Decomposition Matrices in GAP

Thomas Breuer

Lehrstuhl D für Mathematik
RWTH, 52056 Aachen, Germany

June 13th, 1999

Abstract

This is a sample GAP session showing computations concerning decomposition matrices. For the description of the commands for character tables, see the GAP Character Table Library [Bre12] and the chapter "Character Tables" of the GAP Reference Manual (see [GAP04]).

```
gap> LoadPackage( "ctbllib" );
true
```

1 Basis Computations with Characters of M_{11}

We start with the inspection of the Mathieu group M_{11}. Its ordinary character table and 2-modular Brauer table are fetched from the character table library using the command `CharacterTable` and the `mod` operator.

```
gap> ordtbl:= CharacterTable( "M11" );
CharacterTable( "M11" )
gap> p:= 2;
2
gap> modtbl:= ordtbl mod p;
BrauerTable( "M11", 2 )
```

The above commands assign the character tables to the variables `ordtbl` and `modtbl`, respectively.

The matrices of irreducible characters and additional information such as centralizer orders (in factorized form) and power maps are displayed using `Display`.

```
gap> Display( ordtbl );
M11

      2  4  4  1  3  .  1  3  3   .   .
      3  2  1  2  .  .  1  .  .   .   .
      5  1  .  .  .  1  .  .  .   .   .
     11  1  .  .  .  .  .  .  .   1   1

        1a 2a 3a 4a 5a 6a 8a 8b 11a 11b
     2P 1a 1a 3a 2a 5a 3a 4a 4a 11b 11a
     3P 1a 2a 1a 4a 5a 2a 8a 8b 11a 11b
```

```
      5P  1a 2a 3a 4a 1a 6a 8b 8a 11a 11b
     11P  1a 2a 3a 4a 5a 6a 8a 8b  1a  1a

X.1        1  1  1  1  1  1  1  1   1   1
X.2       10  2  1  2  . -1  .  .  -1  -1
X.3       10 -2  1  .  .  1  A -A  -1  -1
X.4       10 -2  1  .  .  1 -A  A  -1  -1
X.5       11  3  2 -1  1  . -1 -1   .   .
X.6       16  . -2  .  1  .  .  .   B  /B
X.7       16  . -2  .  1  .  .  .  /B   B
X.8       44  4 -1  . -1  1  .  .   .   .
X.9       45 -3  .  1  .  . -1 -1   1   1
X.10      55 -1  1 -1  . -1  1  1   .   .

A = E(8)+E(8)^3
  = Sqrt(-2) = i2
B = E(11)+E(11)^3+E(11)^4+E(11)^5+E(11)^9
  = (-1+Sqrt(-11))/2 = b11
gap> Display( modtbl );
M11mod2

      2  4  1  .   .   .
      3  2  2  .   .   .
      5  1  .  1   .   .
     11  1  .  .   1   1

        1a 3a 5a 11a 11b
     2P 1a 3a 5a 11b 11a
     3P 1a 1a 5a 11a 11b
     5P 1a 3a 1a 11a 11b
    11P 1a 3a 5a  1a  1a

X.1      1  1  1   1   1
X.2     10  1  .  -1  -1
X.3     16 -2  1   A  /A
X.4     16 -2  1  /A   A
X.5     44 -1 -1   .   .

A = E(11)+E(11)^3+E(11)^4+E(11)^5+E(11)^9
  = (-1+Sqrt(-11))/2 = b11
```

The restrictions of ordinary irreducible characters to conjugacy classes of element order prime to p decompose into the irreducible p-modular Brauer characters, with nonnegative integer coefficients. The matrix of coefficients, the so-called decomposition matrix, w.r.t. the given ordering of characters is computed by `DecompositionMatrix`. Again `Display` can be used to produce a formatted output.

```
gap> mat:= DecompositionMatrix( modtbl );
[ [ 1, 0, 0, 0, 0 ], [ 0, 1, 0, 0, 0 ], [ 0, 1, 0, 0, 0 ], [ 0, 1, 0, 0, 0 ],
  [ 1, 1, 0, 0, 0 ], [ 0, 0, 1, 0, 0 ], [ 0, 0, 0, 1, 0 ], [ 0, 0, 0, 0, 1 ],
  [ 1, 0, 0, 0, 1 ], [ 1, 1, 0, 0, 1 ] ]
gap> Display( mat );
[ [  1,  0,  0,  0,  0 ],
  [  0,  1,  0,  0,  0 ],
  [  0,  1,  0,  0,  0 ],
  [  0,  1,  0,  0,  0 ],
  [  1,  1,  0,  0,  0 ],
  [  0,  0,  1,  0,  0 ],
  [  0,  0,  0,  1,  0 ],
  [  0,  0,  0,  0,  1 ],
  [  1,  0,  0,  0,  1 ],
  [  1,  1,  0,  0,  1 ] ]
```

