Multiplicity-Free Permutation Characters in GAP

Thomas Breuer

Lehrstuhl D für Mathematik
RWTH, 52056 Aachen, Germany

October 6th, 2000

Abstract

This note shows a few examples of GAP computations concerning multiplicity-free permutation characters, with an emphasis on the classification of the faithful multiplicity-free permutation characters of the sporadic simple groups and their automorphism groups given in [BL96].

For examples on GAP computations with permutation characters in general, see the note [Bre].

For further questions about GAP, consult its Reference Manual; in particular, for the description of the commands for character tables, see the chapter "Character Tables".

Section 1 of this note shows how to interpret the individual data available in the database. In Section 2, the main idea is to gather information from the database as a whole, by filtering items with suitable properties. Finally, Section 3 gives an impression how GAP can be used to obtain results such as the classification of described in [BL96].

Contents

1 The Database of Multiplicity-Free Characters 1

1.1 The Faithful Multiplicity-Free Permutation Characters of M_{11} 2

1.2 The Faithful Multiplicity-Free Permutation Characters of $M_{12}.2$ 4

2 Using the Database 5

3 Using the Functions to Compute Multiplicity-Free Permutation Characters 7

3.1 Using Tables of Marks 10

3.2 Dealing with Possible Permutation Characters 11

1 The Database of Multiplicity-Free Characters

The database lists, for each group G that is either a sporadic simple group or an automorphism group of a sporadic simple group, a description of all conjugacy classes of subgroups H of G such that the action of G on the right cosets of H is a faithful and multiplicity-free permutation representation of G, plus the permutation character of this representation. The format how this information is stored is explained below, subtleties such as possibly equal characters for different classes of subgroups are discussed in Section 2.

(A GAP database providing more information about most of these representations is in preparation; this will cover, i.a., the character tables of the endomorphism rings of these representations and the permutation representations themselves.)

The data is stored in the file `multfree.dat`, which is part of the Character Table Library [Bre12] of the GAP system [GAP04] as well as the file you are currently reading. We load this GAP package and the data file into GAP 4. Afterwards the function `MultFreePermChars` is available.

```
gap> LoadPackage( "ctbllib" );
true
gap> if not IsBound( MultFreePermChars ) then
> ReadPackage( "ctbllib", "tst/multfree.dat" );
--------------------------------------------------------------------
Loading the Database of Multiplicity-Free Permutation Characters
of the Sporadic Simple Groups and Their Automorphism Groups,
by T. Breuer and K. Lux;
call ‘MultFreePermChars( <name> )’ for accessing the data
for the group whose character table has identifier <name>.
--------------------------------------------------------------------
> fi;
```

1.1 The Faithful Multiplicity-Free Permutation Characters of M_{11}

We start with the inspection of the Mathieu group M_{11}, as an example of a **simple** group that is dealt with in the database.

```
gap> info:= MultFreePermChars( "M11" );
[ rec( ATLAS := "1a+10a", character := Character( CharacterTable( "M11" ),
        [ 11, 3, 2, 3, 1, 0, 1, 1, 0, 0 ] ), group := "$M_{11}$", rank := 2,
      subgroup := "$A_6.2_3$" ),
  rec( ATLAS := "1a+10a+11a",
      character := Character( CharacterTable( "M11" ),
        [ 22, 6, 4, 2, 2, 0, 0, 0, 0, 0 ] ), group := "$M_{11}$", rank := 3,
      subgroup := "$A_6 \\leq A_6.2_3$" ),
  rec( ATLAS := "1a+11a", character := Character( CharacterTable( "M11" ),
        [ 12, 4, 3, 0, 2, 1, 0, 0, 1, 1 ] ), group := "$M_{11}$", rank := 2,
      subgroup := "$L_2(11)$" ),
  rec( ATLAS := "1a+11a+16ab+45a+55a",
      character := Character( CharacterTable( "M11" ),
        [ 144, 0, 0, 0, 4, 0, 0, 0, 1, 1 ] ), group := "$M_{11}$", rank := 6,
      subgroup := "$11:5 \\leq L_2(11)$" ),
  rec( ATLAS := "1a+10a+44a",
      character := Character( CharacterTable( "M11" ),
        [ 55, 7, 1, 3, 0, 1, 1, 1, 0, 0 ] ), group := "$M_{11}$", rank := 3,
      subgroup := "$3^2:Q_8.2$" ),
  rec( ATLAS := "1a+10a+44a+55a",
      character := Character( CharacterTable( "M11" ),
        [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ), group := "$M_{11}$", rank := 4,
      subgroup := "$3^2:8 \\leq 3^2:Q_8.2$" ),
  rec( ATLAS := "1a+10a+11a+44a",
      character := Character( CharacterTable( "M11" ),
        [ 66, 10, 3, 2, 1, 1, 0, 0, 0, 0 ] ), group := "$M_{11}$", rank := 4,
      subgroup := "$A_5.2$" ) ]
gap> List( info, x -> x.rank );
[ 2, 3, 2, 6, 3, 4, 4 ]
```

```
gap> chars:= List( info, x -> x.character );;
gap> degrees:= List( chars, x -> x[1] );
[ 11, 22, 12, 144, 55, 110, 66 ]
```

We see that M_{11} has seven multiplicity-free permutation characters, of the ranks and degrees listed above. (Note that for **multiplicity-free** permutation characters, the rank is equal to the number of irreducible constituents.) More precisely, there are exactly seven conjugacy classes of subgroups of M_{11} such that the permutation action on the cosets of these subgroups is faithful and multiplicity-free.

