Computations with the **GAP** Character Table Library

(Version 1.3.11 of CTblLib)

Thomas Breuer

Thomas Breuer Email: `sam@math.rwth-aachen.de`
Homepage: `https://www.math.rwth-aachen.de/~Thomas.Breuer`

Copyright

Contents

Chapter 1

Maintenance Issues for the **GAP** Character Table Library

This chapter collects examples of computations that arose in the context of maintaining the **GAP** Character Table Library. The sections have been added when the issues in question arose; the dates of the additions are shown in the section titles.

1.1 Disproving Possible Character Tables (November 2006)

I do not know a necessary and sufficient criterion for checking whether a given matrix together with a list of power maps describes the character table of a finite group. Examples of *pseudo character tables* (tables which satisfy certain necessary conditions but for which actually no group exists) have been given in [Gag86]. Another such example is described in Section 2.4.17. The tables in the **GAP** Character Table Library satisfy the usual tests. However, there are table candidates for which these tests are not good enough. Another question would be whether a given character table belongs to the group for which it is claimed to belong, see Section 1.1.4 for an example.

1.1.1 A Perfect Pseudo Character Table (November 2006)

(This example arose from a discussion with Jack Schmidt.)

Up to version 1.1.3 of the **GAP** Character Table Library, the table with identifier `"P41/G1/L1/V4/ext2"` was not correct. The problem occurs already in the microfiches that are attached to [HP89].

In the following, we show that this table is not the character table of a finite group, using the **GAP** library of perfect groups. Currently we do not know how to prove this inconsistency alone from the table.

We start with the construction of the inconsistent table; apart from a little editing, the following input equals the data formerly stored in the file `data/ctoholpl.tbl` of the **GAP** Character Table Library.

```
———————————————————————— Example ————————————————————————
  gap> tbl:= rec(
  >    Identifier:= "P41/G1/L1/V4/ext2",
  >    InfoText:= Concatenation( [
  >      "origin: Hanrath library,\n",
  >      "structure is 2^7.L2(8),\n",
```

```
>        "characters sorted with permutation (12,14,15,13)(19,20)" ] ),
>    UnderlyingCharacteristic:= 0,
>    SizesCentralizers:= [64512,1024,1024,64512,64,64,64,64,128,128,64,
>        64,128,128,18,18,14,14,14,14,14,14,18,18,18,18,18,18],
>    ComputedPowerMaps:= [,[1,1,1,1,2,3,3,2,3,2,2,1,3,2,16,16,20,20,22,
>        22,18,18,26,26,27,27,23,23],[1,2,3,4,5,6,7,8,9,10,11,12,13,14,4,
>        1,21,22,17,18,19,20,16,15,15,16,16,15],,,,[1,2,3,4,5,6,7,8,9,10,
>        11,12,13,14,15,16,4,1,4,1,4,1,26,25,28,27,23,24]],
>    Irr:= 0,
>    AutomorphismsOfTable:= Group( [(23,26,27)(24,25,28),(9,13)(10,14),
>        (17,19,21)(18,20,22)] ),
>    ConstructionInfoCharacterTable:= ["ConstructClifford",[[[1,2,3,4,
>        5,6,7,8,9],[1,7,8,3,9,2],[1,4,5,6,2],[1,2,2,2,2,2,2,2,2]],
>        [["L2(8)"],["Dihedral",18],["Dihedral",14],["2^3"]],[[[1,2,3,4],
>        [1,1,1,1],["elab",4,25]],[[1,2,3,4,4,4,4,4,4,4],[2,6,5,2,3,4,5,
>        6,7,8],["elab",10,17]],[[1,2],[3,4],[[1,1],[-1,1]]],[[1,3],[4,
>        2],[[1,1],[-1,1]]],[[1,3],[5,3],[[1,1],[-1,1]]],[[1,3],[6,4],
>        [[1,1],[-1,1]]],[[1,2],[7,2],[[1,1],[1,-1]]],[[1,2],[8,3],[[1,
>        1],[-1,1]]],[[1,2],[9,5],[[1,1],[1,-1]]]]]],
>    );;
gap> ConstructClifford( tbl, tbl.ConstructionInfoCharacterTable[2] );
gap> ConvertToLibraryCharacterTableNC( tbl );;
```

Suppose that there is a group *G*, say, with this table. Then *G* is perfect since the table has only one linear character.

```
────────────────────── Example ──────────────────────
gap> Length( LinearCharacters( tbl ) );
1
gap> IsPerfectCharacterTable( tbl );
true
```

The table satisfies the orthogonality relations, the structure constants are nonnegative integers, and symmetrizations of the irreducibles decompose into the irreducibles, with nonnegative integral coefficients.

```
────────────────────── Example ──────────────────────
gap> IsInternallyConsistent( tbl );
true
gap> irr:= Irr( tbl );;
gap> test:= Concatenation( List( [ 2 .. 7 ],
>             n -> Symmetrizations( tbl, irr, n ) ) );;
gap> Append( test, Set( Tensored( irr, irr ) ) );
gap> fail in Decomposition( irr, test, "nonnegative" );
false
gap> if ForAny( Tuples( [ 1 .. NrConjugacyClasses( tbl ) ], 3 ),
>      t -> not ClassMultiplicationCoefficient( tbl, t[1], t[2], t[3] )
>             in NonnegativeIntegers ) then
>      Error( "contradiction" );
> fi;
```

The GAP Library of Perfect Groups contains representatives of the four isomorphism types of perfect groups of order $|G| = 64512$.

```
──────── Example ────────
gap> n:= Size( tbl );
64512
gap> NumberPerfectGroups( n );
4
gap> grps:= List( [ 1 .. 4 ], i -> PerfectGroup( IsPermGroup, n, i ) );
[ L2(8) 2^6 E 2^1, L2(8) N 2^6 E 2^1 I, L2(8) N 2^6 E 2^1 II,
  L2(8) N 2^6 E 2^1 III ]
```

If we believe that the classification of perfect groups of order $|G|$ is correct then all we have to do is to show that none of the character tables of these four groups is equivalent to the given table.

```
──────── Example ────────
gap> tbls:= List( grps, CharacterTable );;
gap> List( tbls,
>          x -> TransformingPermutationsCharacterTables( x, tbl ) );
[ fail, fail, fail, fail ]
```

In fact, already the matrices of irreducible characters of the four groups do not fit to the given table.

```
──────── Example ────────
gap> List( tbls,
>          t -> TransformingPermutations( Irr( t ), Irr( tbl ) ) );
[ fail, fail, fail, fail ]
```

Let us look closer at the tables in question. Each character table of a perfect group of order $64\,512$ has exactly one irreducible character of degree 63 that takes exactly the values -1, 0, 7, and 63; moreover, the value 7 occurs in exactly two classes.

```
──────── Example ────────
gap> testchars:= List( tbls,
>    t -> Filtered( Irr( t ),
>         x -> x[1] = 63 and Set( x ) = [ -1, 0, 7, 63 ] ) );;
gap> List( testchars, Length );
[ 1, 1, 1, 1 ]
gap> List( testchars, l -> Number( l[1], x -> x = 7 ) );
[ 2, 2, 2, 2 ]
```

(Another way to state this is that in each of the four tables t in question, there are ten preimage classes of the involution class in the simple factor group $L_2(8)$, there are eight preimage classes of this class in the factor group $2^6.L_2(8)$, and that the unique class in which an irreducible degree 63 character of this factor group takes the value 7 splits in t.)

In the erroneous table, however, there is only one class with the value 7 in this character.

```
──────── Example ────────
gap> testchars:= List( [ tbl ],
>    t -> Filtered( Irr( t ),
>         x -> x[1] = 63 and Set( x ) = [ -1, 0, 7, 63 ] ) );;
gap> List( testchars, Length );
[ 1 ]
gap> List( testchars, l -> Number( l[1], x -> x = 7 ) );
[ 1 ]
```

This property can be checked easily for the displayed table stored in fiche 2, row 4, column 7 of [HP89], with the name `6L1<>Z^7<>L2(8); V4; MOD 2`, and it turns out that this table is not correct.

Note that these microfiches contain *two* tables of order 64512, and there were *three* tables of groups of that order in the GAP Character Table Library that contain `origin:  Hanrath library` in their `InfoText` (**Reference: InfoText**) value. Besides the incorrect table, these library tables are the character tables of the groups `PerfectGroup( 64512, 1 )` and `PerfectGroup( 64512, 3 )`, respectively. (The matrices of irreducible characters of these tables are equivalent.)

```
─────────────── Example ───────────────
gap> Filtered( [ 1 .. 4 ], i ->
>        TransformingPermutationsCharacterTables( tbls[i],
>           CharacterTable( "P41/G1/L1/V1/ext2" ) ) <> fail );
[ 1 ]
gap> Filtered( [ 1 .. 4 ], i ->
>        TransformingPermutationsCharacterTables( tbls[i],
>           CharacterTable( "P41/G1/L1/V2/ext2" ) ) <> fail );
[ 3 ]
gap> TransformingPermutations( Irr( tbls[1] ), Irr( tbls[3] ) ) <> fail;
true
```

Since version 1.2 of the GAP Character Table Library, the character table with the `Identifier` (**Reference: Identifier for tables of marks**) value `"P41/G1/L1/V4/ext2"` corresponds to the group `PerfectGroup( 64512, 4 )`. The choice of this group was somewhat arbitrary since the vector system V4 seems to be not defined in [HP89]; anyhow, this group and the remaining perfect group, `PerfectGroup( 64512, 2 )`, have equivalent matrices of irreducibles.

```
─────────────── Example ───────────────
gap> Filtered( [ 1 .. 4 ], i ->
>        TransformingPermutationsCharacterTables( tbls[i],
>           CharacterTable( "P41/G1/L1/V4/ext2" ) ) <> fail );
[ 4 ]
gap> TransformingPermutations( Irr( tbls[2] ), Irr( tbls[4] ) ) <> fail;
true
```

1.1.2 An Error in the Character Table of $E_6(2)$ (March 2016)

In March 2016, Bill Unger computed the character table of the simple group $E_6(2)$ with Magma (see [CP96]) and compared it with the table that was contained in the GAP Character Table Library since 2000. It turned out that the two tables did not coincide.

The differences concern irrational character values on classes of element order 91 and power map values on these classes. (The character values and power maps fit to each other in both tables; thus it may be that the assumption of a wrong power has implied the wrong character values, or vice versa.) Specifically, the 11th power map in the GAP table fixed all elements of order 91. Using the smallest matrix representation of $E_6(2)$ over the field with two elements, one can easily find an element g of order 91, and show that the characteristic polynomials of g and g^{11} differ. Hence these two elements cannot be conjugate in $E_6(2)$. In other words, the GAP table was wrong.

```
─────────────── Example ───────────────
gap> g:= AtlasGroup( "E6(2)" );;
gap> repeat x:= PseudoRandom( g ); until Order( x ) = 91;
```

```
gap> CharacteristicPolynomial( x ) = CharacteristicPolynomial( x^11 );
false
```

The wrong GAP table has been corrected in version 1.3.0 of the GAP Character Table Library.

```
———————————————— Example ————————————————
gap> t:= CharacterTable( "E6(2)" );;
gap> ord91:= Positions( OrdersClassRepresentatives( t ), 91 );
[ 163, 164, 165, 166, 167, 168 ]
gap> PowerMap( t, 11 ){ ord91 };
[ 167, 168, 163, 164, 165, 166 ]
```

1.1.3 An Error in a Power Map of the Character Table of $2.F_4(2).2$ (November 2015)

As a part of the computations for [BMO17], the character table of the group $2.F_4(2).2$ was computed automatically from a representation of the group, using Magma (see [CP96]). It turned out that the 2-nd power map that had been stored on the library character table of $2.F_4(2).2$ had been wrong.

In fact, this was the one and only case of a power map for an Atlas group which was not determined by the character table, and the `InfoText` (**Reference: InfoText**) value of the character table had mentioned the two alternatives.

Note that the ambiguity is not present in the table of the factor group $F_4(2).2$, and only four faithful irreducible characters of $2.F_4(2).2$ distinguish the four relevant conjugacy classes.

```
———————————————— Example ————————————————
gap> t:= CharacterTable( "2.F4(2).2" );;
gap> f:= CharacterTable( "F4(2).2" );;
gap> map:= PowerMap( t, 2 );
[ 1, 1, 1, 1, 1, 1, 1, 1, 9, 9, 11, 11, 3, 3, 3, 5, 5, 5, 3, 6, 6, 5,
  5, 7, 7, 5, 8, 7, 29, 29, 9, 9, 9, 9, 11, 11, 9, 9, 9, 9, 11, 11,
  43, 43, 20, 20, 20, 14, 14, 13, 13, 20, 21, 24, 28, 28, 57, 57, 29,
  29, 29, 29, 33, 33, 35, 37, 37, 37, 37, 33, 33, 37, 37, 35, 41, 41,
  42, 42, 79, 79, 43, 43, 83, 83, 45, 45, 47, 47, 53, 53, 91, 91, 57,
  57, 61, 61, 61, 98, 98, 70, 70, 63, 63, 81, 81, 83, 83, 1, 6, 7,
  11, 16, 17, 24, 24, 21, 27, 27, 25, 26, 29, 41, 53, 53, 53, 46, 56,
  56, 56, 56, 62, 75, 75, 78, 78, 77, 77, 79, 79, 86, 86, 85, 85, 88,
  88, 88, 88, 95, 95, 96, 96 ]
gap> PositionSublist( map, [ 86, 86, 85, 85 ] );
140
gap> OrdersClassRepresentatives( t ){ [ 140 .. 143 ] };
[ 32, 32, 32, 32 ]
gap> SizesCentralizers( t ){ [ 140 .. 143 ] };
[ 64, 64, 64, 64 ]
gap> GetFusionMap( t, f ){ [ 140 ..143 ] };
[ 86, 86, 87, 87 ]
gap> PowerMap( f, 2 ){ [ 86, 87 ] };
[ 50, 50 ]
gap> pos:= PositionsProperty( Irr( t ),
>    x -> x[1] <> x[2] and Length( Set( x{ [ 140 .. 143 ] } ) ) > 1 );
[ 144, 145, 146, 147 ]
gap> List( pos, i -> Irr(t)[i]{ [ 140 .. 143 ] } );
[ [ 2*E(16)-2*E(16)^7, -2*E(16)+2*E(16)^7, 2*E(16)^3-2*E(16)^5,
      -2*E(16)^3+2*E(16)^5 ],
```

```
  [ -2*E(16)+2*E(16)^7, 2*E(16)-2*E(16)^7, -2*E(16)^3+2*E(16)^5,
      2*E(16)^3-2*E(16)^5 ],
  [ -2*E(16)^3+2*E(16)^5, 2*E(16)^3-2*E(16)^5, 2*E(16)-2*E(16)^7,
      -2*E(16)+2*E(16)^7 ],
  [ 2*E(16)^3-2*E(16)^5, -2*E(16)^3+2*E(16)^5, -2*E(16)+2*E(16)^7,
      2*E(16)-2*E(16)^7 ] ]
```

I had not found a suitable subgroup of $2.F_4(2).2$ whose character table could be used to decide the question which of the two alternatives is the correct one.

1.1.4 A Character Table with a Wrong Name (May 2017)

(This example is much older.)

The character table that is shown in [Ost86, p. 126 f.] is claimed to be the table of a Sylow 2 subgroup P of the sporadic simple Lyons group Ly. This table had been contained in the character table library of the CAS system (see [NPP84]), which was one of the predecessors of GAP.

It is easy to see that no subgroup of Ly can have this character table. Namely, the group of that table contains elements of order eight with centralizer order 2^6, and this does not occur in Ly.

```
──────────────────── Example ────────────────────
  gap> tbl:= CharacterTable( "Ly" );;
  gap> orders:= OrdersClassRepresentatives( tbl );;
  gap> order8:= Filtered( [ 1 .. Length( orders ) ], x -> orders[x] = 8 );
  [ 12, 13 ]
  gap> SizesCentralizers( tbl ){ order8 } / 2^6;
  [ 15/2, 3/2 ]
```

The table of P has been computed in [Bre91] with character theoretic methods. Nowadays it would be no problem to take a permutation representation of Ly, to compute its Sylow 2 subgroup, and use this group to compute its character table. However, the task is even easier if we assume that Ly has a subgroup of the structure $3.McL.2$. This subgroup is of odd index, hence it contains a conjugate of P. Clearly the Sylow 2 subgroups in the factor group $McL.2$ are isomorphic with P. Thus we can start with a rather small permutation representation.

```
──────────────────── Example ────────────────────
  gap> g:= AtlasGroup( "McL.2" );;
  gap> NrMovedPoints( g );
  275
  gap> syl:= SylowSubgroup( g, 2 );;
  gap> pc:= Image( IsomorphismPcGroup( syl ) );;
  gap> t:= CharacterTable( pc );;
```

The character table coincides with the one which is stored in the Character Table Library.

```
──────────────────── Example ────────────────────
  gap> IsRecord( TransformingPermutationsCharacterTables( t,
  >                 CharacterTable( "LyN2" ) ) );
  true
```

1.2 Some finite factor groups of perfect space groups (February 2014)

If one wants to find a group to which a given character table from the GAP Character Table Library belongs, one can try the function `GroupInfoForCharacterTable` (**CTblLib: GroupInfoForCharacterTable**). For a long time, this was not successful in the case of 16 character tables that had been computed by W. Hanrath (see Section "Ordinary and Brauer Tables in the GAP Character Table Library" in the CTblLib manual).