Often one is more interested in the decomposition matrices of a specific p-block. The distribution of ordinary irreducible characters to blocks is computed by `PrimeBlocks`.

```
gap> blocks:= PrimeBlocks( ordtbl, p );;
gap> blocks.block;
[ 1, 1, 1, 1, 1, 2, 3, 1, 1, 1 ]
gap> blocks.defect;
[ 4, 0, 0 ]
```

The output means that the first five and the last three irreducibles of M_{11} together lie in the principal block, which has defect 4, and the remaining two irreducibles are defect zero characters.

This information is computed from the ordinary table only, so it is available also if the Brauer table is not known. But if we have access to the brauer table then information concerning the dirstribution of ordinary and Brauer characters to blocks is computed with `BlocksInfo`, applied to the Brauer table.

```
gap> blocksinfo:= BlocksInfo( modtbl );
[ rec( basicset := [ 1, 2, 8 ],
      decinv := [ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ], defect := 4,
      modchars := [ 1, 2, 5 ], ordchars := [ 1, 2, 3, 4, 5, 8, 9, 10 ] ),
  rec( basicset := [ 6 ], decinv := [ [ 1 ] ], defect := 0, modchars := [ 3 ],
      ordchars := [ 6 ] ),
  rec( basicset := [ 7 ], decinv := [ [ 1 ] ], defect := 0, modchars := [ 4 ],
      ordchars := [ 7 ] ) ]
```

If we are interested in the decomposition matrix of the i-th block then i can be entered as second parameter of the command `DecompositionMatrix`. So the following command computes and displays the decomposition matrix of the principal block.

```
gap> Display( DecompositionMatrix( modtbl, 1 ) );
[ [  1,  0,  0 ],
  [  0,  1,  0 ],
  [  0,  1,  0 ],
  [  0,  1,  0 ],
  [  1,  1,  0 ],
  [  0,  0,  1 ],
  [  1,  0,  1 ],
  [  1,  1,  1 ] ]
```

The return value of `BlocksInfo` is a list, the i-th entry being a record that comprises the data about the i-th block. The record for the principal block can be accessed as follows.

```
gap> principalinfo:= blocksinfo[1];
rec( basicset := [ 1, 2, 8 ],
  decinv := [ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ],
  decmat := [ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 1, 0 ], [ 0, 1, 0 ],
      [ 1, 1, 0 ], [ 0, 0, 1 ], [ 1, 0, 1 ], [ 1, 1, 1 ] ], defect := 4,
  modchars := [ 1, 2, 5 ], ordchars := [ 1, 2, 3, 4, 5, 8, 9, 10 ] )
```

We see that additionally to the `blocksinfo` value we had got above, now the decomposition matrix of the block is stored in the record. This is because `DecompositionMatrix` has stored the matrix once it had been computed.

Components of the record can be accessed as follows.

```
gap> ordpos:= principalinfo.ordchars;
[ 1, 2, 3, 4, 5, 8, 9, 10 ]
```