For displaying the characters compatibly with the character table of M_{11}, we can use the `Display` operation. Note that the column and row ordering of character tables in GAP is compatible with that of the tables in the ATLAS of Finite Groups ([CCN+85]).

```
gap> tbl:= CharacterTable( "M11" );
CharacterTable( "M11" )
gap> Display( tbl, rec( chars:= chars ) );
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

Y.1      11  3  2  3  1  .  1  1   .   .
Y.2      22  6  4  2  2  .  .  .   .   .
Y.3      12  4  3  .  2  1  .  .   1   1
Y.4     144  .  .  .  4  .  .  .   1   1
Y.5      55  7  1  3  .  1  1  1   .   .
Y.6     110  6  2  2  .  .  2  2   .   .
Y.7      66 10  3  2  1  1  .  .   .   .
```

The `subgroup` component of each record in `info` describes the isomorphism type of a subgroup U of M_{11} such that the value π of the `character` component is induced from the trivial character of U; in other words, U is a point stabilizer of the permutation representation of M_{11} with character π.

(Contrary to this example, in general it may happen that different classes of subgroups induce the same permutation character, and that these subgroups may also be nonisomorphic; see Section 2 for details.)

```
gap> subgroups:= List( info, x -> x.subgroup );
[ "$A_6.2_3$", "$A_6 \\leq A_6.2_3$", "$L_2(11)$", "$11:5 \\leq L_2(11)$",
  "$3^2:Q_8.2$", "$3^2:8 \\leq 3^2:Q_8.2$", "$A_5.2$" ]
```

Each entry is a LaTeX format string that is either a name of the point stabilizer or has the form `<U> \leq <M>` where `<M>` is the name of a maximal subgroup containing the point stabilizer `<U>` as a proper subgroup; in the former case, the point stabilizer is itself maximal.

Note that a backslash occurring in a `subgroup` string is escaped by another backslash; but only a single backslash is printed when the string is printed via the function `Print`.

```
gap> Print( subgroups[2], "\n" );
$A_6 \leq A_6.2_3$
```

Finally, the `ATLAS` component of each record in `info` describes the `character` value in terms of its irreducible constituents, as is computed by the function `PermCharInfo`. Examples can be found in Section 3; for details about the output format, see the documentation for this function in the GAP Reference Manual.

1.2 The Faithful Multiplicity-Free Permutation Characters of $M_{12}.2$

The automorphism group of a sporadic simple group G is either equal to G or an upward extension of G by an outer automorphism of order 2. The **nonsimple** automorphism group $M_{12}.2$ of the Mathieu group M_{12} serves as an example of the latter situation.

In addition to the aspects mentioned in Section 1.1, here we meet the situation that a permutation character either is induced from a permutation character of M_{12} or extends such a (not necessarily multiplicity-free) permutation character. The former case occurs exactly if the corresponding point stabilizer lies in M_{12}.

```
gap> info:= MultFreePermChars( "M12.2" );;
gap> Length( info );
13
gap> info[1];
rec( ATLAS := "1a^{\\pm}+11ab",
  character := Character( CharacterTable( "M12.2" ),
    [ 24, 0, 8, 6, 0, 4, 4, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ),
  group := "$M_{12}.2$", rank := 3, subgroup := "$M_{11}$" )
gap> info[2];
rec( ATLAS := "1a^++11ab+55a^++66a^+",
  character := Character( CharacterTable( "M12.2" ),
    [ 144, 0, 16, 9, 0, 0, 4, 0, 1, 0, 0, 1, 12, 4, 0, 0, 2, 2, 0, 1, 1 ] ),
  group := "$M_{12}.2$", rank := 4, subgroup := "$L_2(11).2$" )
```

The first character in the list `info` is induced from the trivial character of a subgroup of type M_{11} inside M_{12}, the second character is induced from the trivial character of a $L_2(11).2$ subgroup whose intersection with M_{12} is of type $L_2(11)$.

We can distinguish the two kinds of permutation characters by explicitly using the character tables; for example, a permutation character is induced from a subgroup of a normal subgroup if and only if it vanishes outside the classes forming this subgroup.

```
gap> m12:= CharacterTable( "M12" );;
gap> m122:= UnderlyingCharacterTable( info[1].character );;
gap> fus:= GetFusionMap( m12, m122 );
[ 1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 10, 10, 11, 12, 12 ]
gap> outer:= Difference( [ 1 .. NrConjugacyClasses( m122 ) ], fus );
[ 13, 14, 15, 16, 17, 18, 19, 20, 21 ]
gap> info[1].character{ outer };
[ 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> info[2].character{ outer };
[ 12, 4, 0, 0, 2, 2, 0, 1, 1 ]
```

A perhaps easier way is to look at the `ATLAS` components of the `info` records. Namely, the characters induced from subgroups of M_{12} have both linear characters of $M_{12}.2$ as constituents, which is expressed by the substring `"1a^{\\pm}"`.

More generally, the `ATLAS` component lists the irreducible constituents of the restriction to M_{12}, where the two extensions of a character to $M_{12}.2$ are distinguished by a superscript $+$, $-$, or $\pm$; the latter means that both extensions occur.