Using the information from [HP89], it is straightforward to construct such groups as factor groups of infinite groups. Since version 1.3.0 of the CTblLib package, calling `GroupInfoForCharacterTable` (**CTblLib: GroupInfoForCharacterTable**) for the 16 library tables in question yields nonempty lists and thus allows one to access the results of these constructions, via the function `CTblLib.FactorGroupOfPerfectSpaceGroup`. This is an undocumented auxiliary function that becomes available automatically when `GroupInfoForCharacterTable` (**CTblLib: GroupInfoForCharacterTable**) has been called for the first time.

```
───────────────────────────── Example ─────────────────────────────
  gap> GroupInfoForCharacterTable( "A5" );;
  gap> IsBound( CTblLib.FactorGroupOfPerfectSpaceGroup );
  true
```

Below we list the 16 group constructions. In each case, an epimorphism from the space group in question is defined by mapping the generators returned by by the function `generatorsOfPerfectSpaceGroup` defined below to the generators stored in the attribute `GeneratorsOfGroup` (**Reference: GeneratorsOfGroup**) of the group returned by `CTblLib.FactorGroupOfPerfectSpaceGroup`.

1.2.1 Constructing the space groups in question

In [HP89], a space group S is described as a subgroup $\{M(g,t); g \in P, t \in T\}$ of $\mathrm{GL}(d+1, \mathbb{Z})$, where

$$M(g,t) = \begin{bmatrix} g & 0 \\ V(g)+t & 1 \end{bmatrix},$$

the *point group* P of S is a finite subgroup of $\mathrm{GL}(d, \mathbb{Z})$, the *translation lattice* T of S is a sublattice of $\mathbb{Z}^d$, and the *vector system* V of S is a map from P to $\mathbb{Z}^d$. Note that V maps the identity matrix $I \in \mathrm{GL}(d, \mathbb{Z})$ to the zero vector, and $M(T) := \{M(I,t); t \in T\}$ is a normal subgroup of S that is isomorphic with T. More generally, $M(nT)$ is a normal subgroup of S, for any positive integer n.

Specifically, P is given by generators $g_1, g_2, \ldots, g_k$, T is given by a $\mathbb{Z}$-basis $B = \{b_1, b_2, \ldots, b_d\}$ of T, and V is given by the vectors $V(g_1), V(g_2), \ldots, V(g_k)$.

In the examples below, the matrix representation of P is irreducible, so we need just the following $k+1$ elements to generate S:

$$\begin{bmatrix} g_1 & 0 \\ V(g_1) & 1 \end{bmatrix}, \begin{bmatrix} g_2 & 0 \\ V(g_2) & 1 \end{bmatrix}, \ldots, \begin{bmatrix} g_k & 0 \\ V(g_k) & 1 \end{bmatrix}, \begin{bmatrix} I & 0 \\ b_1 & 1 \end{bmatrix}.$$

These generators are returned by the function `generatorsOfPerfectSpaceGroup`, when the inputs are $[g_1, g_2, \ldots, g_k]$, $[V(g_1), V(g_2), \ldots, V(g_k)]$, and b_1.

```
                                _____ Example _____
  gap> generatorsOfPerfectSpaceGroup:= function( Pgens, V, t )
  >      local d, result, i, m;
  >      d:= Length( Pgens[1] );
  >      result:= [];
  >      for i in [ 1 .. Length( Pgens ) ] do
  >        m:= IdentityMat( d+1 );
  >        m{ [ 1 .. d ] }{ [ 1 .. d ] }:= Pgens[i];
  >        m[ d+1 ]{ [ 1 .. d ] }:= V[i];
  >        result[i]:= m;
  >      od;
  >      m:= IdentityMat( d+1 );
  >      m[ d+1 ]{ [ 1 .. d ] }:= t;
  >      Add( result, m );
  >      return result;
  > end;;
```

1.2.2 Constructing the factor groups in question

The space group S acts on $\mathbb{Z}^d$, via $v \cdot M(g,t) = vg + V(g) + t$. A (not necessarily faithful) representation of $S/M(nT)$ can be obtained from the corresponding action of S on $\mathbb{Z}^d/(n\mathbb{Z}^d)$, that is, by reducing the vectors modulo n. For the GAP computations, we work instead with vectors of length $d+1$, extending each vector in $\mathbb{Z}^d$ by 1 in the last position, and acting on these vectors by right multiplicaton with elements of S. Multiplication followed by reduction modulo n is implemented by the action function returned by `multiplicationModulo` when this is called with argument n.

```
                                _____ Example _____
  gap> multiplicationModulo:= n -> function( v, g )
  >          return List( v * g, x -> x mod n ); end;;
```

In some of the examples, the representation of P given in [HP89] is the action on the factor of a permutation module modulo its trivial submodule. For that, we provide the function `deletedPermutationMat`, cf. [HP89, p. 269].

```
                                _____ Example _____
  gap> deletedPermutationMat:= function( pi, n )
  >      local mat, j, i;
  >      mat:= PermutationMat( pi, n );
  >      mat:= mat{ [ 1 .. n-1 ] }{ [ 1 .. n-1 ] };
  >      j:= n ^ pi;
  >      if j <> n then
  >        for i in [ 1 .. n-1 ] do
  >          mat[i][j]:= -1;
  >        od;
  >      fi;
  >      return mat;
  > end;;
```

After constructing permutation generators for the example groups, we verify that the groups fit to the character tables from the GAP Character Table Library and to the permutation generators stored for the construction of the group via `CTblLib.FactorGroupOfPerfectSpaceGroup`.

```
                              Example
gap> verifyFactorGroup:= function( gens, id )
>       local sm, act, stored, hom;
>       sm:= SmallerDegreePermutationRepresentation( Group( gens ) );
>       gens:= List( gens, x -> x^sm );
>       act:= Images( sm );
>       if not IsRecord( TransformingPermutationsCharacterTables(
>                            CharacterTable( act ),
>                            CharacterTable( id ) ) ) then
>         return "wrong character table";
>       fi;
>       GroupInfoForCharacterTable( id );
>       stored:= CTblLib.FactorGroupOfPerfectSpaceGroup( id );
>       hom:= GroupHomomorphismByImages( stored, act,
>               GeneratorsOfGroup( stored ), gens );
>       if hom = fail or not IsBijective( hom ) then
>         return "wrong group";
>       fi;
>       return true;
> end;;
```

1.2.3   Examples with point group A_5

There are two examples with $d = 5$. The generators of the point group are as follows (see [HP89, p. 272]).

```
                              Example
gap> a:= deletedPermutationMat( (1,3)(2,4), 6 );;
gap> b:= deletedPermutationMat( (1,2,3)(4,5,6), 6 );;
```

In both cases, the vector system is V_2.

```
                              Example
gap> v:= [ [ 2, 2, 0, 0, 1 ], 0 * b[1] ];;
```

In the first example, the translation lattice is the sublattice $L = 2L_1$ of the full lattice $L_1 = \mathbb{Z}^d$.

```
                              Example
gap> t:= [ 2, 0, 0, 0, 0 ];;
```

The library character table with identifier "P1/G2/L1/V2/ext4" belongs to the factor group of S modulo the normal subgroup $M(4L)$, so we compute the action on an orbit modulo 8.

```
                              Example
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 8 );;
gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 1 ], fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P1/G2/L1/V2/ext4" );
true
```

In the second example, the translation lattice is the sublattice $2L_2$ of $\mathbb{Z}^d$ where L_2 has the following basis.

```
──────────────────── Example ────────────────────
gap> bas:= [ [-1,-1, 1, 1, 1 ],
>            [-1, 1,-1, 1, 1 ],
>            [ 1, 1, 1,-1,-1 ],
>            [ 1, 1,-1,-1, 1 ],
>            [-1, 1, 1,-1, 1 ] ];;
```

For the sake of simplicity, we rewrite the action of the point group to one on L_2, and we adjust also the vector system.

```
──────────────────── Example ────────────────────
gap> B:= Basis( Rationals^Length( bas ), bas );;
gap> abas:= List( bas, x -> Coefficients( B, x * a ) );;
gap> bbas:= List( bas, x -> Coefficients( B, x * b ) );;
gap> vbas:= List( v, x -> Coefficients( B, x ) );
[ [ 3/2, 1, 2, 3/2, -1 ], [ 0, 0, 0, 0, 0 ] ]
```

In order to work with integral matrices (which is necessary because `multiplicationModulo` uses GAP's `mod` operator), we double both the vector system and the translation lattice.

```
──────────────────── Example ────────────────────
gap> vbas:= vbas * 2;
[ [ 3, 2, 4, 3, -2 ], [ 0, 0, 0, 0, 0 ] ]
gap> t:= 2 * t;
[ 4, 0, 0, 0, 0 ]
```

The library character table with identifier `"P1/G2/L2/V2/ext4"` belongs to the factor group of S modulo the normal subgroup $M(8L_2)$; since we have doubled the lattice, we compute the action on an orbit modulo 16.

```
──────────────────── Example ────────────────────
gap> sgens:= generatorsOfPerfectSpaceGroup( [ abas, bbas ], vbas, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 16 );;
gap> orb:= Orbit( g, [ 0, 0, 0, 0, 0, 1 ], fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P1/G2/L2/V2/ext4" );
true
```

1.2.4 Examples with point group $L_3(2)$

There are three examples with $d = 6$ and one example with $d = 8$. The generators of the point group for the first three examples are as follows (see [HP89, p. 290]).

```
──────────────────── Example ────────────────────
gap> a:= [ [ 0, 1, 0, 1, 0, 0 ],
>          [ 1, 0, 1, 1, 1, 1 ],
>          [-1,-1,-1,-1, 0, 0 ],
>          [ 0, 0,-1,-1,-1,-1 ],
>          [ 1, 1, 1, 1, 0, 1 ],
>          [ 0, 0, 1, 0, 1, 0 ] ];;
```

```
gap> b:= [ [-1, 0, 0, 0, 0,-1 ],
>          [ 0, 0,-1, 0,-1, 0 ],
>          [ 1, 1, 1, 1, 1, 1 ],
>          [ 0, 0, 1, 0, 0, 0 ],
>          [-1,-1,-1, 0, 0, 0 ],
>          [ 1, 0, 0, 0, 0, 0 ] ];;
```

The first vector system is the trivial vector system V_1 (that is, the space group S is a split extension of the point group and the translation lattice), and the translation lattice is the full lattice $L_1 = \mathbb{Z}^d$.

The library character table with identifier "P11/G1/L1/V1/ext4" belongs to the factor group of S modulo the normal subgroup $M(4L_1)$, so we compute the action on an orbit modulo 4.

```
──────────── Example ────────────
gap> v:= List( [ 1, 2 ], i -> 0 * a[1] );;
gap> t:= [ 1, 0, 0, 0, 0, 0 ];;
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 4 );;
gap> seed:= [ 1, 0, 0, 0, 0, 0, 1 ];;
gap> orb:= Orbit( g, seed, fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P11/G1/L1/V1/ext4" );
true
```

The second vector system is V_2, and the translation lattice is $2L_1$.

The library character table with identifier "P11/G1/L1/V2/ext4" belongs to the factor group of S modulo the normal subgroup $M(8L_1)$, so we compute the action on an orbit modulo 8.

```
──────────── Example ────────────
gap> v:= [ [ 1, 0, 1, 0, 0, 0 ], 0 * a[1] ];;
gap> t:= [ 2, 0, 0, 0, 0, 0 ];;
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 8 );;
gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 0, 1 ], fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P11/G1/L1/V2/ext4" );
true
```

The third vector system is V_3, and the translation lattice is $2L_1$.

The library character table with identifier "P11/G1/L1/V3/ext4" belongs to the factor group of S modulo the normal subgroup $M(8L_1)$, so we compute the action on an orbit modulo 8.

```
──────────── Example ────────────
gap> v:= [ [ 0, 1, 0, 0, 1, 0 ], 0 * a[1] ];;
gap> t:= [ 2, 0, 0, 0, 0, 0 ];;
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 8 );;
gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 0, 1 ], fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P11/G1/L1/V3/ext4" );
true
```

The generators of the point group for the fourth example are as follows (see [HP89, p. 293]).

```
                             ─ Example ─
 gap> a:= [ [ 1, 0, 0, 1, 0,-1, 0, 1 ],
 >          [ 0,-1, 1, 0,-1, 0, 0, 0 ],
 >          [ 1, 0, 0, 1, 0,-1, 0, 0 ],
 >          [ 0,-1, 0,-1, 0, 1, 1,-1 ],
 >          [ 1, 0,-1, 1, 1,-1, 0, 0 ],
 >          [ 1,-1,-1, 0, 0, 0, 1, 0 ],
 >          [ 0,-1, 1, 0,-1, 1, 0,-1 ],
 >          [ 1, 0,-1, 0, 0, 0, 0, 0 ] ];;
 gap> b:= [ [ 1, 0,-2, 0, 1,-1, 1, 0 ],
 >          [ 0,-1, 0, 0, 0, 0, 1,-1 ],
 >          [ 1, 0,-1, 0, 1,-1, 0, 0 ],
 >          [-1,-1, 1,-1,-1, 2, 0,-1 ],
 >          [ 0, 0, 0,-1, 0, 0, 0, 0 ],
 >          [ 0,-1, 0,-1,-1, 1, 1,-1 ],
 >          [ 1,-1, 0, 0, 0, 0, 0, 0 ],
 >          [ 1, 0, 0, 0, 0, 0, 0, 0 ] ];;
```

The vector system is the trivial vector system V_1, and the translation lattice is the full lattice $L_1 = \mathbb{Z}^d$.

The library character table with identifier `"P11/G4/L1/V1/ext3"` belongs to the factor group of S modulo the normal subgroup $M(3L_1)$, so we compute the action on an orbit modulo 3.

```
                             ─ Example ─
 gap> v:= List( [ 1, 2 ], i -> 0 * a[1] );;
 gap> t:= [ 1, 0, 0, 0, 0, 0, 0, 0 ];;
 gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
 gap> g:= Group( sgens );;
 gap> fun:= multiplicationModulo( 3 );;
 gap> seed:= [ 1, 0, 0, 0, 0, 0, 0, 0, 1 ];;
 gap> orb:= Orbit( g, seed, fun );;
 gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
 gap> verifyFactorGroup( permgens, "P11/G4/L1/V1/ext3" );
 true
```

1.2.5 Example with point group SL$_2$(7)

There is one example with $d = 8$. The generators of the point group are as follows (see [HP89, p. 295]).

```
                             ─ Example ─
 gap> a:= KroneckerProduct( IdentityMat( 4 ), [ [ 0, 1 ], [ -1, 0 ] ] );;
 gap> b:= [ [ 0,-1, 0, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 1, 0, 0, 0, 0, 0 ],
 >          [-1, 0, 0, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0,-1, 0 ],
 >          [ 0, 0, 0,-1, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 1, 0, 0 ],
 >          [ 0, 0, 0, 0, 1, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 0, 1 ] ];;
```

The vector system is the trivial vector system V_1, and the translation lattice is the sublattice L_2 of $\mathbb{Z}^d$ that has the following basis, which is called $B(2,8)$ in [HP89, p. 269].

```
─────────────────── Example ───────────────────
gap> bas:= [ [ 1, 1, 0, 0, 0, 0, 0, 0 ],
>            [ 0, 1, 1, 0, 0, 0, 0, 0 ],
>            [ 0, 0, 1, 1, 0, 0, 0, 0 ],
>            [ 0, 0, 0, 1, 1, 0, 0, 0 ],
>            [ 0, 0, 0, 0, 1, 1, 0, 0 ],
>            [ 0, 0, 0, 0, 0, 1, 1, 0 ],
>            [ 0, 0, 0, 0, 0, 0, 1, 1 ],
>            [ 0, 0, 0, 0, 0, 0,-1, 1 ] ];;
```

For the sake of simplicity, we rewrite the action to one on L_2.

```
─────────────────── Example ───────────────────
gap> B:= Basis( Rationals^Length( bas ), bas );;
gap> abas:= List( bas, x -> Coefficients( B, x * a ) );;
gap> bbas:= List( bas, x -> Coefficients( B, x * b ) );;
```