This list is the list of positions of the characters in the principal block, w.r.t. the ordering in `ordtbl`. The ordinary irreducibles in the principal block can then be extracted using the `Irr` command as follows.

```
gap> ordchars:= Irr( ordtbl ){ ordpos };
[ Character( CharacterTable( "M11" ), [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ] ),
  Character( CharacterTable( "M11" ), [ 10, 2, 1, 2, 0, -1, 0, 0, -1, -1 ] ),
  Character( CharacterTable( "M11" ), [ 10, -2, 1, 0, 0, 1, E(8)+E(8)^3,
      -E(8)-E(8)^3, -1, -1 ] ), Character( CharacterTable( "M11" ),
    [ 10, -2, 1, 0, 0, 1, -E(8)-E(8)^3, E(8)+E(8)^3, -1, -1 ] ),
  Character( CharacterTable( "M11" ), [ 11, 3, 2, -1, 1, 0, -1, -1, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 44, 4, -1, 0, -1, 1, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 45, -3, 0, 1, 0, 0, -1, -1, 1, 1 ] ),
  Character( CharacterTable( "M11" ), [ 55, -1, 1, -1, 0, -1, 1, 1, 0, 0 ] ) ]
```

We restrict these characters to the p-regular classes, and use `Decomposition` to compute the decomposition of these characters into the Brauer characters of the principal block. The result is again the decomposition matrix, and this is exactly what `DecompositionMatrix` does.

```
gap> rest:= RestrictedClassFunctions( ordchars, modtbl );
[ Character( BrauerTable( "M11", 2 ), [ 1, 1, 1, 1, 1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 10, 1, 0, -1, -1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 10, 1, 0, -1, -1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 10, 1, 0, -1, -1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 11, 2, 1, 0, 0 ] ),
  Character( BrauerTable( "M11", 2 ), [ 44, -1, -1, 0, 0 ] ),
  Character( BrauerTable( "M11", 2 ), [ 45, 0, 0, 1, 1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 55, 1, 0, 0, 0 ] ) ]
gap> modchars:= Irr( modtbl ){ principalinfo.modchars };
[ Character( BrauerTable( "M11", 2 ), [ 1, 1, 1, 1, 1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 10, 1, 0, -1, -1 ] ),
  Character( BrauerTable( "M11", 2 ), [ 44, -1, -1, 0, 0 ] ) ]
gap> dec:= Decomposition( modchars, rest, "nonnegative" );
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 1, 0 ], [ 0, 1, 0 ], [ 1, 1, 0 ],
  [ 0, 0, 1 ], [ 1, 0, 1 ], [ 1, 1, 1 ] ]
```

For printing decomposition matrices, GAP provides the function `LaTeXStringDecompositionMatrix` that produces LaTeX code. (If the second argument is omitted then LaTeX code for printing the full decomposition matrix of the group is produced.)

```
gap> Print( LaTeXStringDecompositionMatrix( modtbl, 1 ) );
\[
\begin{array}{r|rrr} \hline
 & {\tt Y}_{1}
 & {\tt Y}_{2}
 & {\tt Y}_{5}
 \rule[-7pt]{0pt}{20pt} \\ \hline
{\tt X}_{1} & 1 & . & . \rule[0pt]{0pt}{13pt} \\
{\tt X}_{2} & . & 1 & . \\
{\tt X}_{3} & . & 1 & . \\
{\tt X}_{4} & . & 1 & . \\
{\tt X}_{5} & 1 & 1 & . \\
{\tt X}_{8} & . & . & 1 \\
{\tt X}_{9} & 1 & . & 1 \\
{\tt X}_{10} & 1 & 1 & 1 \rule[-7pt]{0pt}{5pt} \\
\hline
\end{array}
\]
```

In a LaTeX document, this looks as follows.

	$\mathtt{Y}_1$	$\mathtt{Y}_2$	$\mathtt{Y}_5$
$\mathtt{X}_1$	1	.	.
$\mathtt{X}_2$	.	1	.
$\mathtt{X}_3$	.	1	.
$\mathtt{X}_4$	.	1	.
$\mathtt{X}_5$	1	1	.
$\mathtt{X}_8$	.	.	1
$\mathtt{X}_9$	1	.	1
$\mathtt{X}_{10}$	1	1	1

2 Blocks of Symmetric Groups

The conjugacy classes and the irreducibles characters of the symmetric group of degree n, say, correspond to the partitions of n in a natural way, and it is reasonable to use these partitions as labels for the (ordinary) irreducibles in decomposition matrices.