The ATLAS components describing the constituents relative to a subgroup of index 2 can be computed using the GAP function PermCharInfoRelative, see Section 3.

It should be noted that the \leq substrings in the subgroup component cannot be used to distinguish the two kinds of permutation characters, since these substrings refer only to maximal subgroups **different from** M_{12}. Examples are the first entry in info (see above), the fourth entry (containing a character that is induced from a subgroup of type $A_6.2_2$ which lies in a $A_6.2^2$ subgroup that is maximal in M_{11}), and the nineth entry (containing a character induced from a subgroup of index 2 in a $(2^2 \times A_5).2$ subgroup that is maximal in $M_{12}.2$.

```
gap> info[4];
rec( ATLAS := "1a^{\\pm}+11ab+54a^{\\pm}+66a^{\\pm}",
  character := Character( CharacterTable( "M12.2" ),
    [ 264, 24, 24, 12, 0, 4, 4, 0, 0, 2, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ),
  group := "$M_{12}.2$", rank := 7, subgroup := "$A_6.2_2 \\leq A_6.2^2$" )
gap> info[9];
rec(
  ATLAS := "1a^++16ab+45a^++54a^{\\pm}+55a^-+66a^{\\pm}+99a^-+144a^++176a^-",
  character := Character( CharacterTable( "M12.2" ),
    [ 792, 32, 24, 0, 6, 0, 2, 2, 0, 0, 2, 0, 0, 0, 8, 0, 0, 0, 2, 0, 0 ] ),
  group := "$M_{12}.2$", rank := 11,
  subgroup := "$(2 \\times A_5).2 \\leq (2^2 \\times A_5).2$" )
```

2 Using the Database

In this section, we study the complete list of multiplicity-free permutation characters of the sporadic simple groups and their automorphism groups as a whole.

```
gap> info:= MultFreePermChars( "all" );;
gap> Length( info );
267
gap> Length( Set( info ) );
262
gap> chars:= List( info, x -> x.character );;
gap> Length( Set( chars ) );
261
```

We see that there are exactly 267 conjugacy classes of subgroups such that the permutation representation on the cosets is multiplicity-free. Only 262 of the info records are different, and there is exactly one case where two different info records belong to the same permutation character.

Let us look where these multiple entries arise.

```
gap> distrib:= List( info, x -> Position( chars, x.character ) );;
gap> ambiguous:= Filtered( InverseMap( distrib ), IsList );
[ [ 12, 15 ], [ 40, 41 ], [ 83, 84 ], [ 88, 90 ], [ 132, 133 ], [ 202, 203 ] ]
gap> except:= Filtered( ambiguous, x -> info[ x[1] ] <> info[ x[2] ] );
[ [ 83, 84 ] ]
gap> ambiguous:= Difference( ambiguous, except );;
gap> info{ except[1] };
[ rec( ATLAS := "1a+22a+230a",
      character := Character( CharacterTable( "M23" ),
        [ 253, 29, 10, 5, 3, 2, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0 ] ),
      group := "$M_{23}$", rank := 3, subgroup := "$L_3(4).2_2$" ),
  rec( ATLAS := "1a+22a+230a",
      character := Character( CharacterTable( "M23" ),
        [ 253, 29, 10, 5, 3, 2, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0 ] ),
      group := "$M_{23}$", rank := 3, subgroup := "$2^4:A_7$" ) ]
```

So the Mathieu group M_{23} contains two classes of maximal subgroups, of the structures $L_3(4).2_2$ and $2^4 : A_7$, respectively, such that the characters of the permutation representations on the cosets of these subgroups are equal.

Furthermore, it is a consequence of the classification in [BL96] that in all cases except this one, the isomorphism types of the point stabilizers are uniquely determined by the permutation characters.

```
gap> ambiginfo:= info{ List( ambiguous, x -> x[1] ) };;
gap> for pair in ambiginfo do
>      Print( pair.group, ", ", pair.subgroup, ", ", pair.ATLAS, "\n" );
> od;
$M_{12}$, $A_6.2_1 \leq A_6.2^2$, 1a+11ab+54a+55a
$M_{22}$, $A_7$, 1a+21a+154a
$HS$, $U_3(5).2$, 1a+175a
$McL$, $M_{22}$, 1a+22a+252a+1750a
$Fi_{22}$, $O_7(3)$, 1a+429a+13650a
```

In the other five cases of ambiguities, the whole `info` records are equal, and from the above list we conclude that for each pair, the point stabilizers are isomorphic. In fact the subgroups are conjugate in the outer automorphism groups of the simple groups involved.

Next let us look at the distribution of ranks.

```
gap> Collected( List( info, x -> x.rank ) );
[ [ 2, 11 ], [ 3, 31 ], [ 4, 25 ], [ 5, 43 ], [ 6, 24 ], [ 7, 21 ],
  [ 8, 26 ], [ 9, 16 ], [ 10, 17 ], [ 11, 9 ], [ 12, 9 ], [ 13, 8 ],
  [ 14, 4 ], [ 15, 3 ], [ 16, 3 ], [ 17, 5 ], [ 18, 5 ], [ 19, 2 ],
  [ 20, 2 ], [ 23, 1 ], [ 26, 1 ], [ 34, 1 ] ]
gap> max:= Filtered( info, x -> x.rank = 34 );;
gap> max[1].group;  max[1].subgroup;  max[1].character[1];
"$F_{3+}.2$"
"$O_{10}^-(2) \\leq O_{10}^-(2).2$"
100354720284
```

The maximal rank, 34, is attained for a degree 100 354 720 284 character of $F_{3+}.2 = Fi_{24}$.