The library character table with identifier `"P12/G1/L2/V1/ext2"` belongs to the factor group of S modulo the normal subgroup $M(2L_2)$. The action on an orbit modulo 2 is not faithful, its kernel contains the centre of $\mathrm{SL}(2,7)$. We can compute a faithful representation by acting on pairs: One entry is the usual vector and the other entry carries the action of the point group.

```
─────────────────── Example ───────────────────
gap> v:= List( [ 1, 2 ], i -> 0 * a[1] );;
gap> t:= [ 1, 0, 0, 0, 0, 0, 0, 0 ];;
gap> sgens:= generatorsOfPerfectSpaceGroup( [ abas, bbas ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 2 );;
gap> funpairs:= function( pair, g )
>    return [ fun( pair[1], g ), pair[2] * g ];
>    end;;
gap> seed:= [ [ 1, 0, 0, 0, 0, 0, 0, 0, 1 ],
>             [ 1, 0, 0, 0, 0, 0, 0, 0, 0 ] ];;
gap> orb:= Orbit( g, seed, funpairs );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, funpairs ) );;
gap> verifyFactorGroup( permgens, "P12/G1/L2/V1/ext2" );
true
```

1.2.6   Example with point group $2^3.L_3(2)$

There is one example with $d = 7$. The generators of the point group are as follows (see [HP89, p. 297]).

```
─────────────────── Example ───────────────────
gap> a:= PermutationMat( (2,4)(5,7), 7 );;
gap> b:= PermutationMat( (1,3,2)(4,6,5), 7 );;
gap> c:= DiagonalMat( [ -1, -1, 1, 1, -1, -1, 1 ] );;
```

The vector system is the trivial vector system V_1, and the translation lattice is the sublattice L_2 of $\mathbb{Z}^d$ that has the following basis, which is called $B(2,7)$ in [HP89, p. 269].

```
  ─────────────────────────── Example ───────────────────────────
 gap> bas:= [ [ 1, 1, 0, 0, 0, 0, 0 ],
 >            [ 0, 1, 1, 0, 0, 0, 0 ],
 >            [ 0, 0, 1, 1, 0, 0, 0 ],
 >            [ 0, 0, 0, 1, 1, 0, 0 ],
 >            [ 0, 0, 0, 0, 1, 1, 0 ],
 >            [ 0, 0, 0, 0, 0, 1, 1 ],
 >            [ 0, 0, 0, 0, 0,-1, 1 ] ];;
```

For the sake of simplicity, we rewrite the action to one on L_2.

```
  ─────────────────────────── Example ───────────────────────────
 gap> B:= Basis( Rationals^Length( bas ), bas );;
 gap> abas:= List( bas, x -> Coefficients( B, x * a ) );;
 gap> bbas:= List( bas, x -> Coefficients( B, x * b ) );;
 gap> cbas:= List( bas, x -> Coefficients( B, x * c ) );;
```

The library character table with identifier "P13/G1/L2/V1/ext2" belongs to the factor group of S modulo the normal subgroup $M(2L_2)$, so we compute the action on an orbit modulo 2.

```
  ─────────────────────────── Example ───────────────────────────
 gap> v:= List( [ 1 .. 3 ], i -> 0 * a[1] );;
 gap> t:= [ 1, 0, 0, 0, 0, 0, 0 ];;
 gap> sgens:= generatorsOfPerfectSpaceGroup( [ abas,bbas,cbas ], v, t );;
 gap> g:= Group( sgens );;
 gap> fun:= multiplicationModulo( 2 );;
 gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 0, 0, 1 ], fun );;
 gap> act:= Action( g, orb, fun );;
 gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
 gap> verifyFactorGroup( permgens, "P13/G1/L2/V1/ext2" );
 true
```

1.2.7 Examples with point group A_6

There are two examples with $d = 10$. In both cases, the generators of the point group are as follows (see [HP89, p. 307]).

```
  ─────────────────────────── Example ───────────────────────────
 gap> b:= [ [ 0,-1, 0, 0, 0, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 0,-1, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 ],
 >          [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 ] ];;
 gap> c:= [ [ 0, 0, 0, 0, 0, 0, 0,-1, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 0,-1, 1,-1 ],
 >          [ 0, 0, 0, 0,-1, 1, 0,-1, 0, 0 ],
 >          [ 0,-1, 1, 0, 0, 0, 0,-1, 0, 0 ],
 >          [ 0, 0, 0, 0, 0, 0, 0, 0, 0,-1 ],
```

```
>            [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 ],
>            [ 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 ],
>            [ 0, 0, 0, 0, 0, 1,-1, 0, 0, 1 ],
>            [ 0, 0, 1,-1, 0, 0, 0, 0, 0, 1 ],
>            [-1, 0, 1, 0, 0,-1, 0, 0, 0, 0 ] ];;
```

In both examples, the vector system is the trivial vector system V_1, and the translation lattices are the lattices L_2 and L_5, respectively, which have the following bases.

```
——————————— Example ———————————
gap> bas2:= [ [ 0, 1,-1, 0, 0, 0, 0, 0, 0, 0 ],
>            [ 0, 0, 1,-1, 0, 0, 0, 0, 0, 0 ],
>            [ 0, 0, 0, 0, 1,-1, 0, 0, 0, 0 ],
>            [ 0, 0, 0, 0, 0, 1,-1, 0, 0, 0 ],
>            [ 0, 0, 0, 0, 0, 1, 0,-1, 0, 0 ],
>            [ 0, 0, 0, 0, 0, 0, 0, 1,-1, 0 ],
>            [ 0, 0, 0, 0, 0, 0, 0, 0, 1,-1 ],
>            [ 0, 0, 0, 1, 0, 0, 0, 0, 0,-1 ],
>            [ 0, 1, 0, 0, 0, 0, 0, 1, 0, 0 ],
>            [ 1, 0, 0, 0, 1, 0, 0, 0, 0, 0 ] ];;
gap> bas5:= [ [ 0,-1, 1, 1,-1, 1, 1,-1,-1, 0 ],
>            [ 1, 0,-1,-1,-1, 1, 1,-1,-1, 0 ],
>            [ 0, 1, 1,-1, 1, 1,-1, 0, 1, 1 ],
>            [ 1, 1, 0,-1, 0,-1, 1,-1, 1,-1 ],
>            [-1, 0,-1, 1, 1, 0,-1,-1, 1,-1 ],
>            [ 0, 1,-1, 1, 1,-1, 1, 1, 0,-1 ],
>            [-1,-1, 1, 1, 0,-1,-1,-1,-1, 0 ],
>            [ 1,-1, 0,-1, 1,-1, 1, 1, 0,-1 ],
>            [-1, 1,-1, 1,-1, 0,-1, 1, 0,-1 ],
>            [ 1,-1,-1, 1, 1, 1, 0, 0,-1,-1 ] ];;
```

For the sake of simplicity, we rewrite the action to actions on L_2 and L_5, respectively.

```
——————————— Example ———————————
gap> B2:= Basis( Rationals^Length( bas2 ), bas2 );;
gap> bbas2:= List( bas2, x -> Coefficients( B2, x * b ) );;
gap> cbas2:= List( bas2, x -> Coefficients( B2, x * c ) );;
gap> B5:= Basis( Rationals^Length( bas5 ), bas5 );;
gap> bbas5:= List( bas5, x -> Coefficients( B5, x * b ) );;
gap> cbas5:= List( bas5, x -> Coefficients( B5, x * c ) );;
```

The library character table with identifier `"P21/G3/L2/V1/ext2"` belongs to the factor group of S modulo the normal subgroup $M(2L_2)$, so we compute the action on an orbit modulo 2.

```
——————————— Example ———————————
gap> v:= List( [ 1, 2 ], i -> 0 * bbas2[1] );;
gap> t:= [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ];;
gap> sgens:= generatorsOfPerfectSpaceGroup( [ bbas2, cbas2 ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 2 );;
gap> seed:= [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 ];;
gap> orb:= Orbit( g, seed, fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
```

```
gap> verifyFactorGroup( permgens, "P21/G3/L2/V1/ext2" );
true
```

The library character table with identifier "P21/G3/L5/V1/ext2" belongs to the factor group of S modulo the normal subgroup $M(2L_5)$, so we compute the action on an orbit modulo 2.

```
                              ____ Example ____
gap> sgens:= generatorsOfPerfectSpaceGroup( [ bbas5, cbas5 ], v, t );;
gap> g:= Group( sgens );;
gap> orb:= Orbit( g, seed, fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P21/G3/L5/V1/ext2" );
true
```

1.2.8 Examples with point group $L_2(8)$

There are two examples with $d = 7$. In both cases, the generators of the point group are as follows (see [HP89, p. 327]).

```
                              ____ Example ____
gap> a:= [ [ 0,-1, 0, 1, 0,-1, 1],
>          [ 0, 0,-1, 0, 1,-1, 0],
>          [ 0, 0, 0,-1, 1, 0, 0],
>          [ 0, 0, 0,-1, 0, 0, 0],
>          [ 0, 0, 1,-1, 0, 0, 0],
>          [ 0,-1, 1, 0,-1, 0, 0],
>          [ 1,-1, 0, 1, 0,-1, 0] ];;
gap> b:= [ [-1, 0, 1, 0,-1, 1, 0],
>          [ 0,-1, 0, 1,-1, 0, 0],
>          [ 0, 0,-1, 1, 0, 0, 0],
>          [ 0, 0,-1, 0, 0, 0, 0],
>          [ 0, 1,-1, 0, 0, 0, 0],
>          [-1, 1, 0,-1, 0, 0, 0],
>          [-1, 0, 1, 0,-1, 0, 1] ];;
```

In both examples, the vector system is V_2. The translation lattice in the first example is the lattice $L = 3\mathbb{Z}^d$.

```
                              ____ Example ____
gap> v:= [ [ 2, 1, 0, 0, 0, 1, 4 ],
>          [ 2, 0, 0, 0, 0, 0, 0 ] ];;
gap> t:= [ 3, 0, 0, 0, 0, 0, 0 ];;
```

The library character table with identifier "P41/G1/L1/V3/ext3" belongs to the factor group of S modulo the normal subgroup $M(3L)$, so we compute the action on an orbit modulo 9.

The orbits in this action are quite long. we choose a seed vector from the fixed space of an element of order 7.

```
                              ____ Example ____
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> aa:= sgens[1];;
gap> bb:= sgens[2];;
```

```
gap> elm:= aa*bb;;
gap> Order( elm );
7
gap> fixed:= NullspaceMat( elm - aa^0 );
[ [ 1, 1, 1, 1, 1, 1, 1, 0 ], [ -4, 1, 1, -5, -5, 2, 0, 1 ] ]
gap> fun:= multiplicationModulo( 9 );;
gap> seed:= fun( fixed[2], aa^0 );
[ 5, 1, 1, 4, 4, 2, 0, 1 ]
gap> orb:= Orbit( g, seed, fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P41/G1/L1/V3/ext3" );
true
```

The translation lattice in the second example is the lattice $L = 6\mathbb{Z}^d$.

```
───────────────── Example ─────────────────
gap> t:= [ 6, 0, 0, 0, 0, 0, 0 ];;
```

The library character table with identifier "P41/G1/L1/V4/ext3" belongs to the factor group of S modulo the normal subgroup $M(6L)$, so we compute the action on an orbit modulo 18.

```
───────────────── Example ─────────────────
gap> fun:= multiplicationModulo( 18 );;
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> seed:= fun( fixed[2], aa^0 );
[ 14, 1, 1, 13, 13, 2, 0, 1 ]
gap> orb:= Orbit( g, seed, fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P41/G1/L1/V4/ext3" );
true
```

1.2.9   Example with point group M_{11}

There is one example with $d = 10$. The generators of the point group are as follows (see [HP89, p. 334]).

```
───────────────── Example ─────────────────
gap> a:= deletedPermutationMat( (1,9)(3,5)(7,11)(8,10), 11 );;
gap> b:= deletedPermutationMat( (1,4,3,2)(5,8,7,6), 11 );;
```

The vector system is V_2, and the translation lattice is $L = 2\mathbb{Z}^d$.

```
───────────────── Example ─────────────────
gap> v:= [ 0 * a[1],
>          [ 0, 0, 0, 0, 0, 0, 0, 0, 1, 1 ] ];;
gap> t:= [ 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ];;
```

The library character table with identifier "P48/G1/L1/V2/ext2" belongs to the factor group of S modulo the normal subgroup $M(2L)$, so we compute the action on an orbit modulo 4.

```
                         Example
  gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
  gap> g:= Group( sgens );;
  gap> fun:= multiplicationModulo( 4 );;
  gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 ], fun );;
  gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
  gap> verifyFactorGroup( permgens, "P48/G1/L1/V2/ext2" );
  true
```

1.2.10 Example with point group $U_3(3)$

There is one example with $d = 7$. The generators of the point group are as follows (see [HP89, p. 335]).

```
                         Example
  gap> a:= [ [ 0, 0,-1, 1, 0,-1, 1 ],
  >          [ 1, 0,-1, 1, 1,-1, 0 ],
  >          [ 0, 1,-1, 0, 1, 0,-1 ],
  >          [ 0, 1, 0,-1, 1, 0,-1 ],
  >          [-1, 1, 1,-1, 0, 1, 0 ],
  >          [-1, 0, 1,-1, 0, 0, 1 ],
  >          [ 0, 0, 0, 0, 0, 0, 1 ] ];;
  gap> b:= [ [ 0, 0, 0, 0, 0, 0, 1 ],
  >          [ 0, 0,-1, 1, 0,-1, 1 ],
  >          [ 1, 0,-1, 1, 1,-1, 0 ],
  >          [ 0, 1,-1, 0, 1, 0,-1 ],
  >          [ 0, 1, 0,-1, 1, 0,-1 ],
  >          [-1, 1, 1,-1, 0, 1, 0 ],
  >          [-1, 0, 1,-1, 0, 0, 1 ] ];;
```

The vector system is V_2, and the translation lattice is $L = 3\mathbb{Z}^d$.

```
                         Example
  gap> v:= [ [ 2, 1, 0, 0, 2, 1, 0 ],
  >          0 * b[1] ];;
  gap> t:= [ 3, 0, 0, 0, 0, 0, 0 ];;
```

The library character table with identifier `"P49/G1/L1/V2/ext3"` belongs to the factor group of S modulo the normal subgroup $M(3L)$, so we compute the action on an orbit modulo 9.

```
                         Example
  gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
  gap> g:= Group( sgens );;
  gap> fun:= multiplicationModulo( 9 );;
```

The orbits in this action are quite long. we choose a seed vector from the fixed space of an element of order 12.

```
                         Example
  gap> aa:= sgens[1];;
  gap> bb:= sgens[2];;
  gap> elm:= aa*bb^4;;
  gap> Order( elm );
```

```
12
gap> fixed:= NullspaceMat( elm - aa^0 );
[ [ -1, -1, 1, 1, -1, -1, 1, 0 ], [ 0, -3, 1, 1, -1, -2, 0, 1 ] ]
gap> seed:= fun( fixed[2], aa^0 );
[ 0, 6, 1, 1, 8, 7, 0, 1 ]
gap> orb:= Orbit( g, seed, fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P49/G1/L1/V2/ext3" );
true
```

1.2.11 Examples with point group $U_4(2)$

There are two examples with $d = 6$. In both cases, the generators of the point group are as follows (see [HP89, p. 336]).

```
─────────────────────────── Example ───────────────────────────
gap> a:= [ [ 0, 1, 0,-1,-1, 1 ],
>          [ 1, 0,-1, 0, 1, 0 ],
>          [ 0, 0, 0,-1, 0, 1 ],
>          [ 0, 0,-1, 0, 0, 1 ],
>          [ 0, 0, 0, 0, 1, 0 ],
>          [ 0, 0, 0, 0, 0, 1 ] ];;
gap> b:= [ [ 0,-1, 0, 1, 0,-1 ],
>          [ 0, 1, 0,-1,-1, 0 ],
>          [ 0, 0, 1, 1, 0,-1 ],
>          [ 0, 0, 0, 0,-1, 0 ],
>          [ 0, 1, 0, 0, 0, 0 ],
>          [ 1, 0, 0, 0, 0, 0 ] ];;
```

In both examples, the vector system is the trivial vector system V_1, and the translation lattice is the full lattice $L_1 = \mathbb{Z}^d$.

```
─────────────────────────── Example ───────────────────────────
gap> v:= List( [ 1, 2 ], i -> 0 * a[1] );;
gap> t:= [ 1, 0, 0, 0, 0, 0 ];;
```

The library character table with identifier "P50/G1/L1/V1/ext3" belongs to the factor group of S modulo the normal subgroup $M(3L_1)$, so we compute the action on an orbit modulo 3.

```
─────────────────────────── Example ───────────────────────────
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 3 );;
gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 0, 1 ], fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P50/G1/L1/V1/ext3" );
true
```

The library character table with identifier "P50/G1/L1/V1/ext4" belongs to the factor group of S modulo the normal subgroup $M(4L_1)$, so we compute the action on an orbit modulo 4.

```
─────────────────────── Example ───────────────────────
gap> sgens:= generatorsOfPerfectSpaceGroup( [ a, b ], v, t );;
gap> g:= Group( sgens );;
gap> fun:= multiplicationModulo( 4 );;
gap> orb:= Orbit( g, [ 1, 0, 0, 0, 0, 0, 1 ], fun );;
gap> permgens:= List( sgens, x -> Permutation( x, orb, fun ) );;
gap> verifyFactorGroup( permgens, "P50/G1/L1/V1/ext4" );
true
```

1.2.12   A remark on one of the example groups

The (perfect) character table with identifier `"P1/G2/L2/V2/ext4"` has the property that its character degrees are exactly the divisors of 60.

```
─────────────────────── Example ───────────────────────
gap> degrees:= CharacterDegrees( CharacterTable( "P1/G2/L2/V2/ext4" ) );
[ [ 1, 1 ], [ 2, 2 ], [ 3, 2 ], [ 4, 2 ], [ 5, 1 ], [ 6, 5 ],
  [ 10, 4 ], [ 12, 4 ], [ 15, 20 ], [ 20, 2 ], [ 30, 29 ], [ 60, 8 ] ]
gap> List( degrees, x -> x[1] ) = DivisorsInt( 60 );
true
```

There are nilpotent groups with the same set of character degrees, for example the direct product of four extraspecial groups of the orders 2^3, 2^3, 3^3, and 5^3, respectively. This phenomenon has been described in [NR14].