First we need a little utility for converting a list denoting a partition into a short string. For example, `[ 2, 1, 1, 1, 1 ]` shall be written as $2\ 1^4$.

```
gap> StringOfPartition:= function( part )
>    local pair;
>
>    part:= List( Reversed( Collected( part ) ),
>                 pair -> List( pair, String ) );
>    for pair in part do
>      if pair[2] = "1" then
>        Unbind( pair[2] );
>      else
>        pair[2]:= Concatenation( "{", pair[2], "}" );
>      fi;
>    od;
>    return JoinStringsWithSeparator( List( part,
>               p -> JoinStringsWithSeparator( p, "^" ) ), " \\ " );
>  end;;
gap> StringOfPartition( [ 2, 1, 1, 1, 1 ] );
"2 \\ 1^{4}"
```

Now we put also the construction of the row labels and of the final string into a little function. Here we use that the partitions corresponding to the ordinary irreducibles of a character table of a symmetric group are stored in the attribute `CharacterParameters`, provided that the table was constructed with `CharacterTable( "Symmetric", n )` and not from the symmetric group; for small values of n, for example $n = 7$, also `CharacterTable( "S7" )` can be used. Each entry in the `CharacterParameters` list for the table of a symmetric group is a pair, the desired partition is the second entry.

```
gap> MyLaTeXMatrix:= function( symt, blocknr )
>    local alllabels, rowlabels;
>
>    alllabels:= List( CharacterParameters( OrdinaryCharacterTable( symt ) ),
>                      x -> StringOfPartition( x[2] ) );
>    rowlabels:= alllabels{ BlocksInfo( symt )[ blocknr ].ordchars };
>
>    return LaTeXStringDecompositionMatrix( symt, blocknr,
>               rec(  phi:= "\\varphi", rowlabels:= rowlabels ) );
> end;;
```

Now we get for example the following output for the principal block of the 5-modular table of S_6.

```
gap> t:= CharacterTable( "S6" ) mod 5;
BrauerTable( "A6.2_1", 5 )
gap> b:= 1;
1
gap> Print( MyLaTeXMatrix( t, b ) );
\[
\begin{array}{r|rrrr} \hline
 & \varphi_{1}
 & \varphi_{2}
 & \varphi_{7}
 & \varphi_{8}
 \rule[-7pt]{0pt}{20pt} \\ \hline
6 & 1 & . & . & . \rule[0pt]{0pt}{13pt} \\
1^{6} & . & 1 & . & . \\
3 \ 2 \ 1 & . & . & 1 & 1 \\
4 \ 2 & 1 & . & 1 & . \\
2^{2} \ 1^{2} & . & 1 & . & 1 \rule[-7pt]{0pt}{5pt} \\
\hline
\end{array}
\]
```

In the LaTeX document, this looks as follows.

	φ_1	φ_2	φ_7	φ_8
6	1	.	.	.
1^6	.	1	.	.
3 2 1	.	.	1	1
4 2	1	.	1	.
$2^2\ 1^2$	.	1	.	1

For breaking the array into several row and column portions, we could put the optional components `nrows` and `ncols` into the record that is given as the second argument of `LaTeXStringDecompositionMatrix`.

3 Computations with Atlas Tables in GAP

When dealing with ordinary character tables contained in the ATLAS of Finite Groups and Brauer character tables contained in the ATLAS of Brauer Characters, it is convenient to refer to characters and conjugacy classes via the labels chosen in these books.

For a simple group G, the ordering of characters and classes in the ATLAS table and in the GAP library table coincide. This means that the ordinary irreducible character with ATLAS label χ_i is the i-th entry in the list of irreducibles returned by the function `Irr` when called with the ordinary GAP library table. Analogously, the j-th entry in the `Irr` list of the p-modular GAP library table has ATLAS label φ_j.

The function `AtlasLabelsOfIrreducibles` returns the list of ATLAS labels of a GAP library table.

```
gap> ordtbl:= CharacterTable( "A6" );
CharacterTable( "A6" )
gap> ordlabels:= AtlasLabelsOfIrreducibles( ordtbl );
[ "\\chi_{1}", "\\chi_{2}", "\\chi_{3}", "\\chi_{4}", "\\chi_{5}",
  "\\chi_{6}", "\\chi_{7}" ]
gap> modtbl:= ordtbl mod 5;
BrauerTable( "A6", 5 )
gap> modlabels:= AtlasLabelsOfIrreducibles( modtbl );
[ "\\varphi_{1}", "\\varphi_{2}", "\\varphi_{3}", "\\varphi_{4}",
  "\\varphi_{5}" ]
```