For the nonsimple automorphism groups of sporadic simple groups, the simple group G involved is of index 2, and each permutation characters either is induced from a character of G or extends a permutation character of G.

```
gap> nonsimple:= Filtered( info,
>        x -> not IsSimple( UnderlyingCharacterTable( x.character ) ) );;
gap> Length( nonsimple );
120
gap> ind:= Filtered( nonsimple, x -> ScalarProduct( x.character,
>              Irr( UnderlyingCharacterTable( x.character ) )[2] ) = 1 );;
gap> Length( ind );
48
```

There are exactly 120 multiplicity-free permutation characters of nonsimple automorphism groups of sporadic simple groups, and 48 of them are induced from characters of the simple groups. (Note that the second irreducible character of the GAP character tables in question is the unique nontrivial linear character.)

```
gap> ind[1];
rec( ATLAS := "1a^{\\pm}+11ab",
  character := Character( CharacterTable( "M12.2" ),
    [ 24, 0, 8, 6, 0, 4, 4, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ),
  group := "$M_{12}.2$", rank := 3, subgroup := "$M_{11}$" )
gap> ForAll( ind, x -> x.ATLAS{ [ 1 .. 8 ] } = "1a^{\\pm}" );
true
```

Another possibility to select the induced characters is to check whether the initial part of the ATLAS component is the string "1a^{\\pm}".

3 Using the Functions to Compute Multiplicity-Free Permutation Characters

The functions MultFreeFromTomAndTable and MultFree will be used later on.

(The functions can also be found in the file multfree.g, which can be downloaded from the same webpage where also this file can be found.)

For a character table tbl for which the table of marks is available in the GAP library, the function MultFreeFromTomAndTable returns the list of all multiplicity-free permutation characters of tbl.

```
gap> BindGlobal( "MultFreeFromTomAndTable", function( tbl )
>     local tom,     # the table of marks
>           fus,     # fusion map from ‘t’ to ‘tom’
>           perms;   # perm. characters of ‘t’
>
>     if HasFusionToTom( tbl ) or HasUnderlyingGroup( tbl ) then
>       tom:= TableOfMarks( tbl );
>     else
>       Error( "no table of marks for character table <tbl> available" );
>     fi;
>     fus:= FusionCharTableTom( tbl, tom );
>     if fus = fail then
>       Error( "no unique fusion from <tbl> to the table of marks" );
>     fi;
>     perms:= PermCharsTom( tbl, tom );
>     return Filtered( perms,
>                x -> ForAll( Irr( tbl ),
>                             y -> ScalarProduct( tbl, x, y ) <= 1 ) );
>     end );
```

TestPerm calls the GAP library functions TestPerm1, TestPerm2, and TestPerm3; the return value is true if the argument pi is a possible permutation character of the character table tbl, and false otherwise.

```
gap> BindGlobal( "TestPerm", function( tbl, pi )
>     return     TestPerm1( tbl, pi ) = 0
>            and TestPerm2( tbl, pi ) = 0
>            and not IsEmpty( TestPerm3( tbl, [ pi ] ) );
>     end );
```

Let H be a character table, S be a list of characters of H, psi a character of H, scprS a matrix, the i-th entry being the coefficients of the decomposition of the induced character of S[i] to a supergroup G, say, of H, scprpsi the decomposition of psi induced to G, and k a positive integer.

`CharactersInducingWithBoundedMultiplicity` returns the list $C($ `S`, `psi`, `k` $)$; this is the list of all those characters `psi` $+ \vartheta$ of multiplicity at most `k` such that all constituents of ϑ are contained in `S`.

```
gap> DeclareGlobalFunction( "CharactersInducingWithBoundedMultiplicity" );

gap> InstallGlobalFunction( CharactersInducingWithBoundedMultiplicity,
>     function( H, S, psi, scprS, scprpsi, k )
>     local result,       # the list $S( .. )$
>           chi,          # $\chi$
>           scprchi,      # decomposition of $\chi^G$
>           i,            # loop from `1' to `k'
>           allowed,      # indices of possible constituents
>           Sprime,       # $S^{\prime}_i$
>           scprSprime;   # decomposition of characters in $S^{\prime}_i$,
>                         # induced to $G$
>
>     if IsEmpty( S ) then
>
>       # Test whether `psi' is a possible permutation character.
>       if TestPerm( H, psi ) then
>         result:= [ psi ];
>       else
>         result:= [];
>       fi;
>
>     else
>
>       # Fix a character $\chi$.
>       chi     := S[1];
>       scprchi := scprS[1];
>
>       # Form the union.
>       result:= CharactersInducingWithBoundedMultiplicity( H,
>                    S{ [ 2 .. Length( S ) ] }, psi,
>                    scprS{ [ 2 .. Length( S ) ] }, scprpsi, k );
>       for i in [ 1 .. k ] do
>         allowed    := Filtered( [ 2 .. Length( S ) ],
>                          j -> Maximum( i * scprchi + scprS[j] ) <= k );
>         Sprime     := S{ allowed };
>         scprSprime := scprS{ allowed };
>
>         Append( result, CharactersInducingWithBoundedMultiplicity( H,
>                             Sprime, psi + i * chi,
>                             scprSprime, scprpsi + i * scprchi, k ) );
>       od;
>
>     fi;
>
>     return result;
>     end );
```

