2}[s;1^4 ]- + {41^2 }[s;1^2]-
+ {41^2 }[s;0]- + {321}[s;1^4 ]- + {321}[s;1^2 ]- + {321}[s;0]-
DP>
->dim last
Dimension = 392000
```

7 Group−subgroup branchings

SCHUR incorporates some 59 different possibilities for obtaining group−subgroup decompositions. Many of these are unique to SCHUR and most involve algorithms that exploit the properties of Schur functions and are, unlike the common weight space algorithms, essentially rank independent. Thus for example if under $U(N) \Rightarrow U(N-1)$ the vector irreducible representation $\{1\}$ of $U(N)$ decomposes as $\{1\} \Rightarrow \{1\} + \{0\}$ then for an arbitrary irreducible representation $\{\lambda\}$ of $U(N)$ decomposes as

$$\{\lambda\} \Rightarrow \{\lambda/M\}$$

where M is the infinite set of Schur functions indexed by one part partitions. SCHUR generates the finite set of the members of the $M-$series that can skew with the Schur function indexed by the partition$\{\lambda\}$. Any Schur functions in the resultant having more than $N-1$ non−zero parts are non−standard and must be modified to produce either a null result or a standard irreducible representation label. In the above case the modification rule is simply to make null all Schur functions involving partitions of more than $N-1$ parts. For other group−subgroup structures more complicated modification rules arise but these are automatically carried out in SCHUR without requiring user intervention.

SCHUR includes branching rules for some supersymmetric groups, the various decompositions, up to a user defined cutoff, for the non−compact groups $Sp(2N,R))$ and $Mp(2N)$ required in studies of symplectic models of nuclei and quantum dots. Branching rules for group−subgroup combinations such as $O(N) \Rightarrow S(N)$ and $S(M+N) \Rightarrow S(N) \times S(N)$ are also available as well as those involving the exceptional groups. SCHUR can branch successively through a chain of groups. A special mode, the BRMode, is available for obtaining branching rules rapidly but without further processing. Some examples are given below:

```
DP>
->brm
Branch Mode
enter branching & rule numbers
->2,6
U(6) to Sp(6)
BRM>
->321
```

$$<321> \ + \ <31> \ + \ <2^2> \ + \ <21^2> \ + \ <2> \ + \ <1^2> \qquad (1)$$

```
BRM>
->stop enter branching & rule numbers
->27,4,6
SO(10) to SO(4) ⋆ SO(6)
```

```
BRM>
->2111
```

$$\begin{array}{lllll} [21]_+[1^2] & +[21]\_[1^2] & +[2][1^3]_+ & +[2][1^3]\_ & +[2][1] \\ +[1^2]_+[21] & +[1^2]_+[1^3]_+ & +[1^2]_+[1^3]\_ & +[1^2]_+[1] & +[1^2]\_[21] \\ +[1^2]\_[1^3]_+ & +[1^2]\_[1^3]\_ & +[1^2]\_[1] & +[1][21^2]_+ & +[1][21^2]\_ \\ +[1][2] & +3[1][1^2] & +[1][0] & +[0][21] & +[0][1^3]_+ \\ +[0][1^3]\_ & +[0][1] & & & \end{array} \qquad (2)$$

```
BRM>
->stop
->35,4,6
OSp(4/6) to O(4) ⋆ Sp(6)
BRM>
->21
```

$$\begin{array}{lllll} [21]<0> & +[2]<1> & +[1^2]<1> & +[1]<2> & +[1]<1^2> \\ +[1]<0> & +[0]<21> & +[0]<1> & & \end{array} \qquad (3)$$

```
BRM>
->stop
enter branching & rule numbers
->37,4
Sp(4,R) to Sp(2,R) * O(2)
BRM>
->1;11
```

$$\begin{array}{llll} <2;(12)>[10] & +<2;(12)>[6] & +<2;(12)>[2] & +<2;(10)>[8] \\ +<2;(10)>4 & +<2;(10)>[0]\# & +<2;(8)>[6] & +<2;(8)>[2] \\ +<2;(6)>4 & +<2;(6)>[0]\# & +<2;(4)>[2] & +<2;(2)>[0]\# \end{array} \qquad (4)$$

In each of the above examples the output has been set by operating SCHUR to produce TeX output. Note that in each example SCHUR has automatically encased each partition in brackets appropriate to each group and all necessary modification rules have been used to produce only standard irreducible representations . Example (3) gives an example of a branching involving the orthosymplectic group giving the complete decomposition whereas in example (4) the output has been truncated by setting the highest weight of the partitions computed. In example (4) a hash sign # has been used where required to distinguish members of associate pairs of representations of the group $O(2)$.

An illustration of the successive branching through a chain of groups is given below

