SYMMETRICA, AN OBJECT ORIENTED COMPUTER ALGEBRA SYSTEM FOR REPRESENTATIONS, COMBINATORICS AND APPLICATIONS OF SYMMETRIC GROUPS

ADALBERT KERBER AND AXEL KOHNERT
Department of Mathematics, University of Bayreuth,
D-95440 Bayreuth, Germany

ABSTRACT

This is a brief review of some of the methods and topics of SYMMETRICA, a computer algebra package devoted to the representations theory, the combinatorics, the invariant theory and the applications of symmetric groups and of related classes of groups, for example, alternating groups general linear groups and wreath products of symmetric groups.

1. The Design of SYMMETRICA

The basic philosophy of SYMMETRICA was that it should run on any computer with a C-compiler. It was designed by the second author as a collection of C-callable functions in an object oriented way [3], in order to avoid the introduction of a special language on top of the program system which would have been a contradiction to that basic philosophy. Many people contributed to it, in particular many results obtained by students in the course of writing their diploma theses or doctoral theses under the supervision of the first author were incorporated. Remarkable contributions are due to people from Paris (A. Lascoux and collaborators), from Aberystwyth (T. McDonough), from Graz (H. Fripertinger) and from Tel-Aviv (M. Muzychuk). SYMMETRICA is still in progress, at present the main emphasize lies on projective matrix representations of symmetric groups, on finite group actions, and on the constructions of discrete structures. SYMMETRICA is public domain, it can be fetched, via anonymous ftp, from

btm2x7.mat.uni-bayreuth.de:dist/SYM.tar.Z

After uncompressing and unpacking you find in the subdirectory USER (see USER.tex) a LaTeX-file of the manual and many example files. You may read information on SYMMETRICA using World Wide Web at

http://btm2xd.mat.uni-bayreuth.de/axel/symmetrica.html

1.1. Object Orientation

There are many objects that are already defined in SYMMETRICA and, of course, you may introduce your own additional objects, as it is described in the manual. The advantage of objects is that you may write

$$mult(a, b, c)$$

and the program itself looks, which kind of objects are called a and b, if they can be multiplied and how that has to be done. It then does this multiplication, and the desired result you will find under c.

1.2. The File test.c

The user of SYMMETRICA usually works with a file called $test.c$ where he writes his own program. He gets it compiled by the command $make$, and then it can be run. Here is an example:

```
#include"def.h"
#include"macro.h"
main()
{
OP a, b;
anfang();
a = callocobject();
b = callocobject();
scan(INTEGER, a); fakul(a, b); println(b);
freeall(a); freeall(b);
ende();
}
```

These lines show that first of all objects a and b are introduced, then space is reserved for them, and — after the main line

$$scan(INTEGER, a); fakul(a, b); println(b);$$

(which means "read the integer a from the monitor, evaluate factorial a, and print it out") — the space reserved for a and b is set free.

In the case when you want to evaluate the character table of S_n, say, you need only to replace $fakul(a, b)$ by $chartafel(a, b)$. You see that it is relatively easy to do easy things with SYMMETRICA . We admit that this is no wonder.

1.3. The Main Topics of SYMMETRICA

Here are some of the topics covered by SYMMETRICA :

- Ordinary irreducible and Brauer characters as well as decomposition numbers of symmetric groups,

- ordinary irreducible characters of alternating groups,
- ordinary irreducible characters of wreath products of symmetric groups,
- ordinary and modular irreducible matrix representations of symmetric groups,
- ordinary irreducible polynomial representations of general linear groups $GL_m(\mathbf{C})$,
- ordinary irreducible projective representations of symmetric groups,
- multivariate polynomials and in particular Schubert polynomials, also zonal polynomials,
- algebra of symmetric functions, including plethysm,
- Schur polynomials as well as several other series of symmetric polynomials together with base change matrices,
- zonal polynomials and Jack symmetric functions,
- cycle indicator polynomials for combinatorial enumeration,
- finite field arithmetic,
- the ordinary group algebra of the symmetric groups, including manipulation of tableaux.