Let `G` and `H` be character tables of groups $G$ and H, respectively, such that H is a subgroup of $G$ and the class fusion from `H` to `G` is stored on `H`. `MultAtMost` returns the list of all characters φ^G of $G$ of multiplicity at most `k` such that φ is a possible permutation character of H.

```
gap> BindGlobal( "MultAtMost", function( G, H, k )
>     local triv,     # $1_H$
>           permch,   # $(1_H)^G$
>           scpr1H,   # decomposition of $(1_H)^G$
>           rat,      # rational irreducible characters of $H$
>           ind,      # induced rational irreducible characters
>           mat,      # decomposition of ‘ind’
>           allowed,  # indices of possible constituents
>           S0,       # $S_0$
>           scprS0,   # decomposition of characters in $S_0$,
>                     # induced to $G$, with $Irr(G)$
>           cand;     # list of multiplicity-free candidates, result
>
>     # Compute $(1_H)^G$ and its decomposition into irreducibles of $G$.
>     triv   := TrivialCharacter( H );
>     permch := Induced( H, G, [ triv ] );
>     scpr1H := MatScalarProducts( G, Irr( G ), permch )[1];
>
>     # If $(1_H)^G$ has multiplicity larger than ‘k’ then we are done.
>     if Maximum( scpr1H ) > k then
>       return [];
>     fi;
>
>     # Compute the set $S_0$ of all possible nontrivial
>     # rational constituents of a candidate of multiplicity at most ‘k’,
>     # that is, all those rational irreducible characters of
>     # $H$ that induce to $G$ with multiplicity at most ‘k’.
>     rat:= RationalizedMat( Irr( H ) );
>     ind:= Induced( H, G, rat );
>     mat:= MatScalarProducts( G, Irr( G ), ind );
>     allowed:= Filtered( [ 1.. Length( mat ) ],
>                         x -> Maximum( mat[x] + scpr1H ) <= k );
>     S0     := rat{ allowed };
>     scprS0 := mat{ allowed };
>
>     # Compute $C( S_0, 1_H, k )$.
>     cand:= CharactersInducingWithBoundedMultiplicity( H,
>                S0, triv, scprS0, scpr1H, k );
>
>     # Induce the candidates to $G$, and return the sorted list.
>     cand:= Induced( H, G, cand );
>     Sort( cand );
>     return cand;
>     end );
```

MultFree returns MultAtMost(G, H, 1).

```
gap> BindGlobal( "MultFree", function( G, H )
>     return MultAtMost( G, H, 1 );
>     end );
```

Let tbl be a character table with known Maxes value, and k a positive integer. The function PossiblePermutationCharactersWithBoundedMultiplicity returns a record with the following components.

identifier the `Identifier` value of `tbl`,

maxnames the list of names of the maximal subgroups of `tbl`,

permcand at the i-th position the list of those possible permutation characters of `tbl` whose multiplicity is at most `k` and which are induced from the i-th maximal subgroup of `tbl`, and

k the given bound `k` for the multiplicity.

```
gap> BindGlobal( "PossiblePermutationCharactersWithBoundedMultiplicity",
>     function( tbl, k )
>     local permcand, # list of all mult. free perm. character candidates
>           maxname,  # loop over tables of maximal subgroups
>           max;      # one table of a maximal subgroup
>
>     if not HasMaxes( tbl ) then
>       return fail;
>     fi;
>
>     permcand:= [];
>
>     # Loop over the tables of maximal subgroups.
>     for maxname in Maxes( tbl ) do
>
>       max:= CharacterTable( maxname );
>       if max = fail or GetFusionMap( max, tbl ) = fail then
>
>         Print( "#E  no fusion ‘", maxname, "’ -> ‘", Identifier( tbl ),
>                "’ stored\n" );
>         Add( permcand, Unknown() );
>
>       else
>
>         # Compute the possible perm. characters inducing through ‘max’.
>         Add( permcand, MultAtMost( tbl, max, k ) );
>
>       fi;
>     od;
>
>     # Return the result record.
>     return rec( identifier := Identifier( tbl ),
>                 maxnames   := Maxes( tbl ),
>                 permcand   := permcand,
>                 k          := k );
>     end );
```

3.1 Using Tables of Marks

As a small example for the computation of multiplicity-free permutation characters from the table of marks of a group, we consider the alternating group A_5. Its character table as well as its table of marks are accessible from the respective GAP library, via the identifier `A5`.

```
gap> tbl:= CharacterTable( "A5" );;
gap> chars:= MultFreeFromTomAndTable( tbl );
[ Character( CharacterTable( "A5" ), [ 12, 0, 0, 2, 2 ] ),
  Character( CharacterTable( "A5" ), [ 10, 2, 1, 0, 0 ] ),
  Character( CharacterTable( "A5" ), [ 6, 2, 0, 1, 1 ] ),
  Character( CharacterTable( "A5" ), [ 5, 1, 2, 0, 0 ] ),
  Character( CharacterTable( "A5" ), [ 1, 1, 1, 1, 1 ] ) ]
```

As the GAP databases do not provide information about the isomorphism types of arbitrary subgroups, there is no way to compute automatically the `subgroup` strings as contained in the database of multiplicity-free permutation characters (cf. Section 1). Of course it is easy to see that the above characters of A_5 are induced from the trivial characters of the cyclic group of order 5, the dihedral groups of orders 6 and 10, the alternating group A_4, and the group A_5 itself, respectively.