1.3   Generality problems (December 2004/October 2015)

The term "generality problem" is used for problems concerning consistent choices of conjugacy classes of Brauer tables for the same group, in different characteristics. The definition and some examples are given in [JLPW95, p. x].

Section 1.3.1 shows how to detect generality problems and lists the known generality problems, and Section 1.3.2 gives an example that actually arose.

1.3.1   Listing possible generality problems

We use the following idea for finding character tables which may involve generality problems. (The functions shown in this section are based on GAP 3 code that was originally written by Jürgen Müller.)

If the p-modular Brauer table *mtbl*, say, of a group contributes to a generality problem then some choice of conjugacy classes is necessary in order to write down this table, in the sense that some symmetry of the corresponding ordinary table *tbl*, say, is broken in *mtbl*. This situation can be detected as follows. We assume that the class fusion from *mtbl* to *tbl* has been fixed. All possible class fusions are obtained as the orbit of this class fusion under the actions of table automorphisms of *tbl*, via mapping the images of the class fusion (with the function `OnTuples` (**Reference: OnTuples**)), and of the table automorphisms of *mtbl*, via permuting the preimages. The case of broken symmetries occurs if and only if this orbit splits into several orbits when only the action of the table automorphisms of *mtbl* is considered. Equivalently, symmetries are broken if and only if the orbit under table automorphisms of *mtbl* is not closed under the action of table automorphisms of *tbl*.

```
――――――――――――― Example ―――――――――――――
gap> BrokenSymmetries:= function( ordtbl, modtbl )
>     local taut, maut, triv, fus, orb;
>     taut:= AutomorphismsOfTable( ordtbl );
>     maut:= AutomorphismsOfTable( modtbl );
>     triv:= TrivialSubgroup( taut );
>     fus:= GetFusionMap( modtbl, ordtbl );
>     orb:= MakeImmutable( Set( OrbitFusions( maut, fus, triv ) ) );
>     return ForAny( GeneratorsOfGroup( taut ),
>               x -> ForAny( orb,
>                     fus -> not OnTuples( fus, x ) in orb ) );
> end;;
```

Remark: (Thanks to Klaus Lux for discussions on this topic.)

- It may happen that some symmetry σ_m of a Brauer table does not belong to a symmetry σ_o of the corresponding ordinary table, in the sense that permuting the preimage classes of a fusion f between the two tables with σ_m and permuting the image classes with σ_o yields f.

  For example, consider the group $G = 2.A_6.2_1$, the double cover of the symmetric group S_6 on six points. The 2-modular Brauer table of G, which is essentially equal to that of S_6, has a table automorphism group order two, and the nonidentity element in it swaps the two classes of element order three. The automorphism group of the ordinary character table of G, however, fixes the two classes of element order three; note that exactly one of these classes possesses square roots in the "outer half" $G \setminus G'$.

  Thus it is not sufficient to compare the orbit of the fixed class fusion under the automorphisms of the ordinary table with the orbit of the same fusion under the automorphisms of the Brauer table.

```
――――――――――――― Example ―――――――――――――
gap> t:= CharacterTable( "2.A6.2_1" );;
gap> m:= t mod 2;;
gap> GetFusionMap( m, t );
[ 1, 4, 6, 9 ]
gap> AutomorphismsOfTable( t );
Group([ (16,17), (14,15), (14,15)(16,17) ])
gap> AutomorphismsOfTable( m );
Group([ (2,3) ])
gap> Display( m );
2.A6.2_1mod2

     2  5  2  2  1
     3  2  2  2  .
     5  1  .  .  1

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

X.1     1  1  1  1
X.2     4  1 -2 -1
```

```
 X.3     4 -2  1 -1
 X.4    16 -2 -2  1
gap> Display( t );
2.A6.2_1

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

           1a  2a 4a 3a 6a 3b 6b 8a 5a 10a 2b 4b 8b 6c 6d 12a 12b
       2P  1a  1a 2a 3a 3a 3b 3b 4a 5a  5a 1a 2a 4a 3a 3a  6b  6b
       3P  1a  2a 4a 1a 2a 1a 2a 8a 5a 10a 2b 4b 8b 2b 2b  4b  4b
       5P  1a  2a 4a 3a 6a 3b 6b 8a 1a  2a 2b 4b 8b 6d 6c 12b 12a

 X.1       1   1  1  1  1  1  1  1  1   1  1  1  1  1  1   1   1
 X.2       1   1  1  1  1  1  1  1  1   1 -1 -1 -1 -1 -1  -1  -1
 X.3       5   5  1  2  2 -1 -1 -1  .   .  3 -1  1  .  .  -1  -1
 X.4       5   5  1  2  2 -1 -1 -1  .   . -3  1 -1  .  .   1   1
 X.5       5   5  1 -1 -1  2  2 -1  .   . -1  3  1 -1 -1   .   .
 X.6       5   5  1 -1 -1  2  2 -1  .   .  1 -3 -1  1  1   .   .
 X.7      16  16  . -2 -2 -2 -2  .  1   1  .  .  .  .  .   .   .
 X.8       9   9  1  .  .  .  .  1 -1  -1  3  3 -1  .  .   .   .
 X.9       9   9  1  .  .  .  .  1 -1  -1 -3 -3  1  .  .   .   .
 X.10     10  10 -2  1  1  1  1  .  .   .  2 -2  . -1 -1   1   1
 X.11     10  10 -2  1  1  1  1  .  .   . -2  2  .  1  1  -1  -1
 X.12      4  -4  . -2  2  1 -1  . -1   1  .  .  .  .  .   B  -B
 X.13      4  -4  . -2  2  1 -1  . -1   1  .  .  .  .  .  -B   B
 X.14      4  -4  .  1 -1 -2  2  . -1   1  .  .  .  A -A   .   .
 X.15      4  -4  .  1 -1 -2  2  . -1   1  .  .  . -A  A   .   .
 X.16     16 -16  . -2  2 -2  2  .  1  -1  .  .  .  .  .   .   .
 X.17     20 -20  .  2 -2  2 -2  .  .   .  .  .  .  .  .   .   .

A = E(3)-E(3)^2
  = Sqrt(-3) = i3
B = -E(12)^7+E(12)^11
  = Sqrt(3) = r3
```

When considering several characteristics in parallel, one argues as follows. The possible class fusions from a Brauer table *mtbl* to its ordinary table *tbl* are given by the orbit of a fixed class fusion under the action of the table automorphisms of *tbl*. If there are several orbits under the action of the automorphisms of *mtbl* then we choose one orbit. Due to this choice, only those table automorphisms of *tbl* are admissible for other characteristics that stabilize the chosen orbit. For the second characteristic, we take again the set of all class fusions from the Brauer table to *tbl*, and split it into orbits under the table automorphisms of the Brauer table. Now there are two possibilities. Either the action of the admissible subgroup of automorphisms of *tbl* joins these orbits into one orbit or not. In the former case, we choose again one of the orbits, replace the group of admissible automorphisms of *tbl* by the stabilizer of this orbit, and proceed with the next characteristic. In the latter case, we have found a generality problem, since we are not free to choose an arbitrary class fusion from the set of possibilities.

The following function returns the set of primes which may be involved in generality problems for the given ordinary character table. Note that the procedure sketched above does not tell which

characteristics are actually involved or which classes are affected by the choices; for example, we could argue that one is always free to choose a fusion for the first characteristics, and that only the other ones cause problems. We return *all* those primes p for which broken symmetries between the p-modular table and the ordinary table have been detected.

```
───────────────── Example ─────────────────
gap> PrimesOfGeneralityProblems:= function( ordtbl )
>     local consider, p, modtbl, taut, triv, admiss, fusion, maut,
>           allfusions, orbits, orbit, reps;
>     # Find the primes for which symmetries are broken.
>     consider:= [];
>     for p in Filtered( PrimeDivisors( Size( ordtbl ) ), IsPrimeInt ) do
>       modtbl:= ordtbl mod p;
>       if modtbl <> fail and BrokenSymmetries( ordtbl, modtbl ) then
>         Add( consider, p );
>       fi;
>     od;
>     # Compute the choices and detect generality problems.
>     taut:= AutomorphismsOfTable( ordtbl );
>     triv:= TrivialSubgroup( taut );
>     admiss:= taut;
>     for p in consider do
>       modtbl:= ordtbl mod p;
>       fusion:= GetFusionMap( modtbl, ordtbl );
>       maut:= AutomorphismsOfTable( modtbl );
>       # - We need not apply the action of 'maut' here,
>       #   since 'maut' will later be used to get representatives.
>       # - We need not apply all elements in 'taut' but only
>       #   representatives of left cosets of 'admiss' in 'taut',
>       #   since 'admiss' will later be used to get representatives.
>       # allfusions:= OrbitFusions( maut, fusion, taut );
>       allfusions:= Set( RightTransversal( taut, admiss ),
>                         x -> OnTuples( fusion, x^-1 ) );
>       # For computing representatives, 'RepresentativesFusions' is not
>       # suitable because 'allfusions' is in generally not closed
>       # under the actions.
>       # reps:= RepresentativesFusions( maut, allfusions, admiss );
>       orbits:= [];
>       while not IsEmpty( allfusions ) do
>         orbit:= OrbitFusions( maut, allfusions[1], admiss );
>         Add( orbits, orbit );
>         SubtractSet( allfusions, orbit );
>       od;
>       reps:= List( orbits, x -> x[1] );
>       if Length( reps ) = 1 then
>         # Reduce the symmetries that are still available.
>         admiss:= Stabilizer( admiss,
>                             Set( OrbitFusions( maut, fusion, triv ) ),
>                             OnSetsTuples );
>       else
>         # We have found a generality problem.
>         return consider;
>       fi;
```

```
>         od;
>         # There is no generality problem for this table.
>         return [];
> end;;
```

Let us look at a small example, the 5-modular character table of the group $2.A_5.2$. The irreducible characters of degree 2 have the values $\pm\sqrt{-2}$ on the classes 8a and 8b, and the values $\pm\sqrt{-3}$ on the classes 6b and 6c. When we define which of the two classes of element order 8 is called 8a, this will also define which class is called 6b. The ordinary character table does not relate the two pairs of classes, there are table automorphisms which interchange each pair independently. This symmetry is thus broken in the 5-modular character table.

```
──────────────────────── Example ────────────────────────
 gap> t:= CharacterTable( "2.A5.2" );;
 gap> m:= t mod 5;;
 gap> Display( m );
 2.A5.2mod5

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

        1a 2a 4a 3a 6a 2b 8a 8b 6b 6c
     2P 1a 1a 2a 3a 3a 1a 4a 4a 3a 3a
     3P 1a 2a 4a 1a 2a 2b 8a 8b 2b 2b
     5P 1a 2a 4a 3a 6a 2b 8b 8a 6c 6b

 X.1      1  1  1  1  1  1  1  1  1  1
 X.2      1  1  1  1  1 -1 -1 -1 -1 -1
 X.3      3  3 -1  .  .  1 -1 -1 -2 -2
 X.4      3  3 -1  .  . -1  1  1  2  2
 X.5      5  5  1 -1 -1  1 -1 -1  1  1
 X.6      5  5  1 -1 -1 -1  1  1 -1 -1
 X.7      2 -2  . -1  1  .  A -A  B -B
 X.8      2 -2  . -1  1  . -A  A -B  B
 X.9      4 -4  .  1 -1  .  .  .  B -B
 X.10     4 -4  .  1 -1  .  .  . -B  B

 A = E(8)+E(8)^3
   = Sqrt(-2) = i2
 B = E(3)-E(3)^2
   = Sqrt(-3) = i3
 gap> AutomorphismsOfTable( t );
 Group([ (11,12), (9,10) ])
 gap> AutomorphismsOfTable( m );
 Group([ (7,8)(9,10) ])
 gap> GetFusionMap( m, t );
 [ 1, 2, 3, 4, 5, 8, 9, 10, 11, 12 ]
 gap> BrokenSymmetries( t, m );
 true
 gap> BrokenSymmetries( t, t mod 2 );
 false
 gap> BrokenSymmetries( t, t mod 3 );
```

```
false
gap> PrimesOfGeneralityProblems( t );
[  ]
```

Since no symmetry is broken in the 2- and 3-modular character tables of G, there is no generality problem in this case.

For an example of a generality problem, we look at the smallest Janko group J_1. As is mentioned in [JLPW95, p. x], the unique irreducible 11-modular Brauer character of degree 7 distinguishes the two (algebraically conjugate) classes of element order 5. Since also the unique irreducible 19-modular Brauer character of degree 22 distinguishes these classes, we have to choose these classes consistently.

```
——————————————————— Example ———————————————————
gap> t:= CharacterTable( "J1" );;
gap> m:= t mod 11;;
gap> Display( m, rec( chars:= Filtered( Irr( m ), x -> x[1] = 7 ) ) );
J1mod11

     2  3  3  1  1  1  1  .  1  1  .  .  .  .  .  .
     3  1  1  1  1  1  1  .  .  .  1  1  .  .  .
     5  1  1  1  1  1  .  .  1  1  1  1  .  .  .
     7  1  .  .  .  .  .  1  .  .  .  .  .  .  .
    11  1  .  .  .  .  .  .  .  .  .  .  .  .  .
    19  1  .  .  .  .  .  .  .  .  .  .  1  1  1

        1a 2a 3a 5a 5b 6a 7a 10a 10b 15a 15b 19a 19b 19c
    2P  1a 1a 3a 5b 5a 3a 7a  5b  5a 15b 15a 19b 19c 19a
    3P  1a 2a 1a 5b 5a 2a 7a 10b 10a  5b  5a 19b 19c 19a
    5P  1a 2a 3a 1a 1a 6a 7a  2a  2a  3a  3a 19b 19c 19a
    7P  1a 2a 3a 5b 5a 6a 1a 10b 10a 15b 15a 19a 19b 19c
   11P  1a 2a 3a 5a 5b 6a 7a 10a 10b 15a 15b 19a 19b 19c
   19P  1a 2a 3a 5a 5b 6a 7a 10a 10b 15a 15b  1a  1a  1a

Y.1     7 -1  1  A *A -1  .   B  *B   C  *C   D   E   F

A = E(5)+E(5)^4
  = (-1+Sqrt(5))/2 = b5
B = -E(5)-2*E(5)^2-2*E(5)^3-E(5)^4
  = (3+Sqrt(5))/2 = 2+b5
C = -2*E(5)-2*E(5)^4
  = 1-Sqrt(5) = 1-r5
D = -E(19)-E(19)^2-E(19)^3-E(19)^5-E(19)^7-E(19)^8-E(19)^11-E(19)^12-E\
(19)^14-E(19)^16-E(19)^17-E(19)^18
E = -E(19)^2-E(19)^3-E(19)^4-E(19)^5-E(19)^6-E(19)^9-E(19)^10-E(19)^13\
-E(19)^14-E(19)^15-E(19)^16-E(19)^17
F = -E(19)-E(19)^4-E(19)^6-E(19)^7-E(19)^8-E(19)^9-E(19)^10-E(19)^11-E\
(19)^12-E(19)^13-E(19)^15-E(19)^18
gap> m:= t mod 19;;
gap> Display( m, rec( chars:= Filtered( Irr( m ), x -> x[1] = 22 ) ) );
J1mod19

     2  3  3  1  1  1  1  .  1  1  .  .  .
     3  1  1  1  1  1  1  .  .  .  .  1  1
```

```
    5  1  1  1  1  1  .  .  1  1  .  1  1
    7  1  .  .  .  .  .  1  .  .  .  .  .
   11  1  .  .  .  .  .  .  .  .  1  .  .
   19  1  .  .  .  .  .  .  .  .  .  .  .

       1a 2a 3a 5a 5b 6a 7a 10a 10b 11a 15a 15b
    2P 1a 1a 3a 5b 5a 3a 7a  5b  5a 11a 15b 15a
    3P 1a 2a 1a 5b 5a 2a 7a 10b 10a 11a  5b  5a
    5P 1a 2a 3a 1a 1a 6a 7a  2a  2a 11a  3a  3a
    7P 1a 2a 3a 5b 5a 6a 1a 10b 10a 11a 15b 15a
   11P 1a 2a 3a 5a 5b 6a 7a 10a 10b  1a 15a 15b
   19P 1a 2a 3a 5a 5b 6a 7a 10a 10b 11a 15a 15b

Y.1    22 -2  1  A *A  1  1  -A -*A   .   B  *B

A = E(5)+E(5)^4
  = (-1+Sqrt(5))/2 = b5
B = -2*E(5)-2*E(5)^4
  = 1-Sqrt(5) = 1-r5
```