```
DP>
->gr e8
Group is E(8)
DP> ->[21^7]
         (21^7)
DP>
```

```
->dim last
Dimension = 248
DP>
->br50gr1last
Group is SO(16)
```

$$[1^2]+[s;0]_+ \tag{5}$$

```
DP>
->br27,6,10gr1last
Groups are   SO(6) ⋆ SO(10)
```

$$[1^2][0] \quad [1][1] \quad [s;0]_+[s;0]_+ \quad + [s;0]_-[s;0]_- \quad [0][1] \tag{6}$$

```
DP>
->br27,6,4gr2last
Groups are   SO(6) ⋆ SO(6) ⋆ SO(4)
```

$$\begin{array}{llll} [1^2][0][0] & +[1][1][0] & +[1][0][1] & +[s;0]_+[s;0]_+[s;0]_+ \\ +[s;0]_+[s;0]_-[s;0]_- & +[s;0]_-[s;0]_+[s;0]_- & +[s;0]_-[s;0]_-[s;0]_+ & +[0][1^2][0] \\ +[0][0][1] & +[0][0][1^2]_+ & +[0][0][1^2]_- & \end{array} \tag{7}$$

```
DP>
->contract1,2last
Groups are   SO(6) ⋆ SO(4)
```

$$\begin{array}{llll} [2][0] & +[1^3]_+[s;0]_+ & +[1^3]_-[s;0]_+ & +2[1^2][s;0]_- \\ +[3][1^2][0] & +2[1][1] & +2[1][s;0]_- & +[0][1^2]_+ \\ +[0][1^2]_- & +2[0][s;0]_- & +[0][0] & \end{array} \tag{8}$$

In the above examples we have first computed the dimension of the fundamental representation of the exceptional group $E(8)$ and then made the group−subgroup restriction for $E(8) \Rightarrow SO(16)$ to give the output (5). This output then forms the input to permit the restriction $SO(16) \Rightarrow SO(6) \times SO(10)$ producing output (6). Note that both spin and tensor representations arise in the output. Next the representations of the second group, $SO(10)$, have been restricted to the subgroup $SO(6) \times SO(4)$ to produce the output (7). Finally an instruction is issued to carry out the reduction $SO(6) \times SO(6) \Rightarrow SO(6)$ to produce the output (8) and then the dimension of the output is computed which is found to agree with that of the initial $E(8)$ representation (21^7).

8 Setting of functions in SCHUR

All of the commands in the above example could be combined into a function that can be written by any suitable text editor, or even within SCHUR, and the function read in when desired. The function could read

```
gr e8
enter rv
dim[rv1]
supoutpt false
br50gr1[rv1]
br27,6,10gr1last
br27,6,4gr2last
contract1,2last
dim last
stop
```

where rv1 is any representation, or list of resentations, of nd the function read in when desired. The function could read $E(8)$.

It is possible to construct very complex functions that can be useful in many problems in group theory. In our final example we consider the very complex formula given by King [9] for the group−subgroup reduction $U(mn)\Rightarrow U(m)\times U(n)$ for mixed tensor representations of the form $\{\mu;\lambda\}$

$$\{\mu;\lambda\} = \sum_{\xi,\rho,\eta,\tau,\pi} (-1)^{w_\xi}\{(\mu/\xi\circ\rho)/\eta;(\lambda/\tilde{\xi}\circ\tau)/\eta)\}\times\{\rho/\pi;\tau/\pi\}$$

where $\circ$ indicates an inner Schur function multiplication, $\tilde{\xi}$ indicates the conjugation of a Schur function. The calculation involves several sums over Schur functions and the formation of skew products of Schur functions.

A function that carries out the task for the $U(15)\Rightarrow U(3)\times U(5)$ reduction is given below. The mixed tensor $\{\mu;\lambda\}$ is entered in terms of two Schur functions, sv1 for μ and sv2 for λ.