The `ATLAS` information used in the database records can be computed using the GAP function `PermCharInfo`.

```
gap> PermCharInfo( tbl, chars ).ATLAS;
[ "1a+3ab+5a", "1a+4a+5a", "1a+5a", "1a+4a", "1a" ]
```

As an example for a nonsimple group, we repeat the computation of all multiplicity-free permutation characters of $M_{12}.2$, using the GAP table of marks.

```
gap> tbl:= CharacterTable( "M12.2" );;
gap> chars:= MultFreeFromTomAndTable( tbl );;
gap> lib:= MultFreePermChars( "M12.2" );;
gap> Length( lib );  Length( chars );
13
15
gap> Difference( chars, List( lib, x -> x.character ) );
[ Character( CharacterTable( "M12.2" ), [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
      1, 1, 1, 1, 1, 1, 1, 1, 1 ] ), Character( CharacterTable( "M12.2" ),
    [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
```

This confirms the classification for $M_{12}.2$, since the additional characters found from the table of marks are not faithful.

The corresponding `ATLAS` information is computed using the GAP function `PermCharInfoRelative`, since the constituents shall be listed relative to the simple group M_{12}.

```
gap> tblsimple:= CharacterTable( "M12" );;
gap> PermCharInfoRelative( tblsimple, tbl, chars ).ATLAS;
[ "1a^++16ab+45a^-+54a^{\\pm}+55a^{\\pm}bc+66a^++99a^{\\pm}+144a^++176a^+",
  "1a^++11ab+45a^-+54a^{\\pm}+55a^++66a^{\\pm}+99a^-+120a^{\\pm}+144a^{\\pm}",
  "1a^{\\pm}+11ab+45a^{\\pm}+54a^{\\pm}+55a^{\\pm}bc+99a^{\\pm}+120a^{\\pm}",
  "1a^++16ab+45a^++54a^{\\pm}+55a^-+66a^{\\pm}+99a^-+144a^++176a^-",
  "1a^++16ab+45a^-+54a^{\\pm}+66a^++99a^-+144a^+",
  "1a^++11ab+54a^{\\pm}+55a^++66a^++99a^-+144a^+",
  "1a^{\\pm}+11ab+54a^{\\pm}+55a^{\\pm}+99a^{\\pm}",
  "1a^++16ab+45a^++54a^{\\pm}+66a^++144a^+",
  "1a^{\\pm}+11ab+54a^{\\pm}+66a^{\\pm}", "1a^++16ab+45a^++66a^+",
  "1a^++11ab+55a^++66a^+", "1a^{\\pm}+11ab+54a^{\\pm}", "1a^{\\pm}+11ab",
  "1a^{\\pm}", "1a^+" ]
```

For more information about tables of marks, see [Pfe97].

3.2 Dealing with Possible Permutation Characters

In this section, we deal with **possible permutation characters**, that is, characters that have certain properties of permutation characters but for which no subgroups need to exist from whose trivial characters they are induced. For more information about such characters, see the section "Possible Permutation Characters" in the GAP Reference Manual, the paper [BP98], and the note [Bre].

We can compute possible permutation characters from the character table of the group in question, the table of marks need not be available. The problem is of course that for classifying the permutation characters, we have to decide which of the candidates are in fact permutation characters.

Here we show only two small examples that could also be handled via tables of marks. (The GAP code shown uses only standard functions lists, such as `List`, `Filtered`, and `ForAll`, and functions for character tables, such as `Irr` and `ScalarProduct`; if you are not familiar with these functions, consult the corresponding sections in the GAP Reference Manual.)

The first example is the Mathieu group M_{11} that has been inspected already in Section 1.1. This group is small enough for the computation of all possible permutation characters, and then filtering out the multiplicity-free ones.

```
gap> tbl:= CharacterTable( "M11" );;
gap> perms:= PermChars( tbl );;
gap> multfree:= Filtered( perms,
>       x -> ForAll( Irr( tbl ), chi -> ScalarProduct( chi, x ) <= 1 ) );
[ Character( CharacterTable( "M11" ), [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ] ),
  Character( CharacterTable( "M11" ), [ 11, 3, 2, 3, 1, 0, 1, 1, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 12, 4, 3, 0, 2, 1, 0, 0, 1, 1 ] ),
  Character( CharacterTable( "M11" ), [ 22, 6, 4, 2, 2, 0, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 55, 7, 1, 3, 0, 1, 1, 1, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 66, 10, 3, 2, 1, 1, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 144, 0, 0, 0, 4, 0, 0, 0, 1, 1 ] ) ]
gap> Length( multfree );
8
```

Comparing this list with the seven faithful multiplicity-free permutation characters of M_{11} shown in Section 1.1, we see that all candidates are in fact permutation characters. Without this information, we have to show, for each candidate, the existence of a subgroup that serves as the point stabilizer.