Note that the degree 7 character above also distinguishes the three classes of element order 19, and the same holds for the unique irreducible degree 31 character from characteristic 7. Thus also the prime 7 occurs in the list of candidates for generality problems.

```
————————————————— Example ————————————————
gap> PrimesOfGeneralityProblems( t );
[ 7, 11, 19 ]
```

Finally, we list the candidates for generality problems from GAP's Character Table Library.

```
————————————————— Example ————————————————
gap> list:= [];;
gap> isGeneralityProblem:= function( ordtbl )
>      local res;
>      res:= PrimesOfGeneralityProblems( ordtbl );
>      if res = [] then
>        return false;
>      fi;
>      Add( list, [ Identifier( ordtbl ), res ] );
>      return true;
> end;;
gap> AllCharacterTableNames( IsDuplicateTable, false,
>        isGeneralityProblem, true );;
gap> PrintArray( SortedList( list ) );
[ [            (2.A4x2.G2(4)).2,          [ 2, 5, 7, 13 ] ],
  [            (2^2x3).L3(4).2_1,                [ 5, 7 ] ],
  [              (2x12).L3(4),              [ 2, 3, 7 ] ],
  [              (4^2x3).L3(4),             [ 2, 3, 7 ] ],
  [                (7:3xHe):2,            [ 5, 7, 17 ] ],
  [                (A5xA12):2,                [ 2, 3 ] ],
  [                (D10xHN).2,     [ 2, 3, 5, 7, 11, 19 ] ],
  [                (S3x2.Fi22).2,           [ 3, 11, 13 ] ],
  [                  12.M22,           [ 2, 5, 7, 11 ] ],
```

```
[                 12.M22.2,          [ 2, 5, 7, 11 ] ],
[          12_1.L3(4).2_1,                  [ 5, 7 ] ],
[              12_2.L3(4),              [ 2, 3, 7 ] ],
[          12_2.L3(4).2_1,              [ 3, 5, 7 ] ],
[          12_2.L3(4).2_2,              [ 2, 3, 7 ] ],
[          12_2.L3(4).2_3,              [ 2, 3, 7 ] ],
[          2.(A4xG2(4)).2,          [ 2, 5, 7, 13 ] ],
[                 2.2E6(2),              [ 13, 19 ] ],
[               2.2E6(2).2,              [ 13, 19 ] ],
[                   2.A10,                  [ 5, 7 ] ],
[                   2.A11,              [ 3, 5, 7 ] ],
[                 2.A11.2,              [ 5, 7, 11 ] ],
[                   2.A12,          [ 2, 3, 5, 7 ] ],
[                 2.A12.2,              [ 5, 7, 11 ] ],
[                   2.A13,       [ 2, 3, 5, 7, 11 ] ],
[                 2.A13.2,              [ 5, 7, 13 ] ],
[                2.Alt(14),          [ 2, 3, 5, 7 ] ],
[                2.Alt(15),              [ 2, 5, 7 ] ],
[                2.Alt(16),          [ 2, 3, 5, 7 ] ],
[                2.Alt(17),          [ 2, 3, 5, 7 ] ],
[                2.Alt(18),          [ 2, 3, 5, 7 ] ],
[                     2.B,              [ 17, 23 ] ],
[                  2.F4(2),         [ 2, 7, 13, 17 ] ],
[                2.Fi22.2,              [ 11, 13 ] ],
[                  2.G2(4),                  [ 2, 7 ] ],
[                2.G2(4).2,              [ 5, 7, 13 ] ],
[                    2.HS,          [ 3, 5, 7, 11 ] ],
[                  2.HS.2,                 [ 3, 11 ] ],
[           2.L3(4).2_1,                  [ 5, 7 ] ],
[                    2.Ru,         [ 5, 7, 13, 29 ] ],
[                   2.Suz,              [ 2, 5, 11 ] ],
[                 2.Suz.2,              [ 3, 7, 13 ] ],
[                2.Sym(15),              [ 3, 5, 7 ] ],
[                2.Sym(16),              [ 3, 5, 7 ] ],
[                2.Sym(17),              [ 3, 5, 7 ] ],
[                2.Sym(18),                  [ 5, 7 ] ],
[                  2.Sz(8),             [ 2, 5, 13 ] ],
[               2^2.2E6(2),              [ 13, 19 ] ],
[             2^2.2E6(2).2,              [ 13, 19 ] ],
[              2^2.Fi22.2,           [ 3, 11, 13 ] ],
[           2^2.L3(4).2^2,                  [ 5, 7 ] ],
[           2^2.L3(4).2_1,                  [ 5, 7 ] ],
[               2^2.Sz(8),             [ 2, 5, 13 ] ],
[                2x2.F4(2),         [ 2, 7, 13, 17 ] ],
[                2x3.Fi22,              [ 2, 3, 5 ] ],
[                2x6.Fi22,              [ 2, 3, 5 ] ],
[                 2x6.M22,              [ 2, 5, 11 ] ],
[                2xFi22.2,              [ 11, 13 ] ],
[                  2xFi23,             [ 3, 17, 23 ] ],
[                 3.Fi22,              [ 2, 3, 5 ] ],
[                3.Fi22.2,          [ 2, 5, 11, 13 ] ],
[                    3.J3,             [ 2, 17, 19 ] ],
```

```
[                       3.J3.2,              [ 2, 5, 17, 19 ] ],
[                  3.L3(4).2_3,                  [ 2, 3, 7 ] ],
[                3.L3(4).3.2_3,                  [ 2, 3, 7 ] ],
[                   3.L3(7).2,                 [ 3, 7, 19 ] ],
[                  3.L3(7).S3,                 [ 3, 7, 19 ] ],
[                       3.McL,                 [ 2, 5, 11 ] ],
[                     3.McL.2,              [ 2, 3, 5, 11 ] ],
[                        3.ON,        [ 3, 7, 11, 19, 31 ] ],
[                      3.ON.2,     [ 3, 5, 7, 11, 19, 31 ] ],
[                     3.Suz.2,                 [ 2, 3, 13 ] ],
[                   3x2.F4(2),          [ 2, 7, 13, 17 ] ],
[                  3x2.Fi22.2,                 [ 11, 13 ] ],
[                   3x2.G2(4),                    [ 2, 7 ] ],
[                     3xFi23,              [ 3, 17, 23 ] ],
[                        3xJ1,             [ 7, 11, 19 ] ],
[                   3xL3(7).2,                 [ 3, 7, 19 ] ],
[                      4.HS.2,                 [ 5, 7, 11 ] ],
[                       4.M22,                    [ 5, 7 ] ],
[              4_1.L3(4).2_1,                    [ 5, 7 ] ],
[              4_2.L3(4).2_1,                 [ 3, 5, 7 ] ],
[                     6.Fi22,                 [ 2, 3, 5 ] ],
[                   6.Fi22.2,             [ 2, 5, 11, 13 ] ],
[                6.L3(4).2_1,                    [ 5, 7 ] ],
[                       6.M22,                 [ 2, 5, 11 ] ],
[                     6.O7(3),                 [ 3, 5, 13 ] ],
[                   6.O7(3).2,                 [ 3, 5, 13 ] ],
[                       6.Suz,                 [ 2, 5, 11 ] ],
[                     6.Suz.2,          [ 2, 3, 5, 7, 13 ] ],
[                   6x2.F4(2),          [ 2, 7, 13, 17 ] ],
[                         A12,                    [ 2, 3 ] ],
[                         A14,                 [ 2, 5, 7 ] ],
[                         A17,                    [ 2, 7 ] ],
[                         A18,              [ 2, 3, 5, 7 ] ],
[                           B,        [ 13, 17, 23, 31 ] ],
[                         F3+,             [ 17, 23, 29 ] ],
[                       F3+.2,             [ 17, 23, 29 ] ],
[                      Fi22.2,                 [ 11, 13 ] ],
[                        Fi23,              [ 3, 17, 23 ] ],
[                          HN,          [ 2, 3, 11, 19 ] ],
[                        HN.2,          [ 5, 7, 11, 19 ] ],
[                          He,                 [ 5, 17 ] ],
[                        He.2,                 [ 5, 7, 17 ] ],
[         Isoclinic(12.M22.2),             [ 2, 5, 7, 11 ] ],
[  Isoclinic(12_1.L3(4).2_1),                    [ 5, 7 ] ],
[  Isoclinic(12_2.L3(4).2_1),                 [ 3, 5, 7 ] ],
[  Isoclinic(12_2.L3(4).2_3),                 [ 2, 3, 7 ] ],
[         Isoclinic(2.A11.2),                 [ 5, 7, 11 ] ],
[         Isoclinic(2.A12.2),                 [ 5, 7, 11 ] ],
[         Isoclinic(2.A13.2),                 [ 5, 7, 13 ] ],
[        Isoclinic(2.Fi22.2),                 [ 11, 13 ] ],
[        Isoclinic(2.G2(4).2),                [ 5, 7, 13 ] ],
[          Isoclinic(2.HS.2),                 [ 3, 11 ] ],
```

```
[             Isoclinic(2.HSx2),          [ 3, 5, 7, 11 ] ],
[         Isoclinic(2.L3(4).2_1),              [ 5, 7 ] ],
[           Isoclinic(2.Suz.2),          [ 3, 7, 13 ] ],
[   Isoclinic(4_1.L3(4).2_1),                  [ 5, 7 ] ],
[   Isoclinic(4_2.L3(4).2_1),          [ 3, 5, 7 ] ],
[         Isoclinic(6.Fi22.2),        [ 2, 5, 11, 13 ] ],
[     Isoclinic(6.L3(4).2_1),                  [ 5, 7 ] ],
[         Isoclinic(6.O7(3).2),          [ 3, 5, 13 ] ],
[           Isoclinic(6.Suz.2),      [ 2, 3, 5, 7, 13 ] ],
[                           J1,          [ 7, 11, 19 ] ],
[                         J1x2,          [ 7, 11, 19 ] ],
[                           J3,          [ 2, 17, 19 ] ],
[                         J3.2,       [ 2, 5, 17, 19 ] ],
[                     L3(4).2_3,              [ 3, 7 ] ],
[                   L3(4).3.2_3,          [ 2, 3, 7 ] ],
[                       L3(7).2,          [ 3, 7, 19 ] ],
[                       L3(7).S3,          [ 3, 7, 19 ] ],
[                     L3(9).2_1,          [ 3, 7, 13 ] ],
[                       L5(2).2,          [ 2, 7, 31 ] ],
[                            Ly,          [ 7, 37, 67 ] ],
[                           M23,          [ 2, 3, 23 ] ],
[                            ON,    [ 3, 7, 11, 19, 31 ] ],
[                          ON.2,    [ 3, 5, 7, 11, 19, 31 ] ],
[                            Ru,          [ 5, 7, 13, 29 ] ],
[                     S3xFi22.2,             [ 11, 13 ] ],
[                         Suz.2,              [ 3, 13 ] ] ]
```

Note that this list may become longer as new Brauer tables become available. (For example, the prime 2 was added to the entries for extensions of $F_4(2)$ when the 2-modular table of $F_4(2)$ became available.)

1.3.2 A generality problem concerning the group J_3 (April 2015)

In March 2015, Klaus Lux reported an inconsistency in the character data of GAP:

The sporadic simple Janko group J_3 has a unique 19-modular irreducible Brauer character of degree 110. In the character table that is printed in the Atlas of Brauer characters [JLPW95, p. 219], the Brauer character value on the class 17A is b_{17}. The Atlas of Group Representations [WWT$^+$] provides a straight line program for computing class representatives of J_3. If we compute the Brauer character value in question, we do not get b_{17} but its algebraic conjugate, $-1 - b_{17}$.

```
─────────────────────── Example ───────────────────────
gap> t:= CharacterTable( "J3" );;
gap> m:= t mod 19;;
gap> cand:= Filtered( Irr( m ), x -> x[1] = 110 );;
gap> Length( cand );
1
gap> slp:= AtlasProgram( "J3", "classes" );;
gap> 17a:= Position( slp.outputs, "17A" );
18
gap> info:= OneAtlasGeneratingSetInfo( "J3", Characteristic, 19,
>             Dimension, 110 );;
gap> gens:= AtlasGenerators( info );;
```

```
gap> reps:= ResultOfStraightLineProgram( slp.program,
>               gens.generators );;
gap> Quadratic( BrauerCharacterValue( reps[ 17a ] ) );
rec( ATLAS := "-1-b17", a := -1, b := -1, d := 2,
  display := "(-1-Sqrt(17))/2", root := 17 )
```

How shall we resolve this inconsistency, by replacing the straight line program or by swapping the classes 17A and 17B in the character table? Before we decide this, we look at related information.

Table 1.3.2 lists the p-modular irreducible characters of J_3, according to [JLPW95], that can be used to define which of the two classes of element order 17 shall be called 17A; a $+$ sign in the last column of the table indicates that the representation is available in the Atlas of Group Representations.

p	$\varphi(1)$	$\varphi(17A)$	$\varphi(17B)$	Atlas?
2	78	$1 - b_{17}$	$2 + b_{17}$	$+$
2	80	$3 - b_{17}$	$4 + b_{17}$	$+$
2	244	$b_{17} - 2$	$-3 - b_{17}$	$+$
2	966	$r_{17} - 3$	$-3 - r_{17}$	$+$
19	110	b_{17}	$-1 - b_{17}$	$+$
19	214	$1 - b_{17}$	$2 + b_{17}$	$+$
19	706	$-b_{17}$	$1 + b_{17}$	$+$
19	1214	$-1 + b_{17}$	$-2 - b_{17}$	$-$

Table: Representations of J_3 that may define 17A

Note that the irreducible Brauer characters in characteristic 3 and 5 that distinguish the two classes 17A and 17B occur in pairs of Galois conjugate characters.

The following computations show that the given straight line program is compatible with the four characters in characteristic 2 but is not compatible with the three available characters in characteristic 19.

```
────────────────────── Example ──────────────────────
gap> table:= [];;
gap> for pair in [ [  2, [ 78, 80, 244, 966 ] ],
>                  [ 19, [ 110, 214, 706 ] ] ] do
>      p:= pair[1];
>      for d in pair[2] do
>        info:= OneAtlasGeneratingSetInfo( "J3", Characteristic, p,
>                   Dimension, d );
>        gens:= AtlasGenerators( info );
>        reps:= ResultOfStraightLineProgram( slp.program,
>                   gens.generators );
>        val:= BrauerCharacterValue( reps[ 17a ] );
>        Add( table, [ p, d, Quadratic( val ).ATLAS,
>                      Quadratic( StarCyc( val ) ).ATLAS ] );
>      od;
>    od;
gap> PrintArray( table );
[ [        2,       78,    1-b17,    2+b17 ],
  [        2,       80,    3-b17,    4+b17 ],
  [        2,      244,   -2+b17,   -3-b17 ],
  [        2,      966,   -3+r17,   -3-r17 ],
```

```
[     19,    110,   -1-b17,      b17 ],
[     19,    214,    2+b17,    1-b17 ],
[     19,    706,    1+b17,     -b17 ] ]
```

We see that the problem is an inconsistency between the 2-modular and the 19-modular character table of J_3 in [JLPW95]. In particular, changing the straight line program would not help to resolve the problem.

How shall we proceed in order to fix the problem? We can decide to keep the 19-modular table of J_3, and to swap the two classes of element order 17 in the 2-modular table; then also the straight line program has to be changed, and the classes of element orders 17 and 51 in the 2-modular character table of the triple cover $3.J_3$ of J_3 have to be adjusted. Alternatively, we can keep the 2-modular table of J_3 and the straight line program, and adjust the conjugacy classes of element orders divisible by 17 in the 19-modular character tables of J_3, $3.J_3$, $J_3.2$, and $3.J_3.2$.