Using these structures and appropriate procedures, you can evaluate irreducible characters and decompose reducible ones. You can do combinatorial enumeration to some extent, and you can also apply symmetry adapted bases by an application of irreducible matrix representations, which can be evaluated explicitly. For these procedures you can use

- integer arithmetic, including long integers which are used automatically and if necessary,
- cyclotomic fields, which are necessary, for example, if you want to evaluate characters or matrix representations of alternating groups.

2. Symmetry adapted bases

One of the most interesting applications of representation theory is the application of matrix representations to the evaluation of symmetry adapted bases, which means the decomposition of a vector space according to the symmetry of a given symmetric operator [2]. Here is a brief description what can be done by SYMMETRICA in this context.

2.1. Matrix representations

SYMMETRICA can provide the orthogonal form of representing matrices for the ordinary irreducible matrix representations of symmetric groups, and also the seminormal form via

$$odg(), sdg()$$

as well as three versions of the rational integral form:

$$bdg(), ndg(), specht\_dg()$$

The first one is the one described in H. Boerner's book, first edition, the second one is described in the book by D. E. Rutherford, in Boerner'st book, second edition, as well as in the book by G. D. James and A. Kerber [1]. The third form is due to W. Specht, and it has the advantage that it allows to evaluate a matrix representation corresponding to a skew diagram. Here is a suitable program:

$$\begin{aligned}
&scan(PARTITION, part);\\
&scan(PERMUTATION, perm);\\
&bdg(part, perm, D);\\
&tex(D);\\
&sdg(part, perm, D);\\
&tex(D);\\
&odg(part, perm, D);\\
&tex(D);\\
&specht_dg(part, perm, D);\\
&tex(D);
\end{aligned}$$

If you input the partition 2 3 (SYMMETRICA uses the French notation, i.e. a partition is coded as increasing sequence) and the permutation (in list notation) [23451], say, then you will receive an output which you can easily transform in a TEX-readible file of the following form which shows you the corresponding matrices:

$$\begin{matrix}
-1 & -1 & 1 & 1 & 0\\
-1 & 0 & 0 & 0 & 1\\
0 & -1 & 0 & 0 & 0\\
-1 & 0 & 0 & 1 & 0\\
0 & -1 & 0 & 1 & 0
\end{matrix}$$

$$\begin{matrix}
1/4 & -3/8 & 3/8 & -9/16 & 0\\
-1/2 & 1/-4 & -3/4 & 3/-8 & 0\\
-1/2 & -1/4 & 1/4 & 1/8 & -2/3\\
1 & 1/-6 & 1/-2 & 1/12 & 4/-9\\
0 & 1 & 0 & 1/-2 & 1/-3
\end{matrix}$$

$$\begin{matrix}
1/4 & -1/4\sqrt{3} & 1/4\sqrt{3} & -3/4 & 0\\
-1/4\sqrt{3} & 1/-4 & -3/4 & 1/-4\sqrt{3} & 0\\
-1/4\sqrt{3} & -1/4 & 1/4 & 1/12\sqrt{3} & -1/3\sqrt{6}\\
3/4 & 1/-12\sqrt{3} & 1/-4\sqrt{3} & 1/12 & 1/-3\sqrt{2}\\
0 & 1/3\sqrt{6} & 0 & 1/-3\sqrt{2} & 1/-3
\end{matrix}$$

$$\begin{array}{ccccc} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & -1 & 1 \end{array}$$

There are also routines for the evaluation of the ordinary irreducible polynomial representations of general linear groups $GL_m(\mathbf{C})$:

$$glmndg()$$

uses symmetry adapted bases in order to decompose the tensor product $\otimes^n(\mathbf{C}^m)$ as $GL_m(\mathbf{C})$-module. The output is a vector of polynomial matrices corresponding to the irreducible constituents (which means the ordinary irreducible polynomial matrix representations of GL_m associated with the partitions of n with at most m parts). Here you can use *odg* (if you call $0L$, see the first line in the following program) or either *bdg*() (the second line, calling $1L$) for the representations of symmetric groups, and, as far as it has been tested with *bdg*(), the polymials in the representing matrices turned out to have integral coefficients. Here is the main part of the program:

$$\begin{array}{l} scan(INTEGER, m); \\ scan(INTEGER, n); \\ glmndg(m, n, M, 0L); \\ println(M); \\ for(i = 0L; i < S\_V\_LI(M); + + i) \\ \quad glm\_homtest(m, S\_V\_I(M, i)); \\ \quad glmndg(m, n, M, 1L); \\ \quad println(M); \\ for(i = 0L; i < S\_V\_LI(M); + + i) \\ \quad glm\_homtest(m, S\_V\_I(M, i)); \end{array}$$