Additionally, if we are interested in the subgroup information contained in the database (cf. the `subgroup` components of the `info` records in Section 1), we want to relate the point stabilizers to the maximal subgroups of M_{11}.

In the case of the sporadic simple groups and their automorphism groups, we can use the fact that for many of these groups, the character tables of all maximal subgroups and the class fusions of these tables are known. Since each multiplicity-free permutation character of a group is either trivial or induced from a multiplicity-free permutation character of a maximal subgroup, we can thus reduce our problem to the computation of multiplicity-free possible permutation characters of all maximal subgroups. (That this really is a reduction can be read in [BL96].) This approach is implemented in the function `MultFree`.

```
gap> tbl:= CharacterTable( "M11" );
CharacterTable( "M11" )
gap> maxes:= Maxes( tbl );
[ "A6.2_3", "L2(11)", "3^2:Q8.2", "A5.2", "2.S4" ]
gap> name:= maxes[1];;
gap> MultFree( tbl, CharacterTable( name ) );
[ Character( CharacterTable( "M11" ), [ 11, 3, 2, 3, 1, 0, 1, 1, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 22, 6, 4, 2, 2, 0, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ) ]
```

The function `MultFree` computes all multiplicity-free characters of the given character table that are induced from possible permutation characters of the given character table of a subgroup. (Note that these characters need not necessarily be faithful.) If we loop over all classes of maximal subgroups then we get all candidates for M_{11}, together with the information in which maximal subgroup the hypothetical point stabilizer lies.

```
gap> cand:= [];;
gap> for name in maxes do
>      max:= CharacterTable( name );
>      Append( cand, List( MultFree( tbl, max ),
>                     chi -> [ name, Size( tbl ) / Size( max ), chi ] ) );
> od;
gap> cand;
[ [ "A6.2_3", 11, Character( CharacterTable( "M11" ),
        [ 11, 3, 2, 3, 1, 0, 1, 1, 0, 0 ] ) ],
  [ "A6.2_3", 11, Character( CharacterTable( "M11" ),
        [ 22, 6, 4, 2, 2, 0, 0, 0, 0, 0 ] ) ],
  [ "A6.2_3", 11, Character( CharacterTable( "M11" ),
        [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ) ],
  [ "L2(11)", 12, Character( CharacterTable( "M11" ),
        [ 12, 4, 3, 0, 2, 1, 0, 0, 1, 1 ] ) ],
  [ "L2(11)", 12, Character( CharacterTable( "M11" ),
        [ 144, 0, 0, 0, 4, 0, 0, 0, 1, 1 ] ) ],
  [ "3^2:Q8.2", 55, Character( CharacterTable( "M11" ),
        [ 55, 7, 1, 3, 0, 1, 1, 1, 0, 0 ] ) ],
  [ "3^2:Q8.2", 55, Character( CharacterTable( "M11" ),
        [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ) ],
  [ "A5.2", 66, Character( CharacterTable( "M11" ),
        [ 66, 10, 3, 2, 1, 1, 0, 0, 0, 0 ] ) ] ]
gap> Length( cand );  Length( Set( cand, x -> x[3] ) );
8
7
```

We immediately see that the candidates of degrees 11, 12, 55, and 66 are permutation characters, since they are obtained by inducing the trivial characters of the maximal subgroups. The permutation characters of degrees 22 and 144 can be established in two steps. First we note that the group $A_6.2_3$ contains the subgroup A_6 of index 2, and the group $L_2(11)$ contains a class of subgroups of index 12, of isomorphism type $11:5$. Second the possible permutation characters of degrees 2 and 12 of these maximal subgroups of M_{11} are uniquely determined, and inducing these characters to M_{11} yields in fact multiplicity-free characters.

```
gap> max1:= CharacterTable( maxes[1] );;
gap> perms1:= PermChars( max1, [ 2 ] );
[ Character( CharacterTable( "A6.2_3" ), [ 2, 2, 2, 2, 2, 0, 0, 0 ] ) ]
gap> perms1[1]^tbl = cand[2][3];
true
gap> max2:= CharacterTable( maxes[2] );;
gap> perms2:= PermChars( max2, [ 12 ] );
[ Character( CharacterTable( "L2(11)" ), [ 12, 0, 0, 2, 2, 0, 1, 1 ] ) ]
gap> perms2[1]^tbl = cand[5][3];
true
```

The last candidate to deal with is the degree 110 character, which might be induced from a subgroup of $A_6.2_3$ or $3^2:Q_8.2$ or both. Let us first look at the possible permutation characters of degree 10 of $A_6.2_3$.

```
gap> PermChars( max1, [ 10 ] );
```

```
[ Character( CharacterTable( "A6.2_3" ), [ 10, 2, 1, 2, 0, 0, 2, 2 ] ),
  Character( CharacterTable( "A6.2_3" ), [ 10, 2, 1, 2, 0, 2, 0, 0 ] ) ]
gap> OrdersClassRepresentatives( max1 );
[ 1, 2, 3, 4, 5, 4, 8, 8 ]
```

There are two possibilities, and only the first induces the candidate of degree 110. The latter follows from the fact that the nonzero character value of the candidate on classes of element order 8 means that the hypothetical point stabilizer contains elements of order 8, cf. the `Display` call in Section 1.1.