We decide to change the 19-modular character tables. Note that these character tables ---or equivalently, the corresponding Brauer trees--- have been described in [HL89], where explicit choices are mentioned that lead to the shown Brauer trees. Questions about the consistency with Brauer tables in other characteristic had not been an issue in this book. (Only the consistency of the Brauer trees among the 19-blocks of $3.J_3$ is mentioned.) In fact, the book mentions that the 19-modular Brauer trees for J_3 had been computed already by W. Feit. The inconsistency of Brauer character tables in different characteristic has apparently been overlooked when the data for [JLPW95] have been put together, and had not been detected until now.

Remarks:

- Such a change of a Brauer table can in general affect the class fusions from and to this table. Note that Brauer tables may impose conditions on the choice of the fusion among possible fusions that are equivalent w. r. t. the table automorphisms of the ordinary table. In this particular case, in fact no class fusion had to be changed, see the sections 9.6.1 and Section 9.6.3.

- The change of the character tables affects the decomposition matrices. Thus the PDF files containing the 19-modular decomposition matrices had to be updated, see `http://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib/dec/tex/J3/index.html`.

- Jürgen Müller has checked that the conjugacy classes of all Brauer tables of J_3, $3.J_3$, $J_3.2$, $3.J_3.2$ are consistent after the fix described above.

1.3.3   A generality problem concerning the group *HN* (August 2022)

The classes 20A, 20B of the Harada-Norton group *HN* in the 11- and 19-modular character tables are determined by unique Brauer characters that have different values on these classes. Once we have *defined* these classes in one characteristic, the two Brauer characters tell us how to *choose* them consistently in the other characteristic. Thus the question is whether the two Brauer tables are consistent w.r.t. this property or not.

(Note that this question can be answered independently of all other questions of this kind for *HN*, because the permutation that swaps exactly the classes 20A and 20B is a table automorphism of the ordinary character table of *HN*.)

We start with the ordinary character table of *HN*. There are exactly two ordinary irreducible characters that take different values on the classes 20A, 20B, these are χ_{51} and χ_{52}.

```
gap> t:= CharacterTable( "HN" );;
gap> pos20:= Positions( OrdersClassRepresentatives( t ), 20 );
[ 39, 40, 41, 42, 43 ]
gap> diff:= Filtered( Irr( t ), x -> x[39] <> x[40] );;
gap> List( diff, x -> Position( Irr( t ), x ) );
[ 51, 52 ]
```

These values are irrational and lie in the field that is generated by the square root of 5.

```
gap> List( diff, x -> List( x{ [ 1, 39, 40 ] },
>                           CTblLib.StringOfAtlasIrrationality ) );
[ [ "5103000", "2r5+1", "-2r5+1" ], [ "5103000", "-2r5+1", "2r5+1" ] ]
```

In each of the characteristics $p \in \{11, 19\}$, the p-modular reductions of χ_{51} and χ_{52} decompose differently into irreducibles. Note that the Galois automorphism of the ordinary character table that maps $\sqrt{5}$ to $-\sqrt{5}$ does not live in the 11- and 19-modular Brauer tables.

For $p = 11$, the reduction of χ_{51} is $\varphi_{40} + \varphi_{48}$, with $\varphi_{40}(1) = 1\,575\,176$ and $\varphi_{48}(1) = 3\,527\,824$, and the reduction of χ_{52} is $\varphi_{39} + \varphi_{49}$, with $\varphi_{39}(1) = 1\,361\,919$ and $\varphi_{49}(1) = 3\,741\,081$.

```
gap> t11:= t mod 11;;
gap> rest11:= RestrictedClassFunctions( diff, t11 );;
gap> dec11:= Decomposition( Irr( t11 ), rest11, "nonnegative" );;
gap> List( dec11, Set );
[ [ 0, 1 ], [ 0, 1 ] ]
gap> List( dec11, x -> Positions( x, 1 ) );
[ [ 40, 48 ], [ 39, 49 ] ]
gap> List( Irr( t11 ){ [ 40, 48 ] }, x -> x[1] );
[ 1575176, 3527824 ]
gap> List( Irr( t11 ){ [ 39, 49 ] }, x -> x[1] );
[ 1361919, 3741081 ]
```

For $p = 19$, the reduction of χ_{51} is $\varphi_{42} + \varphi_{45}$, with $\varphi_{42}(1) = 2\,125\,925$ and $\varphi_{45}(1) = 2\,977\,075$, and the reduction of χ_{52} is $\varphi_{33} + \varphi_{48}$, with $\varphi_{33}(1) = 1\,197\,330$ and $\varphi_{48}(1) = 3\,905\,670$.

```
gap> t19:= t mod 19;;
gap> rest19:= RestrictedClassFunctions( diff, t19 );;
gap> dec19:= Decomposition( Irr( t19 ), rest19, "nonnegative" );;
gap> List( dec19, Set );
[ [ 0, 1 ], [ 0, 1 ] ]
gap> List( dec19, x -> Positions( x, 1 ) );
[ [ 42, 45 ], [ 33, 48 ] ]
gap> List( Irr( t19 ){ [ 42, 45 ] }, x -> x[1] );
[ 2125925, 2977075 ]
gap> List( Irr( t19 ){ [ 33, 48 ] }, x -> x[1] );
[ 1197330, 3905670 ]
```

The Frobenius–Schur indicators of all involved p-modular constituents are $+$. This implies that χ_{51} reduces orthogonally stably modulo 11 but not orthogonally stably modulo 19, whereas χ_{52} reduces orthogonally stably modulo 19 but not orthogonally stably modulo 11.

```
──────────────────── Example ────────────────────
gap> Indicator( t11, 2 ){ [ 39, 40, 48, 49 ] };
[ 1, 1, 1, 1 ]
gap> Indicator( t19, 2 ){ [ 33, 42, 45, 48 ] };
[ 1, 1, 1, 1 ]
```

In version up to 1.3.4 of the character table library, this condition was not satisfied: The reduction of χ_{51} modulo both 11 and 19 was orthogonally stable, and the reduction of χ_{52} modulo both 11 and 19 was not orthogonally stable. However, this cannot happen, due to theoretical results about orthogonal discriminants of the involved characters. Thus we have found a way to decide the consistency of the classes 20A and 20B in characteristics 11 and 19: Either the 11-modular character table or the 19-modular character table of *HN* had to be changed for version 1.3.5, by swapping the classes 20A and 20B.

We decided to change the 11-modular table, because there are no other generality problems for *HN* involving the 19-modular table, and hence we are sure that this table will not need to be changed because of new solutions to generality problems.

For *HN*.2, the situation is similar. There are additionally two classes of element order 40 that have to be swapped if 20A and 20B get swapped. Thus we have to change the 11-modular table of *HN*.2 accordingly.

Changes of this kind may affect also derived character tables in the library. In this case, the 11-modular table with identifier `"(D10xHN).2"` was changed as well. Note that this table is not stored explicitly in the data files, it gets constructed from ordinary and modular library tables via `ConstructIndexTwoSubdirectProduct` (**CTblLib: ConstructIndexTwoSubdirectProduct**).

1.4   Brauer Tables that can be derived from Known Tables

In a few situations, one can derive the *p*-modular Brauer character table of a group from known character theoretic information.

For quite some time, a method is available in GAP that computes the Brauer characters of *p*-solvable groups (see (**Reference: BrauerTable**) and (**Reference: IsPSolubleCharacterTable**)).

The following sections list other situations where Brauer tables can be computed by GAP.

1.4.1   Brauer Tables via Construction Information

If a given ordinary character table *t*, say, has been constructed from other ordinary character tables then GAP may be able to create the *p*-modular Brauer table of *t* from the *p*-modular Brauer tables of the "ingredients". This happens currently in the following cases.

- *t* has been constructed with `CharacterTableDirectProduct` (**Reference: CharacterTableDirectProduct**), and GAP can compute the *p*-modular Brauer tables of the direct factors.

- *t* has been constructed with `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**), and GAP can compute the *p*-modular Brauer table of the table that is stored in *t* as the value of the attribute `SourceOfIsoclinicTable` (**Reference: SourceOfIsoclinicTable**).

- *t* has the attribute `ConstructionInfoCharacterTable` (**CTblLib: ConstructionInfoCharacterTable**) set, the first entry of this list *l*, say, is one of the strings `"ConstructGS3"` (see

2.3.2), `"ConstructIndexTwoSubdirectProduct"` (see 2.3.6), `"ConstructMGA"` (see 2.3.1), `"ConstructPermuted"`, `"ConstructV4G"` (see 2.3.4), and GAP can construct the p-modular Brauer table(s) of the relevant ordinary character table(s), which are library tables whose names occur in l.

1.4.2 Liftable Brauer Characters (May 2017)

Let B be a p-block of cyclic defect for the finite group G. It can be read off from the set $\mathrm{Irr}(B)$ of ordinary irreducible characters of B whether all irreducible Brauer characters in B are restrictions of ordinary characters to the p-regular classes of G, as follows.

If B has only one irreducible Brauer character then all ordinary characters in B restrict to this Brauer character. So let us assume that B contains at least two irreducible Brauer characters, and consider the set S, say, of restrictions of $\mathrm{Irr}(B)$ to the p-regular classes of G.

The block B contains exactly $|S| - 1$ irreducible Brauer characters, and the decomposition of the characters in S into these Brauer characters is described by an $|S|$ by $|S| - 1$ matrix M, say, whose entries are zero and one, such that exactly two nonzero entries occur in each column. (See for example [HL89, Theorem 2.1.5], which refers to [Dad66].)

If all irreducible Brauer characters of B occur in S then the matrix M contains $|S| - 1$ rows that contain exactly one nonzero entry, hence the remaining row consists only of 1s. This means that the element of largest degree in S is equal to the sum of all other elements in S. Conversely, if the element of largest degree in S is equal to the sum of all other elements in S then the matrix M has the structure as stated above, hence all irreducible Brauer characters of B occur in S.

Alternatively, one could state that all irreducible Brauer characters of B are restricted ordinary characters if and only if the Brauer tree of B is a *star* (see [HL89, p. 2]. If B contains at least two irreducible Brauer characters then this happens if and only if one of the types $\times$ or $\circ$ occurs for exactly one node in the Brauer graph of B, see [HL89, Lemma 2.1.13], and the distribution to types is determined by $\mathrm{Irr}(B)$.

The default method for `BrauerTableOp` (**Reference: BrauerTableOp**) that is contained in the GAP library has been extended in version 4.11 such that it checks whether the Sylow p-subgroups of the given group G are cyclic and, if yes, whether all p-blocks of G have the property discussed above. (This feature arose from a discussion with Klaus Lux.)

Examples where this method is successful for all blocks are the p-modular character tables of the groups $\mathrm{PSL}(2,q)$, where p is odd and does not divide q.

```
———————————————— Example ————————————————
gap> t:= CharacterTable( PSL( 2, 11 ) );;
gap> modt:= t mod 5;;
gap> modt <> fail;
true
gap> InfoText( modt );
"computed using that all Brauer characters lift to char. zero"
```

Another such example is the 5-modular table of the Mathieu group M_{11}.

```
———————————————— Example ————————————————
gap> lib:= CharacterTable( "M11" );;
gap> fromgroup:= CharacterTable( MathieuGroup( 11 ) );;
gap> DecompositionMatrix( lib mod 5 );
[ [ 1, 0, 0, 0, 0, 0, 0, 0, 0 ], [ 0, 1, 0, 0, 0, 0, 0, 0, 0 ],
  [ 0, 0, 1, 0, 0, 0, 0, 0, 0 ], [ 0, 0, 0, 1, 0, 0, 0, 0, 0 ],
```

```
     [ 0, 0, 0, 0, 1, 0, 0, 0, 0 ], [ 0, 0, 0, 0, 0, 1, 0, 0, 0 ],
     [ 0, 0, 0, 0, 0, 0, 1, 0, 0 ], [ 1, 0, 0, 0, 1, 1, 1, 0, 0 ],
     [ 0, 0, 0, 0, 0, 0, 0, 1, 0 ], [ 0, 0, 0, 0, 0, 0, 0, 0, 1 ] ]
gap> fromgroup mod 5 <> fail;
true
```

There are cases where all Brauer characters of a block lift to characteristic zero but the defect group of the block is not cyclic, thus the method cannot be used. An example is the 2-modular table of the Mathieu group M_{11}.

```
─────────────────────────── Example ───────────────────────────
gap> DecompositionMatrix( lib mod 2 );
[ [ 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> fromgroup mod 2;
fail
```

1.5   Information about certain subgroups of the Monster group

1.5.1   The Monster group does not contain subgroups of the type $2.U_4(2)$ (August 2023)

In the context of a question about decomposition numbers of the sporadic simple Monster group $\mathbb{M}$, Benjamin Sambale was interested in possible embeddings of certain groups G into $\mathbb{M}$ such that the decomposition matrices of G are known. For a given G, the first steps were to compute the possible class fusions of G in $\mathbb{M}$ and then to check whether the corresponding embeddings would be interesting.

Apparently, calling `PossibleClassFusions` (**Reference: PossibleClassFusions**) with its default parameters often runs very long and requires a lot of space when G is a small group such as $2.U_4(2)$. We can do better by calling the function with the parameter `decompose:= false`. This has the effect that one criterion is omitted that checks the decomposability of restricted characters of $\mathbb{M}$ as an integral linear combination of characters of the subgroup. As a rule of thumb, if the number of classes of the subgroup is small compared to the number of classes of the group and if the result consists of many candidates then it might be faster to omit the decomposability criterion.

```
─────────────────────────── Example ───────────────────────────
gap> s:= CharacterTable( "2.U4(2)" );;
gap> m:= CharacterTable( "M" );;
gap> sfusm:= PossibleClassFusions( s, m, rec( decompose:= false ) );;
gap> Length( sfusm );
2332
```

Looking at the (many) candidates, we see that all map the central involution of $2.U_4(2)$ to the class 2B of $\mathbb{M}$, thus any subgroup of the type $2.U_4(2)$ lies inside the 2B normalizer in $\mathbb{M}$. We compute the possible class fusions into this subgroup.

```
─────────────────────────── Example ───────────────────────────
gap> Set( List( sfusm, x -> x[2] ) );
[ 3 ]
gap> t:= CharacterTable( "MN2B" );
CharacterTable( "2^1+24.Co1" )
```

```
gap> sfust:= PossibleClassFusions( s, t, rec( decompose:= false ) );;
gap> Length( sfust );
0
```

Thus we have shown that $\mathbb{M}$ does not contain subgroups of the type $2.U_4(2)$.

1.5.2 Perfect central extensions of $L_3(4)$ (August 2023)

There was the question in MathOverflow which perfect central extensions of the simple group $G = L_3(4)$ are subgroups of the sporadic simple Monster group $\mathbb{M}$.

First we get the list of perfect central extensions of G (asuming that their character tables are contained in the character table library).

```
                              Example
  gap> simp:= CharacterTable( "L3(4)" );;
  gap> extnames:= AllCharacterTableNames( Identifier,
  >                    x -> EndsWith(x, "L3(4)" ) );;
  gap> ext:= List( extnames, CharacterTable );;
  gap> ext:= Filtered( ext, x -> Length( ClassPositionsOfCentre( x ) ) =
  >                              Size( x ) / Size( simp ) );;
  gap> SortBy( ext, Size );
  gap> names:= List( ext, Identifier );
  [ "L3(4)", "2.L3(4)", "3.L3(4)", "2^2.L3(4)", "4_1.L3(4)",
    "4_2.L3(4)", "6.L3(4)", "(2x4).L3(4)", "(2^2x3).L3(4)",
    "12_1.L3(4)", "12_2.L3(4)", "4^2.L3(4)", "(2x12).L3(4)",
    "(4^2x3).L3(4)" ]
```

The fact that G is *not* isomorphic to a subgroup of $\mathbb{M}$ is shown in [HW08] (at the end of this paper).

And the following embeddings of central extensions of G in $\mathbb{M}$ can be established using known subgroups of $\mathbb{M}$.

- $2.G < 2.U_4(3) < 2^2.U_6(2) < Fi_{23} < 3.Fi'_{24} < \mathbb{M}$.

- $2^2.G < He < 3.Fi'_{24} < \mathbb{M}$.

- $6.G < 2.G_2(4) < 6.Suz < 3^{1+12}_+.2Suz < \mathbb{M}$.