```
gr u15
enter sv1
enter sv2
dim[conv_s mix sv1,sv2]
setr1conv_s sv1
setr2conv_s sv2
gr5u3u3u3u5u5
rule[rv1⋆rv2⋆0⋆0⋆0]sum sk1eq conj3
rule last ch_p3
cont2,3sk last
std last
rule last sum i1eq3
rule last sum i2eq4
rule last sum sk1 eq sk2
cont1,2 mix last
std last
rule last sum sk2 eq sk3
cont2,3 mix last
supoutpt false
std last
dim last
stop
```

Reading in the function and then running it for the representation $\{2;\,1^2\}$ of $U(15)$ we obtain the output given below

```
DP>
->readfn1'unm.br'
=-
DP>
->fn1
Group is U(15)
enter sv1
->2
enter sv2
->11
Dimension = 12376
Groups are   U(3) ⋆ U(3) ⋆ U(3) ⋆ U(5) ⋆ U(5)
Groups are   U(3) ⋆ U(3) ⋆ U(5)  ⋆ U(5)
Groups are   U(3) ⋆ U(5) ⋆ U(5)
Groups are   U(3) ⋆ U(5)
Dimension = 12376
```

$$\begin{array}{llll} \{2;2\}\{2;\,1^2\} & +\{2;2\}\{1;1\} & +\{\,1^2;2\}\{\,1^2;\,1^2\} & +\{\,1^2;2\}\{1;1\} \\ +\{\,1^2;2\}\{0\} & +\{2;\,1^2\}\{2;2\} & +\{2;\,1^2\}\{1;1\} & +\{2;\,1^2\}\{0\} \\ +\{1;1\}\{2;2\} & +\{1;1\}\{\,1^2;2\} & \{1;1\}\{2;\,1^2\} & +\{1;1\}\{\,1^2;\,1^2\} \\ +3\{1;1\}\{1;1\} & +\{1;1\}\{0\} & \{0\}\{\,1^2;2\} & +\{0\}\{2;\,1^2\} \\ +\{0\}\{1;1\} & & & \end{array} \qquad (9)$$

The same function could be run for other $U(15)$ representations while other $U(mn)\Rightarrow U(m)\times U(n)$ group−subgroups can be run with very minor changes. For larger representations the output can become quite voluminous. Complete details of the construction of functions are given in the SCHUR Manual. Some further applications of SCHUR are to be found in references [8], [21], [28].

9 Concluding remarks

In the above we have given some indication of a few of the possibilities of SCHUR. It covers many problems not available in other known packages. No package is universal. While much effort has been put into ease of input, clarity of output and trapping of user errors we recall

No man is wise enough to think of all the ideas that can occur to a fool.
– (Rudolph Peierls *Bird of Passage*, Princeton 1985)

10 Availability and further information

The SCHUR package is available as a compiled C code for UNIX and DOS operating systems for IBM PC compatibles and work stations such as SUN, Hewlett−Packard and Silicon Graphics. The distribution is through S. Christensen, PO Box 16175, Chapel Hill, NC 27516 USA. Email: steve@smc.vnet.net. Additional details are available on the WEB at

`http://smc.vnet.net/Christensen.html`

and at the authors WEB site at

`http://www.phys.torun.pl/~bgw`

which contains downloadable versions of some of the papers referenced below as well as further examples of the use of SCHUR.

Acknowledgements

This work has been supported in part by a Polish KBN Grant. The assistance of Karol Grudzinski at several points is appreciated.

References

[1] G. R. E. Black, R. C. King and B. G. Wybourne. Kronecker products for compact semisimple Lie groups. *J. Phys. A: Math. Gen.* 16. pp. 1555−1589. 1983.

[2] P. Di Francesco, M. Gaudin, C. Itzykson and F. Lesage. *Laughlin's wavefunctions, Coulomb gases and expansions of the discriminant*. Preprint SPhT/93−125. Service de Physique Thèorique de Saclay. 1993.

[3] R. J. Farmer, R. C. King and B. G. Wybourne. Spectrum−generating functions for strings and superstrings. *J. Phys. A: Math. Gen.* 21. pp. 3979−4007. 1988.

[4] S. A. Fulling, R. C. King, B. G. Wybourne and C. J. Cummins. Normal forms for tensor polynomials: I. The Riemann tensor. *Class. Quantum Grav.* 9. pp. 1151−1197. 1992.