(please note that it contains a test for homomorphism property). In case you enter 2 for m as well as for n, you will obtain an output which needs some explanations. Here are the first few lines of the output:

$D = 3 * 1(D_1) + 1 * 1(D_2)$
[
[1
[2 :]
: 1
[1 : 1 :]
: 1
[0 : 2 :]
:]
[2
⋮

The first row of this output indicates that the tensor square $\otimes^2\mathbf{C}^2$ decomposes (as S_2–left module) into three times the representation D_1 of S_2 (which corresponds to the first partition of 2 and therefore is equal to the identity representation, as we are numbering partitions in the reverse lexicographical way) and the (alternating) representation D_2 with multiplicity 1. In the second row of the output we start giving the representing matrix for GL_2 row by row. The first bracket is the bracket of the matrix, the second bracket indicates that here the first row starts. The first entry 1 is the coefficient of the first monomial in the first entry of the representing matrix, and so on. The first monomial is indicated by [2 :] which says that only indeterminates of the form $x_{1,k}$ occur, and just the first one of them, which is x_{11}, and that its exponent is 2. The end of the first row is indicated by :]. Hence the upper left hand corner of the representing matrix is as follows:

$$\begin{pmatrix} 1 \cdot x_{11}^2 & 1 \cdot x_{11}x_{12} & 1 \cdot x_{12}^2 \\ 2 \cdots & \cdots & \cdots \\ \cdots & & \end{pmatrix}$$

A more general monomial, say $x_{12}x_{23}$ is indicated in this method by

1
[[0 : 1 : 0 :]
[0 : 0 : 1 :] :]

Another routine, namely

$$glpdg()$$

gives, for a partition of n with at most m parts, the corresponding irreducible polynomial representation of $GL_m(\mathbf{C})$. It should be noted, that this last routine uses the orthogonal form of the representations of the symmetric group. Moreover we should mention that in this case at present no homomorphism test can be applied. (The homomorphism test, which is the routine $glm\_homtest()$, as you saw already, runs as follows: Two matrices are generated, the entries of which are randomly generated integers between -5 and +5, and it is checked if the product of their images is the image of their product.)

There is also a way of getting the output in LaTeX–readable form (but up to now it works only for the representations of general linear groups). Here is a corresponding test program:

$$\begin{array}{l} \vdots \\ glmndg(m, n, M, 1L); \\ println(M); \\ for(i = 0L; i < S\_V\_LI(M); ++i) \\ \quad latex\_glm\_dar(S\_V\_I(M, i)); \\ \vdots \end{array}$$

In the case when you enter 2 for m and for n, then the LaTeX–readable part of the

output is, after processing it with LaTeX:

$$\left[\begin{array}{c} x_{11}^2 \\ \hline 2x_{11}x_{21} \\ \hline x_{21}^2 \end{array}\right]$$

$$\begin{array}{c} x_{11}x_{12} \\ \hline x_{12}x_{21} + x_{11}x_{22} \\ \hline x_{21}x_{22} \end{array}$$

$$\left[\begin{array}{c} x_{12}^2 \\ \hline 2x_{12}x_{22} \\ \hline x_{22}^2 \end{array}\right]$$

$$\left[\ -x_{12}x_{21} + x_{11}x_{22}\ \right]$$

You see that *latex_glm_dar*() gives each column of the representing matrix in a separate array.