Note that G is a subgroup of $U_4(3)$ but not of $2.U_4(3)$, $3.G$ is a subgroup of $G_2(4)$ but not of $2.G_2(4)$, and $G_2(4)$ is a subgroup of *Suz* but not of $2.Suz$. The positive statements follow from [CCN+85, pp. 52, 97, 131] and the negative ones from the following computations.

```
                              Example
  gap> Length( PossibleClassFusions( CharacterTable( "L3(4)" ),
  >                                  CharacterTable( "2.U4(3)" ) ) );
  0
  gap> Length( PossibleClassFusions( CharacterTable( "3.L3(4)" ),
  >                                  CharacterTable( "2.G2(4)" ) ) );
  0
  gap> Length( PossibleClassFusions( CharacterTable( "G2(4)" ),
  >                                  CharacterTable( "2.Suz" ) ) );
  0
```

The group $3.G$ centralizes an element of order three. If $3.G$ is a subgroup of $\mathbb{M}$ then it is contained in a 3A centralizer (of the structure $3.Fi'_{24}$), a 3B centralizer (of the structure $3^{1+12}_+.2Suz$) or a 3C centralizer (of the structure $3 \times Th$). Clearly the case $3C$ cannot occur, and 3B is excluded by the fact that no class fusion between $3.G$ and the $3B$ normalizer $3^{1+12}_+.2Suz.2$ is possible.

```
————————————— Example —————————————
  gap> t:= CharacterTable( "MN3B" );
  CharacterTable( "3^(1+12).2.Suz.2" )
  gap> Length( PossibleClassFusions( CharacterTable( "3.L3(4)" ), t ) );
  0
```

If $3.G$ is contained in the 3A centralizer then this embedding induces one of G into some maximal subgroup of Fi'_{24}. Using the known character tables of these maximal subgroups in GAP's character table library, one shows that only Fi_{23} admits a class fusion, but this subgroup lifts to $3 \times Fi_{23}$ in $3.Fi'_{24}$ and thus cannot lead to a subgroup of type $3.G.$.

```
————————————— Example —————————————
  gap> mx:= List( Maxes( CharacterTable( "Fi24'" ) ), CharacterTable );;
  gap> s:= CharacterTable( "L3(4)" );;
  gap> Filtered( mx, x -> Length( PossibleClassFusions( s, x ) ) > 0 );
  [ CharacterTable( "Fi23" ) ]
```

The other candidates $m.G$ contain at least one central involution. If $m.G$ is a subgroup of $\mathbb{M}$ then it is contained in a 2A centralizer (of the structure $2.B$) or a 2B centralizer (of the structure 2^{1+24}_+.Co_1). Again we use PossibleClassFusions (**Reference: PossibleClassFusions**) to list all candidates for the class fusion, but here we prescribe the central involution of the 2A or 2B centralizer as an image of one central involution in $m.G$.

```
————————————— Example —————————————
  gap> done:= [ "L3(4)", "2.L3(4)", "3.L3(4)", "2^2.L3(4)", "6.L3(4)" ];;
  gap> names:= Filtered( names, x -> not x in done );
  [ "4_1.L3(4)", "4_2.L3(4)", "(2x4).L3(4)", "(2^2x3).L3(4)",
    "12_1.L3(4)", "12_2.L3(4)", "4^2.L3(4)", "(2x12).L3(4)",
    "(4^2x3).L3(4)" ]
  gap> invcent:= List( [ "MN2A", "MN2B" ], CharacterTable );
  [ CharacterTable( "2.B" ), CharacterTable( "2^1+24.Co1" ) ]
  gap> ForAll( invcent, x -> ClassPositionsOfCentre( x ) = [ 1, 2 ] );
  true
  gap> cand:= [];;
  gap> ords:= "dummy";;   #  Avoid a message about an unbound variable ...
  gap> for name in names do
  >       s:= CharacterTable( name );
  >       ords:= OrdersClassRepresentatives( s );
  >       invpos:= Filtered( ClassPositionsOfCentre( s ), i -> ords[i] = 2 );
  >       for i in invpos do
  >         for t in invcent do
  >           init:= InitFusion( s, t );
  >           if init = fail then
  >             continue;
  >           fi;
  >           init[i]:= 2;
  >           fus:= PossibleClassFusions( s, t, rec( fusionmap:= init,
  >                                                  decompose:= false ) );
```

```
>            if fus <> [] then
>              Add( cand, [ s, t, i, fus ] );
>            fi;
>          od;
>        od;
>      od;
gap> List( cand, x -> x{ [ 1 .. 3 ] } );
[ [ CharacterTable( "4_1.L3(4)" ), CharacterTable( "2^1+24.Co1" ), 3 ]
    , [ CharacterTable( "(2x4).L3(4)" ), CharacterTable( "2.B" ), 2 ],
  [ CharacterTable( "(2x4).L3(4)" ), CharacterTable( "2.B" ), 3 ] ]
```

(Note that we have called `PossibleClassFusions` (**Reference: PossibleClassFusions**) with the option `decompose:= false`, in order to save space and time. See Section 1.5.1 for more details.)

Concerning the candidate $(2 \times 4).G$, we see that only fusions are possible for which the central involution in question is mapped to a 2A element of $\mathbb{M}$. Since we get candidates only for two out of the three central involutions, we see that $(2 \times 4).G$ does not embed into $\mathbb{M}$.

Thus it turns out that exactly one group $m.G$ cannot be excluded this way. Namely, these character-theoretical criteria leave the possibility that $4_1.G$ may occur as a subgroup of $2_+^{1+24}.Co_1$.

Moreover, we see that if this happens then the centre C of $4_1.G$ lies inside the normal subgroup $N = 2_+^{1+24}$. The centralizer of C in N has order 2^{24}, and the centralizer of C in $2_+^{1+24}.Co_1$ has order $2^{24} \cdot |Co_3|$. We see that $4_1.G$, if it exists as a subgroup of $2_+^{1+24}.Co_1$, must lie inside the subgroup $[2^{24}].Co_3$.

```
———————————————————— Example ————————————————————
gap> s:= cand[1][1];
CharacterTable( "4_1.L3(4)" )
gap> t:= cand[1][2];
CharacterTable( "2^1+24.Co1" )
gap> fus:= cand[1][4];
[ [ 1, 5, 2, 5, 9, 8, 23, 27, 24, 27, 49, 49, 50, 50, 70, 74, 71, 74,
      70, 74, 71, 74, 114, 119, 115, 119, 114, 119, 115, 119 ] ]
gap> ClassPositionsOfCentre( s );
[ 1, 2, 3, 4 ]
gap> 5 in ClassPositionsOfPCore( t, 2 );
true
gap> siz:= SizesCentralizers( t )[5] / 2^24;
495766656000
gap> mx:= Filtered( List( Maxes( CharacterTable( "Co1" ) ) ),
>                     CharacterTable ),
>              x -> Size( x ) mod siz = 0 );
[ CharacterTable( "Co3" ) ]
gap> Size( mx[1] ) = siz;
true
```

I do not see a character–theoretic argument that could disprove the existence of such an $4_1.G$ type subgroup.

1.5.3   The character table of $(2 \times O_8^+(3)).S_4 \leq 2.B$ (October 2023)

Consider a maximal subgroup H of type $(3^2 : 2 \times O_8^+(3)).S_4$ in the sporadic simple Monster group. The character table of H has been contributed by Tim Burness. We can view H as $O_8^+(3).(3^2 : 2S_4) =$

$O_8^+(3).F$. The character table of H determines that of F, and this table determines the isomorphism type of F as `SmallGroup( 432, 734 )`.

```
──────────────── Example ────────────────
  gap> tblH:= CharacterTable( "(3^2:2xO8+(3)).S4" );
  CharacterTable( "(3^2:2xO8+(3)).S4" )
  gap> N:= ClassPositionsOfSolvableResiduum( tblH );;
  gap> tblF:= tblH / N;;
  gap> Size( tblF );
  432
  gap> known:= NamesOfEquivalentLibraryCharacterTables( tblF );
  [ "3^2.2.S4", "M12M7" ]
  gap> Filtered( GroupInfoForCharacterTable( known[1] ),
  >              x -> x[1] = "SmallGroup" );
  [ [ "SmallGroup", [ 432, 734 ] ] ]
```

(Note that the precomputed `GroupInfoForCharacterTable` (**CTblLib: GroupInfoForCharacterTable**) information about GAP library character tables means that exactly one isomorphism type of groups fits to the character table of F.)

We compute that $O_3(F) \cong 3^2$ has complements in F, thus H has a subgroup V of the type $O_8^+(3).2S_4$, which is a complement of $O_3(H)$ in H, thus V is isomorphic with $H/O_3(H)$.

```
──────────────── Example ────────────────
  gap> G:= SmallGroup( 432, 734 );;
  gap> P:= PCore( G, 3 );;
  gap> Length( ComplementClassesRepresentatives( G, P ) );
  1
```

We can derive the character table of V from that of H, and compute that the group structure of V is $(2 \times O_8^+(3)).S_4$. For that, we consider the element orders of the unique normal subgroup of order $2|O_8^+(3)|$ in V. If this normal subgroup would not be isomorphic with $2 \times O_8^+(3)$ then it would have one of the structures $O_8^+(3).2_1$ or $O_8^+(3).2_2$, but then it would contain elements of the orders 24.

```
──────────────── Example ────────────────
  gap> tblV:= tblH / ClassPositionsOfPCore( tblH, 3 );;
  gap> ord:= 2 * Size( tblH ) / Size( tblF );
  9904359628800
  gap> classes:= SizesConjugacyClasses( tblV );;
  gap> 2N:= Filtered( ClassPositionsOfNormalSubgroups( tblV ),
  >                   l -> Sum( classes{ l } ) = ord );;
  gap> Length( 2N );
  1
  gap> Set( OrdersClassRepresentatives( tblV ){ 2N[1] } );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 20, 26, 30 ]
  gap> Set( OrdersClassRepresentatives( CharacterTable( "O8+(3)" ) ) );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 20 ]
  gap> Set( OrdersClassRepresentatives( CharacterTable( "O8+(3).2_1" ) ) );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 20, 24, 26, 28,
    30, 36, 40 ]
  gap> Set( OrdersClassRepresentatives( CharacterTable( "O8+(3).2_2" ) ) );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 20, 24, 26, 28,
    30, 36 ]
```

The class fusion of *V* into the Monster group shows that *V* centralizes a 2A element in the Monster, hence *V* is a subgroup of a maximal subgroup of the type 2.*B*.

```
 ──────────────────── Example ────────────────────
  gap> tblM:= CharacterTable( "M" );;
  gap> VfusM:= PossibleClassFusions( tblV, tblM );;
  gap> Length( VfusM );
  4
  gap> ZV:= ClassPositionsOfCentre( tblV );
  [ 1, 2 ]
  gap> Set( List( VfusM, l -> l{ ZV } ) );
  [ [ 1, 2 ] ]
```

From the list of maximal subgroups of *B*, we see that either *V* is contained in the preimage of Fi_{23} under the natural epimorphism from 2.*B* to *B*, or *V* is equal to the preimage of $O_8^+(3).S_4$.

```
 ──────────────────── Example ────────────────────
  gap> tblB:= CharacterTable( "B" );;
  gap> mxB:= List( Maxes( tblB ), CharacterTable );;
  gap> cand:= Filtered( mxB, s -> Size( s ) mod ( Size( tblV ) / 2 ) = 0 );
  [ CharacterTable( "Fi23" ), CharacterTable( "O8+(3).S4" ) ]
```

The former possibility is excluded from the fact that the factor of *V* by its center does not admit a class fusion into Fi_{23}.

```
 ──────────────────── Example ────────────────────
  gap> Length( PossibleClassFusions( tblV / ZV, CharacterTable( "Fi23" ) ) );
  0
```

We conclude that *V* is a maximal subgroup of 2.*B*.

Thus we have used the character table of *H* to construct the character table of a maximal subgroup of 2.*B*, with very little effort.

This table is meanwhile available in the table library, with the identifier `"(2x08+(3)).S4"`.

```
 ──────────────────── Example ────────────────────
  gap> lib:= CharacterTable( "(2x08+(3)).S4" );;
  gap> TransformingPermutationsCharacterTables( tblV, lib ) <> fail;
  true
  gap> Irr( lib ) = Irr( tblV );
  true
```

In order to add the table to the library, we have to provide also the class fusions from *V* to 2.*B*, to the maximal subgroup $(2 \times O_8^+(3)).S_4$ of *B*, and to the maximal subgroup $(3^2 : 2 \times O_8^+(3)).S_4$ of *M*, such that the compositions of fusions from *V* to *B* via $O_8^+(3).S_4$ and 2.*B* are compatible, . . .

```
 ──────────────────── Example ────────────────────
  gap> tblU:= CharacterTable( "O8+(3).S4" );;
  gap> tbl2B:= CharacterTable( "2.B" );;
  gap> CompositionMaps( GetFusionMap( tblU, tblB ),
  >                     GetFusionMap( lib, tblU ) ) =
  >    CompositionMaps( GetFusionMap( tbl2B, tblB ),
  >                     GetFusionMap( lib, tbl2B ) );
  true
```

... and that the compositions of fusions from V to M via $(3^2 : 2 \times O_8^+(3)).S_4$ and $2.B$ are compatible.

```
 ─────────────────────────────── Example ───────────────────────────────
 gap> tblH:= CharacterTable( "(3^2:2xO8+(3)).S4" );;
 gap> CompositionMaps( GetFusionMap( tblH, tblM ),
 >                     GetFusionMap( lib, tblH ) ) =
 >    CompositionMaps( GetFusionMap( tbl2B, tblM ),
 >                     GetFusionMap( lib, tbl2B ) );
 true
```

Chapter 2

Using Table Automorphisms for Constructing Character Tables in **GAP**

Date: June 27th, 2004

This chapter has three aims. First it shows how character table automorphisms can be utilized to construct certain character tables from others using the **GAP** system [GAP24]; the **GAP** functions used for that are part of the **GAP** Character Table Library [Bre25]. Second it documents several constructions of character tables which are contained in the **GAP** Character Table Library. Third it serves as a testfile for the involved **GAP** functions.

2.1 Overview

Several types of constructions of character tables of finite groups from known tables of smaller groups are described in Section 2.3. Selecting suitable character table automorphisms is an important ingredient of these constructions.

Section 2.2 collects the few representation theoretical facts on which these constructions are based.

The remaining sections show examples of the constructions in **GAP**. These examples use the **GAP** Character Table Library, therefore we load this package first.

```
——————————————— Example ———————————————
 gap> LoadPackage( "ctbllib", "1.1.4", false );
 true
```

2.2 Theoretical Background

2.2.1 Character Table Automorphisms

Let G be a finite group, $\mathrm{Irr}(G)$ be the matrix of ordinary irreducible characters of G, $Cl(G)$ be the set of conjugacy classes of elements in G, g^G the G-conjugacy class of $g \in G$, and let

$$pow_p\colon Cl(G) \to Cl(G), g^G \mapsto (g^p)^G$$

be the p-th power map, for each prime integer p.

A *table automorphism* of G is a permutation $\sigma\colon Cl(G) \to Cl(G)$ with the properties that $\chi \circ \sigma \in \mathrm{Irr}(G)$ holds for all $\chi \in \mathrm{Irr}(G)$ and that σ commutes with pow_p, for all prime integers p that divide

51

the order of G. Note that for prime integers p that are coprime to the order of G, pow_p commutes with each σ that permutes $\mathrm{Irr}(G)$, since pow_p acts as a field automorphism on the character values.

In GAP, a character table covers the irreducible characters –a matrix M of character values– as well as the power maps of the underlying group –each power map pow_p being represented as a list pow_p' of positive integers denoting the positions of the image classes. The group of table automorphisms of a character table is represented as a permutation group on the column positions of the table; it can be computed with the GAP function `AutomorphismsOfTable` (**Reference: Automorphism-sOfTable**).

In the following, we will mainly use that each *group automorphism* σ of G induces a table automorphism that maps the class of each element in G to the class of its image under σ.

2.2.2 Permutation Equivalence of Character Tables

Two character tables with matrices M_1, M_2 of irreducibles and p-th power maps $pow_{1,p}$, $pow_{2,p}$ are *permutation equivalent* if permutations ψ and π of row and column positions of the M_i exist such that $[M_1]_{i,j} = [M_2]_{i\psi, j\pi}$ holds for all indices i, j, and such that $\pi \cdot pow_{2,p}' = pow_{1,p}' \cdot \pi$ holds for all primes p that divide the (common) group order. The first condition is equivalent to the existence of a permutation π such that permuting the columns of M_1 with π maps the set of rows of M_1 to the set of rows of M_2.

π is of course determined only up to table automorphisms of the two character tables, that is, two transforming permutations π_1, π_2 satisfy that $\pi_1 \cdot \pi_2^{-1}$ is a table automorphism of the first table, and $\pi_1^{-1} \cdot \pi_2$ is a table automorphism of the second.

Clearly two isomorphic groups have permutation equivalent character tables.

The GAP library function `TransformingPermutationsCharacterTables` (**Reference: TransformingPermutationsCharacterTables**) returns a record that contains transforming permutations of rows and columns if the two argument tables are permutation equivalent, and `fail` otherwise.