The group $A_6.2_3$ has a unique class of subgroups of index 10, which are the Sylow 3 normalizers, of type $3^2 : Q_8$. Since Q_8 has no elements of order 8, the first candidate is **not** a permutation character.

The remaining subgroup from which the degree 110 character can be induced is $3^2 : Q_8.2$; this group has three index 2 subgroups, and the candidate is in fact induced from the trivial character of one of these subgroups.

```
gap> max3:= CharacterTable( maxes[3] );;
gap> classes:= SizesConjugacyClasses( max3 );;
gap> Filtered( ClassPositionsOfNormalSubgroups( max3 ),
>              x -> Sum( classes{ x } ) = Size( max3 ) / 2 );
[ [ 1, 2, 4, 5, 6 ], [ 1, 2, 3, 4, 5, 7 ], [ 1, 2, 4, 5, 8, 9 ] ]
gap> perms3:= PermChars( max3, [ 2 ] );
[ Character( CharacterTable( "3^2:Q8.2" ), [ 2, 2, 0, 2, 2, 0, 0, 2, 2 ] ),
  Character( CharacterTable( "3^2:Q8.2" ), [ 2, 2, 0, 2, 2, 2, 0, 0, 0 ] ),
  Character( CharacterTable( "3^2:Q8.2" ), [ 2, 2, 2, 2, 2, 0, 2, 0, 0 ] ) ]
gap> induced:= List( perms3, x -> x^tbl );
[ Character( CharacterTable( "M11" ), [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 110, 6, 2, 6, 0, 0, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "M11" ), [ 110, 14, 2, 2, 0, 2, 0, 0, 0, 0 ] ) ]
gap> Position( induced, cand[3][3] );
1
```

Putting these considerations together, we thus get a confirmation of the classification for M_{11}.

As a second example, we look at the group $M_{12}.2$. The database contains 13 characters, and the approach using `MultFree` yields 17 different characters. We are interested in disproving the candidates that are not permutation characters.

```
gap> info:= MultFreePermChars( "M12.2" );;
gap> perms:= Set( List( info, x -> x.character ) );;
gap> Length( info );  Length( perms );
13
13
gap> tbl:= CharacterTable( "M12.2" );;
gap> maxes:= Maxes( tbl );
[ "M12", "L2(11).2", "M12.2M3", "(2^2xA5):2", "D8.(S4x2)", "4^2:D12.2",
  "3^(1+2):D8", "S4xS3", "A5.2" ]
gap> cand:= [];;
gap> for name in maxes do
>      max:= CharacterTable( name );
>      Append( cand, List( MultFree( tbl, max ),
>                      chi -> [ name, Size( tbl ) / Size( max ), chi ] ) );
> od;
gap> Length( cand );  Length( Set( List( cand, x -> x[3] ) ) );
25
17
gap> toexclude:= Set( Filtered( cand, x -> not x[3] in perms ) );
```

```
[ [ "M12", 2, Character( CharacterTable( "M12.2" ),
        [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ],
  [ "M12", 2, Character( CharacterTable( "M12.2" ),
        [ 440, 0, 24, 8, 8, 8, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
         ] ) ],
  [ "M12", 2, Character( CharacterTable( "M12.2" ),
        [ 1320, 0, 8, 6, 0, 8, 0, 0, 2, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
         ] ) ],
  [ "M12", 2, Character( CharacterTable( "M12.2" ),
        [ 1320, 0, 24, 6, 0, 4, 0, 0, 6, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
         ] ) ] ]
```

Clearly the degree 2 character is a permutation character, but as it is not faithful, it is not contained in the database.

The other three characters are all induced from candidates of the maximal subgroup M_{12}, and we may use the same approach for M_{12} in order to find out whether they can be permutation characters.

```
gap> m12:= CharacterTable( "M12" );;
gap> subcand:= [];;
gap> submaxes:= Maxes( m12 );
[ "M11", "M12M2", "A6.2^2", "M12M4", "L2(11)", "3^2.2.S4", "M12M7", "2xS5",
  "M8.S4", "4^2:D12", "A4xS3" ]
gap> for name in submaxes do
>      max:= CharacterTable( name );
>      Append( subcand, MultFree( m12, max ) );
> od;
gap> induced:= List( subcand, x -> x^tbl );;
gap> Intersection( induced, List( toexclude, x -> x[3] ) );
[  ]
```

Thus none of the candidates in the list `toexclude` is a permutation character.

References

[BL96] T. Breuer and K. Lux, *The multiplicity-free permutation characters of the sporadic simple groups and their automorphism groups*, Comm. Algebra **24** (1996), no. 7, 2293–2316. MR 1390375 (97c:20020)

[BP98] T. Breuer and G. Pfeiffer, *Finding possible permutation characters*, J. Symbolic Comput. **26** (1998), no. 3, 343–354. MR 1633876 (99e:20005)

[Bre] T. Breuer, *Permutation Characters in GAP*, `http://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib/doc/ctblpope.pdf`.

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

[CCN$^+$85] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson, *Atlas of finite groups*, Oxford University Press, Eynsham, 1985, Maximal subgroups and ordinary characters for simple groups, With computational assistance from J. G. Thackray. MR 827219 (88g:20025)

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

[Pfe97] G. Pfeiffer, *The subgroups of M_{24}, or how to compute the table of marks of a finite group*, Experiment. Math. **6** (1997), no. 3, 247–270. MR 1481593 (98h:20032)