[5] K. Grudzinski and B. G. Wybourne. Computing Properties of the Non−Compact Groups $Mp(2n)$ and $Sp(2n,R)$ using SCHUR. *Proc. Third Intnl School on Theoretical Physics.* pp. 469−483. World Scientific. 1995.

[6] K. Grudzinski and B. G. Wybourne. Symplectic models of n−particle systems. *Rept. Math. Phys.* (To appear).

[7] K. Grudzinski and B. G. Wybourne. Plethysm for the noncompact group $Sp(2n,R)$ and new S−function identities. (To appear in 1996).

[8] M. G. Hirst and B. G. Wybourne. Statistical group theory and the distribution of angular momentum states II. *J. Phys. A: Math. Gen.* 19. pp. 1545−1549. 1986.

[9] R. C. King. Branching rules for classical Lie groups using tensor and spinor methods. *J. Phys. A: Math. Gen.* 8. pp. 429−449. 1975.

[10] R. C. King and A. H. A. Al−Qubanchi. Natural labelling schemes for simple roots and irreducible representations of exceptional Lie algebras. *J. Phys. A: Math. Gen.* 14. pp. 15−49. 1981.

[11] R. C. King and B. G. Wybourne. Holomorphic discrete series and harmonic series unitary irreducible representations of non−compact Lie groups: $Sp(2n,R)$, $U(p,q)$ and $SO^{*}(2n)$. *J. Phys. A: Math. Gen.* 18. pp. 3113−3139. 1985.

[12] R. C. King and B. G. Wybourne. Characters of Hecke algebras $H_n(q)$ of type A_{n-1}. *J. Phys. A:Math. Gen.* 18. pp. 1193−1197. 1991.

[13] R. C. King and B. G. Wybourne. Representations and traces of the Hecke algebras $H_n(q)$ of type A_{n-1}. *J. Math. Phys.* 33. pp. 4−14. 1992.

[14] R. C. King and B. G. Wybourne. The place of the adjoint representation in the Kronecker square of irreducible representations of simple Lie groups. (Submitted for Publication 1996).

[15] D. E. Littlewood. *The Theory of Group Characters and Matrix Representations*. 2nd ed. Clarendon Press. Oxford. 1950.

[16] I. G. Macdonald. *Symmetric Functions and Hall Polynomials.* Clarendon Press. Oxford. 1979.

[17] I. Morrison, P. W. Pieruschka and B. G. Wybourne. The interacting boson model with the exceptional groups G_2 and E_6. *J. Math. Phys.* 32. pp. 356−72. 1991.

[18] M. A. Salam and B. G. Wybourne. Shifted tableaux, Schur's Q−functions and Kronecker products of S_n spin irreps. *J. Math. Phys.* 31. pp. 1310−14. 1990.

[19] T. Scharf, J. Y. Thibon and B. G. Wybourne. Reduced notation, inner plethysms and the symmetric group. *J. Phys. A:Mat. Gen.* 26. pp. 7461−7478. 1993.

[20] T. Scharf, J. Y Thibon and B. G. Wybourne. Expansion of the Vandermonde determinant and the quantum Hall effect. *J. Phys. A:Mat. Gen.* 27. pp. 4211−4219. 1994.

[21] Q. Wang and G. E. Stedman. Time reversal symmetry and two−particle fermion many−body operators in f^N. *J. Phys. B: At. Mol. Opt. Phys.* 27. pp. 3829−3247. 1994.

[22] B. G. Wybourne. The representation space of the nuclear symplectic $Sp(6,R)$ shell model. *J. Phys. A: Math. Gen.* 25. pp. 4389−4398. 1992.

[23] B. G. Wybourne. Application of S−functions to the quantum Hall effect and quantum dots. *Rept. Math. Phys.* 34. pp. 9−16. 1994.

[24] B. G. Wybourne. Exceptional Lie groups in physics. *Lithuanian J. Phys.* 35. pp. 123−131. 1995.

[25] B. G. Wybourne. The eight−fold way of the electronic f−shell. *J. Phys. B: At. Mol. Opt. Phys.* 25. pp. 1683−1696. 1992.

[26] B. G. Wybourne and J. Meller. Enumeration of the order−14 invariants formed from the Riemann tensor. *J. Phys. A: Math. Gen.* 25. pp. 5999−6003. 1992.

[27] M. Yang and B. G. Wybourne. Squares of S−functions Special shapes. *J. Phys. A:Math. Gen.* 28. pp. 7011 −7017. 1995.

[28] M. Yang and B. G. Wybourne. Extended Poincaré supersymmetry, rotation groups and branching rules. *J. Phys. A: Math. Gen.* 19. pp. 2003−2017. 1986.