These labels can be used to replace the default labels used by `LaTeXStringDecompositionMatrix`, via an optional record entered as third argument.

```
gap> rowlabels:= ordlabels{ BlocksInfo( modtbl )[1].ordchars };
[ "\\chi_{1}", "\\chi_{4}", "\\chi_{5}", "\\chi_{6}" ]
gap> collabels:= modlabels{ BlocksInfo( modtbl )[1].modchars };
[ "\\varphi_{1}", "\\varphi_{4}" ]
gap> options:= rec( rowlabels:= rowlabels, collabels:= collabels );;
gap> Print( LaTeXStringDecompositionMatrix( modtbl, 1, options ) );
\[
\begin{array}{r|rr} \hline
 & \varphi_{1}
 & \varphi_{4}
 \rule[-7pt]{0pt}{20pt} \\ \hline
\chi_{1} & 1 & . \rule[0pt]{0pt}{13pt} \\
\chi_{4} & . & 1 \\
\chi_{5} & . & 1 \\
\chi_{6} & 1 & 1 \rule[-7pt]{0pt}{5pt} \\
\hline
\end{array}
\]
```

In the LaTeX document, this looks as follows.

	φ_1	φ_4
χ_1	1	.
χ_4	.	1
χ_5	.	1
χ_6	1	1

The output can be influenced by several other components of the options record, for example bigger matrices can be broken into several portions of rows and columns using components `nrows` and `ncols`.

For central extensions of simple groups by a group of order at least 3, the ordering of characters in the ATLAS table and the GAP table differ in the sense that the ATLAS does not list all irreducibles. In the list of labels of the GAP table, the characters not printed in the ATLAS are denoted as algebraic conjugates of characters printed in the ATLAS.

```
gap> AtlasLabelsOfIrreducibles( CharacterTable( "3.A6" ) );
[ "\\chi_{1}", "\\chi_{2}", "\\chi_{3}", "\\chi_{4}", "\\chi_{5}",
  "\\chi_{6}", "\\chi_{7}", "\\chi_{14}", "\\chi_{14}^{\\ast 11}",
  "\\chi_{15}", "\\chi_{15}^{\\ast 11}", "\\chi_{16}", "\\chi_{16}^{\\ast 2}",
  "\\chi_{17}", "\\chi_{17}^{\\ast 2}", "\\chi_{18}", "\\chi_{18}^{\\ast 2}" ]
gap> AtlasLabelsOfIrreducibles( CharacterTable( "3.A6" ) mod 5 );
[ "\\varphi_{1}", "\\varphi_{2}", "\\varphi_{3}", "\\varphi_{4}",
  "\\varphi_{5}", "\\varphi_{10}", "\\varphi_{10}^{\\ast 2}", "\\varphi_{11}",
  "\\varphi_{11}^{\\ast 2}", "\\varphi_{12}", "\\varphi_{12}^{\\ast 2}" ]
```

Note that due to the numbering of characters in the ATLAS, certain "offsets" in the labels of a table may occur. For example, the first faithful ordinary character of $3.A_6$ has number 14 in the ATLAS but is the 8-th in the GAP table, and the 9-th character in the GAP table is the complex conjugate of this character, which is not printed in the ATLAS.