2.2. Symmetry adaptation

The method used for the evaluation of the irreducible polynomial representations of general linear groups is that of *symmetry adapted bases.* This method is of high interest for all kinds of applications of symmetry in mathematics and sciences. Let us mention the particular case when a given matrix M over $\mathbf{C}$ has a certain symmetry, say,

$$MD(g) = D(g)M,$$

for each element g of a finite group G, and a given (possibly reducible) representation D of G over $\mathbf{C}$. Then we can find a basis such that by transformation M decomposes into

$$AMA^{-1} = \begin{pmatrix} M_1 & & & & & & \\ & \ddots & & & & & \\ & & M_1 & & & & \\ & & & \ddots & & & \\ & & & & M_r & & \\ & & & & & \ddots & \\ & & & & & & M_r \end{pmatrix}$$

Here is a program that does this job:

```
sab_input(S, SMat, M);
group_gen(S, SMat, D, Di);
sab(Di, D, B, M, mcp);
println(mcp);
println(M);
println(B);
```

You see that you need to input S (a set of generating elements of the symmetry group), $SMat$ (the set of irreducible matrices that represent the generators, for each irreducible representation contained in the representation of the symmetry group), and M, the operator that you want to decompose. Here is an example, which we describe along the input file:

input file	comment not in input file
2	2 Generators
4	generator is a permutation of S_4
2 1 3 4	the generator in list notation
4	generator is a permutation of S_4
2 3 4 1	the generator in list notation
5	number of irreps of S_4
1 1 1	size of reresenting matrix, with integer (=1) entries
1	the value of the single entry
1 1 1	size and kind of entries for the second generator
1	the value of the single entry
3 3 1	size of the second irrep
1 0 -1	entries
0 1 -1	entries
0 0 -1	entries
3 3 1	size of the second irrep for the second generator
-1 1 0	entries
-1 0 1	entries
-1 0 0	entries
2 2 1	third irrep
1 -1	entries
0 -1	entries
2 2 1	third irrep on the second generator
-1 0	entries
-1 1	entries
3 3 1	fourth irrep
1 -1 1	entries
0 -1 0	entries
0 0 -1	entries
3 3 1	fourth irrep on the second generator
1 -1 1	entries
1 0 0	entries
0 1 0	entries
1 1 1	size of last irrep
-1	single ntry
1 1 1	size of last irrep
-1	single entry
4 4 1	size of the operator to be decomposed
0 1 1 1	entries of the operator
1 0 1 1	entries of the operator
1 1 0 1	entries of the operator
1 1 1 0	entries of the operator

The first line of this input file says that there is a vector of length 2 to come which contains the generators. These generators are given as permutations (note that at present the symmetry adapted bases can be evaluated only in the case when the symmetry group is acting as a permutation group on the basis of the space in question). Therefore the 4 in the second line means that the following entry of the

If you want to check this example, then please put this input into a file *input*, say, and enter (after having the test program compiled)

$$a.out \ < \ input \ > \ output$$

You will then find at the end of the file *output* the following lines:

```
[1,3]
[
[3:]
,
[-1:]
]
[6:2:2:2:]
[6:2:2:-6:]
[6:2:-6:2:]
[6:-6:2:2:]
```

The first of these lines says that the permutation representation of S_4 decomposes into two irreducible representations D_1 and D_2 (according to the numbering in the input file), that their degrees are 1 and 3, respectively, and so the transformed operator decomposes into 4 blocks, three of which are equal. Thus the first row $[1,3]$ shows the multiplicities of the blocks in the transformed operator. Note that, by theory, these multiplicities are equal to the dimensions of the irreducibles! Finally the blocks are given, each of them just once. Therefore, in our case, the transformed operator looks as follows:

$$\begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

The final 4-rowed matrix contains the elements of the symmetry adapted basis in its rows.

3. References

1. G.D. James and A. Kerber, *The Representation Theory of the Symmetric Group* (Reading Massachusetts, 1981)
2. A. Kerber, *Algebraic Combinatorics via finite group actions* (BI Verlag Mannheim, 1991)
3. A. Kerber, A. Kohnert and A. Lascoux, *SYMMETRICA, an object oriented computer-algebra system for the symmetric group* **14**, 1992, Journal of Symbolic Computation, p. 195-203