For cyclic upward extensions of perfect groups, the labels of irreducibles are formed relative to those of the derived subgroup. Namely, extensions of a character with label χ_i have labels of the form $\chi_{i,j}$, with nonnegative integers j, and an irreducible character whose restriction to the derived subgroup is the sum of pairwise different characters $\chi_{i_1}, \chi_{i_2}, \ldots, \chi_{i_k}$ has label $\chi_{i_1+i_2+\cdots+i_k}$; for the latter kind of labels, a short form χ_{i_1+} are available.

```
gap> AtlasLabelsOfIrreducibles( CharacterTable( "3.A6.2_1" ) );
[ "\\chi_{1,0}", "\\chi_{1,1}", "\\chi_{2,0}", "\\chi_{2,1}", "\\chi_{3,0}",
  "\\chi_{3,1}", "\\chi_{4+5}", "\\chi_{6,0}", "\\chi_{6,1}", "\\chi_{7,0}",
  "\\chi_{7,1}", "\\chi_{14+15\\ast 11}", "\\chi_{14\\ast 11+15}",
  "\\chi_{16+16\\ast 2}", "\\chi_{17+17\\ast 2}", "\\chi_{18+18\\ast 2}" ]
gap> AtlasLabelsOfIrreducibles( CharacterTable( "3.A6.2_1" ), "short" );
[ "\\chi_{1,0}", "\\chi_{1,1}", "\\chi_{2,0}", "\\chi_{2,1}", "\\chi_{3,0}",
  "\\chi_{3,1}", "\\chi_{4+}", "\\chi_{6,0}", "\\chi_{6,1}", "\\chi_{7,0}",
  "\\chi_{7,1}", "\\chi_{14+}", "\\chi_{15+}", "\\chi_{16+}", "\\chi_{17+}",
  "\\chi_{18+}" ]
```

Used for printing decomposition matrices, these labels occur as follows.

```
gap> ordtbl:= CharacterTable( "3.A6.2_1" );;
gap> ordlabels:= AtlasLabelsOfIrreducibles( ordtbl, "short" );;
gap> modtbl:= ordtbl mod 3;;
gap> modlabels:= AtlasLabelsOfIrreducibles( modtbl, "short" );;
gap> rowlabels:= ordlabels{ BlocksInfo( modtbl )[1].ordchars };;
gap> collabels:= modlabels{ BlocksInfo( modtbl )[1].modchars };;
gap> options:= rec( rowlabels:= rowlabels, collabels:= collabels );;
gap> Print( LaTeXStringDecompositionMatrix( modtbl, 1, options ) );
\[
\begin{array}{r|rrrrr} \hline
 & \varphi_{1,0}
 & \varphi_{1,1}
 & \varphi_{2+}
 & \varphi_{4,0}
```

```
  & \varphi_{4,1}
  \rule[-7pt]{0pt}{20pt} \\ \hline
 \chi_{1,0} & 1 & . & . & . & . \rule[0pt]{0pt}{13pt} \\
 \chi_{1,1} & . & 1 & . & . & . \\
 \chi_{2,0} & 1 & . & . & 1 & . \\
 \chi_{2,1} & . & 1 & . & . & 1 \\
 \chi_{3,0} & 1 & . & . & . & 1 \\
 \chi_{3,1} & . & 1 & . & 1 & . \\
 \chi_{4+} & 1 & 1 & 1 & 1 & 1 \\
 \chi_{7,0} & . & . & 1 & 1 & . \\
 \chi_{7,1} & . & . & 1 & . & 1 \\
 \chi_{14+} & . & . & 1 & . & . \\
 \chi_{15+} & . & . & 1 & . & . \\
 \chi_{16+} & 2 & 2 & . & 1 & 1 \\
 \chi_{18+} & 1 & 1 & 2 & 2 & 2 \rule[-7pt]{0pt}{5pt} \\
 \hline
 \end{array}
 \]
```

In the LaTeX document, this looks as follows.

	$\varphi_{1,0}$	$\varphi_{1,1}$	φ_{2+}	$\varphi_{4,0}$	$\varphi_{4,1}$
$\chi_{1,0}$	1	.	.	.	.
$\chi_{1,1}$	.	1	.	.	.
$\chi_{2,0}$	1	.	.	1	.
$\chi_{2,1}$	.	1	.	.	1
$\chi_{3,0}$	1	.	.	.	1
$\chi_{3,1}$	.	1	.	1	.
χ_{4+}	1	1	1	1	1
$\chi_{7,0}$	.	.	1	1	.
$\chi_{7,1}$	.	.	1	.	1
χ_{14+}	.	.	1	.	.
χ_{15+}	.	.	1	.	.
χ_{16+}	2	2	.	1	1
χ_{18+}	1	1	2	2	2

For further questions about GAP, consult the GAP Reference Manual.

References

[Bre12] T. Breuer, *The GAP Character Table Library, Version 1.2*, `http://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib`, Mar 2012, GAP package.

[GAP04] The GAP Group, *GAP–Groups, Algorithms, and Programming, Version 4.4*, 2004, `http://www.gap-system.org`.