In the example sections, the following function for computing representatives from a list of character tables w.r.t. permutation equivalence will be used. More precisely, the input is either a list of character tables or a list of records which have a component `table` whose value is a character table, and the output is a sublist of the input.

```
───────────────────────── Example ─────────────────────────
gap> RepresentativesCharacterTables:= function( list )
>     local reps, entry, r;
>
>     reps:= [];
>     for entry in list do
>       if ForAll( reps, r -> ( IsCharacterTable( r ) and
>             TransformingPermutationsCharacterTables( entry, r ) = fail )
>           or ( IsRecord( r ) and TransformingPermutationsCharacterTables(
>                                 entry.table, r.table ) = fail ) ) then
>         Add( reps, entry );
>       fi;
>     od;
>     return reps;
>     end;;
```

2.2.3 Class Fusions

For two groups H, G such that H is isomorphic with a subgroup of G, any embedding $\iota\colon H \to G$ induces a class function

$$fus_\iota\colon Cl(H) \to Cl(G), h^G \mapsto (\iota(h))^G$$

the *class fusion* of H in G via ι. Analogously, for a normal subgroup N of G, any epimorphism $\pi\colon G \to G/N$ induces a class function

$$fus_\pi\colon Cl(G) \to Cl(G/N), g^G \mapsto (\pi(g))^G$$

the *class fusion* of G onto G/N via π.

When one works only with character tables and not with groups, these class fusions are the objects that describe subgroup and factor group relations between character tables. Technically, class fusions are necessary for restricting, inducing, and inflating characters from one character table to another. If one is faced with the problem to compute the class fusion between the character tables of two groups H and G for which it is known that H can be embedded into G then one can use character-theoretic necessary conditions, concerning that the restriction of all irreducible characters of G to H (via the class fusion) must decompose into the irreducible characters of H, and that the class fusion must commute with the power maps of H and G.

With this character-theoretic approach, one can clearly determine possible class fusions only up to character table automorphisms. Note that one can interpret each character table automorphism of G as a class fusion from the table of G to itself.

If N is a normal subgroup in G then the class fusion of N in G determines the orbits of the conjugation action of G on the classes of N. Often the knowledge of these orbits suffices to identify the subgroup of table automorphisms of N that corresponds to this action of G; for example, this is always the case if N has index 2 in G.

GAP library functions for dealing with class fusions, power maps, and character table automorphisms are described in the chapter "Maps Concerning Character Tables" in the GAP Reference Manual.

2.2.4 Constructing Character Tables of Certain Isoclinic Groups

As is stated in [CCN$^+$85, p. xxiii], two groups G, H are called *isoclinic* if they can be embedded into a group K such that K is generated by $Z(K)$ and G, and also by $Z(K)$ and H. In the following, two special cases of isoclinism will be used, where the character tables of the isoclinic groups are closely related.

(1) $G \cong 2 \times U$ for a group U that has a central subgroup N of order 2, and H is the central product of U and a cyclic group of order four. Here we can set $K = 2 \times H$.

(2) $G \cong 2 \times U$ for a group U that has a normal subgroup N of index 2, and H is the subdirect product of U and a cyclic group of order four, Here we can set $K = 4 \times U$.
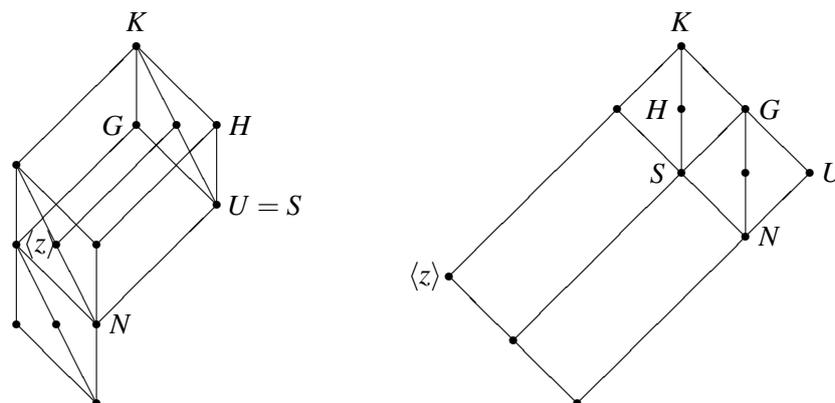
Starting from the group K containing both G and H, we first note that each irreducible representation of G or H extends to K. More specifically, if ρ_G is an irreducible representation of G then we can define an extension ρ of K by defining it suitably on $Z(K)$ and then form ρ_H, the restriction of ρ to H.

In our two cases, we set $S = G \cap H$, so $K = S \cup G \setminus S \cup H \setminus S \cup zS$ holds for some element $z \in Z(K) \setminus (G \cup H)$ of order four, and $G = S \cup gS$ for some $g \in G \setminus S$, and $H = S \cup hS$ where $h = z \cdot g \in H \setminus S$. For defining ρ_H, it suffices to consider $\rho(h) = \rho(z)\rho(g)$, where $\rho(z) = \varepsilon_\rho(z) \cdot I$ is a scalar matrix.

As for the character table heads of G and H, we have $s^G = s^H$ and $z(g \cdot s)^G = (h \cdot s)^H$ for each $s \in S$, so this defines a bijection of the conjugacy classes of G and H. For a prime integer p, $(h \cdot s)^p = (z \cdot g \cdot s)^p = z^p \cdot (g \cdot s)^p$ holds for all $s \in S$, so the p-th power maps of G and H are related as follows: Inside S they coincide for any p. If $p \equiv 1 \bmod 4$ they coincide also outside S, if $p \equiv -1 \bmod 4$ the images differ by exchanging the classes of $(h \cdot s)^p$ and $z^2 \cdot (h \cdot s)^p$ (if these elements lie in different classes), and for $p = 2$ the images (which lie inside S) differ by exchanging the classes of $(h \cdot s)^2$ and $z^2 \cdot (g \cdot s)^2$ (if these elements lie inside S).

Let ρ be an irreducible representation of K. Then ρ_G and ρ_H are related as follows: $\rho_G(s) = \rho_H(s)$ and $\rho(z) \cdot \rho_G(g \cdot s) = \rho_H(h \cdot s)$ for all $s \in S$. If χ_G and χ_H are the characters afforded by ρ_G and ρ_H, respectively, then $\chi_G(s) = \chi_H(s)$ and $\varepsilon_\rho(z) \cdot \chi_G(g \cdot s) = \chi_H(h \cdot s)$ hold for all $s \in S$. In the case $\chi_G(z^2) = \chi(1)$ we have $\varepsilon_\rho(z) = \pm 1$, and both cases actually occur if one considers all irreducible representations of K. In the case $\chi_G(z^2) = -\chi(1)$ we have $\varepsilon_\rho(z) = \pm i$, and again both cases occur. So we obtain the irreducible characters of H from those of G by multiplying the values outside S in all those characters by i that do not have z^2 in their kernels.

In GAP, the function `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) can be used for computing the character table of H from that of G, and vice versa. (Note that in the above two cases, also the groups U and H are isoclinic by definition, but `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) does not transfer the character table of U to that of H.)

One could construct the character tables mentioned above by forming the character tables of certain factor groups or normal subgroups of direct products. However, the construction via `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) has the advantage that the result stores from which sources it arose, and this information can be used to derive also the Brauer character tables, provided that the Brauer character tables of the source tables are known.

2.2.5   Character Tables of Isoclinic Groups of the Structure $p.G.p$ (October 2016)

Since the release of GAP 4.11, `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) admits the construction of the character tables of the isoclinic variants of groups of the structure $p.G.p$,

also for odd primes p.

This feature will be used in the construction of the character table of $9.U_3(8).3_3$, in order to construct the table of the subgroup $3.(3 \times U_3(8))$ and of the factor group $(3 \times U_3(8)).3_3$, see Section 2.4.16. These constructions are a straightforward generalization of those described in detail in Section 2.2.4.

There are several examples of Atlas groups of the structure $3.G.3$. The character table of one such group is shown in the Atlas, the tables of their isoclinic variants can now be obtained from `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**).

For example, the group $GL(3,4)$ has the structure $3.L_3(4).3$. There are three pairwise nonisomorphic isoclinic variants of groups of this structure.

```
                        ─── Example ───
 gap> t:= CharacterTable( "3.L3(4).3" );
 CharacterTable( "3.L3(4).3" )
 gap> iso1:= CharacterTableIsoclinic( t );
 CharacterTable( "Isoclinic(3.L3(4).3,1)" )
 gap> iso2:= CharacterTableIsoclinic( t, rec( k:= 2 ) );
 CharacterTable( "Isoclinic(3.L3(4).3,2)" )
 gap> TransformingPermutationsCharacterTables( t, iso1 );
 fail
 gap> TransformingPermutationsCharacterTables( t, iso2 );
 fail
 gap> TransformingPermutationsCharacterTables( iso1, iso2 );
 fail
```

The character table of $GL(3,4)$ is in fact the one which is shown in the Atlas.

```
                        ─── Example ───
 gap> IsRecord( TransformingPermutationsCharacterTables( t,
 >                 CharacterTable( GL( 3, 4 ) ) ) );
 true
```

2.2.6 Isoclinic Double Covers of Almost Simple Groups

The function `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) can also be used to switch between the character tables of double covers of groups of the type $G.2$, where G is a perfect group, see [CCN+85, Section 6.7]. Typical examples are the double covers of symmetric groups.

Note that these double covers may be isomorphic. This happens for $2.S_6$. More generally, this happens for all semilinear groups $\Sigma L(2, p^2)$, for odd primes p. The smallest examples are $\Sigma L(2,9) = 2.A_6.2_1$ and $\Sigma L(2,25) = 2.L_2(25).2_2$. This implies that the character table and its isoclinic variant are permutation isomorphic.

```
                        ─── Example ───
 gap> t:= CharacterTable( "2.A6.2_1" );
 CharacterTable( "2.A6.2_1" )
 gap> TransformingPermutationsCharacterTables( t,
 >         CharacterTableIsoclinic( t ) );
 rec( columns := (4,6)(5,7)(11,12)(14,16)(15,17),
   group := Group([ (16,17), (14,15) ]),
   rows := (3,5)(4,6)(10,11)(12,15,13,14) )
 gap> t:= CharacterTable( "2.L2(25).2_2" );
 CharacterTable( "2.L2(25).2_2" )
 gap> TransformingPermutationsCharacterTables( t,
```

```
>          CharacterTableIsoclinic( t ) );
rec( columns := (7,9)(8,10)(20,21)(23,24)(25,27)(26,28),
  group := <permutation group with 4 generators>,
  rows := (3,5)(4,6)(14,15)(16,17)(19,22,20,21) )
```

For groups of the type $4.G.2$, two different situations can occur. Either the distinguished central cyclic subgroup of order four in $4.G$ is inverted by the elements in $4.G.2 \setminus 4.G$, or this subgroup is central in $4.G.2$. In the first case, calling `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) with the character table of $4.G.2$ yields a character table with the same set of irreducibles, only the 2-power map will in general differ from that of the input table. In the second case, the one argument version of `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) returns a permutation isomorphic table. By supplying additional arguments, there is a chance to construct tables of different groups.

We demonstrate this phenomenon with the various groups of the structure $4.L_3(4).2$.

```
———————————————————— Example ————————————————————
gap> tbls:= [];;
gap> for m in [ "4_1", "4_2" ] do
>      for a in [ "2_1", "2_2", "2_3" ] do
>        Add( tbls, CharacterTable( Concatenation( m, ".L3(4).", a ) ) );
>      od;
>    od;
gap> tbls;
[ CharacterTable( "4_1.L3(4).2_1" ), CharacterTable( "4_1.L3(4).2_2" )
    , CharacterTable( "4_1.L3(4).2_3" ),
  CharacterTable( "4_2.L3(4).2_1" ), CharacterTable( "4_2.L3(4).2_2" )
    , CharacterTable( "4_2.L3(4).2_3" ) ]
gap> case1:= Filtered( tbls, t -> Size( ClassPositionsOfCentre( t ) ) = 2 );
[ CharacterTable( "4_1.L3(4).2_1" ), CharacterTable( "4_1.L3(4).2_2" )
    , CharacterTable( "4_2.L3(4).2_1" ),
  CharacterTable( "4_2.L3(4).2_3" ) ]
gap> case2:= Filtered( tbls, t -> Size( ClassPositionsOfCentre( t ) ) = 4 );
[ CharacterTable( "4_1.L3(4).2_3" ),
  CharacterTable( "4_2.L3(4).2_2" ) ]
```

The centres of the groups $4_1.L_3(4).2_1$, $4_1.L_3(4).2_2$, $4_2.L_3(4).2_1$, and $4_2.L_3(4).2_3$ have order two, that is, these groups belong to the first case. Each of these groups is not permutation equivalent to its isoclinic variant but has the same irreducible characters.

```
———————————————————— Example ————————————————————
gap> isos1:= List( case1, CharacterTableIsoclinic );;
gap> List( [ 1 .. 4 ], i -> Irr( case1[i] ) = Irr( isos1[i] ) );
[ true, true, true, true ]
gap> List( [ 1 .. 4 ],
>      i -> TransformingPermutationsCharacterTables( case1[i], isos1[i] ) );
[ fail, fail, fail, fail ]
```

The groups $4_1.L_3(4).2_3$ and $4_2.L_3(4).2_2$ belong to the second case because their centres have order four.

```
———————————————————— Example ————————————————————
gap> isos2:= List( case2, CharacterTableIsoclinic );;
gap> List( [ 1, 2 ],
```

```
>        i -> TransformingPermutationsCharacterTables( case2[i], isos2[i] ) );
[ rec( columns := (26,27,28,29)(30,31,32,33)(38,39,40,41)(42,43,44,45)
        , group := <permutation group with 5 generators>,
     rows := (16,17)(18,19)(20,21)(22,23)(28,29)(32,33)(36,37)(40,
        41) ),
  rec( columns := (28,29,30,31)(32,33)(34,35,36,37)(38,39,40,41)(42,
        43,44,45)(46,47,48,49), group := <permutation group with
        3 generators>, rows := (15,16)(17,18)(20,21)(22,23)(24,25)(26,
        27)(28,29)(34,35)(38,39)(42,43)(46,47) ) ]
gap> isos3:= List( case2, t -> CharacterTableIsoclinic( t,
>                         ClassPositionsOfCentre( t ) ) );;
gap> List( [ 1, 2 ],
>        i -> TransformingPermutationsCharacterTables( case2[i], isos3[i] ) );
[ fail, fail ]
```

2.2.7 Characters of Normal Subgroups

Let G be a group and N be a normal subgroup of G. We will need the following well–known facts about the relation between the irreducible characters of G and N.

For an irreducible (Brauer) character χ of N and $g \in G$, we define χ^g by $\chi^g(n) = \chi(n^g)$ for all $n \in N$, and set $I_G(\chi) = \{g \in G; \chi^g = \chi\}$ (see [Fei82, p. 86]).

If $I_G(\chi) = N$ then the induced character χ^G is an irreducible (Brauer) character of G (see [Fei82, Lemma III 2.11] or [Nav98, Theorem 8.9] or [LP10, Corollary 4.3.8]).

If G/N is cyclic and if $I_G(\chi) = G$ then $\chi = \psi_N$ for an irreducible (Brauer) character ψ of G, and each irreducible (Brauer) character θ with the property $\chi = \theta_N$ is of the form $\theta = \psi \cdot \varepsilon$, where ε is an irreducible (Brauer) character of G/N (see [Fei82, Theorem III 2.14] or [Nav98, Theorem 8.12] or [LP10, Theorem 3.6.13]).

Clifford's theorem ([Fei82, Theorem III 2.12] or [Nav98, Corollary 8.7] or [LP10, Theorem 3.6.2]) states that the restriction of an irreducible (Brauer) character of G to N has the form $e\sum_{i=1}^{t} \varphi_i$ for a positive integer e and irreducible (Brauer) characters φ_i of N, where t is the index of $I_G(\varphi_1)$ in G.

Now assume that G is a normal subgroup in a larger group H, that G/N is an abelian chief factor of H and that ψ is an ordinary irreducible character of G such that $I_H(\psi) = H$. Then either $t = 1$ and e^2 is one of 1, $|G/N|$, or $t = |G/N|$ and $e = 1$ (see [Isa76, Theorem 6.18]).

2.3 The Constructions

2.3.1 Character Tables of Groups of the Structure $M.G.A$

(This kind of table construction is described in [Bre11].)

Let N denote a downward extension of the finite group G by a finite group M, let H denote an automorphic (upward) extension of N by a finite cyclic group A such that M is normal in H, and set $F = H/M$. We consider the situation that each irreducible character of N that does not contain M in its kernel induces irreducibly to H. Equivalently, the action of $A = \langle a \rangle$ on the characters of N, via $\chi \mapsto \chi^a$, has only orbits of length exactly $|A|$ on the set $\{\chi \in \mathrm{Irr}(N); M \nsubseteq \ker(\chi)\}$.
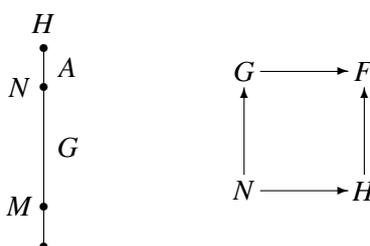
This occurs for example if M is central in N and A acts fixed-point freely on M, we have $|M| \equiv 1 \bmod |A|$ in this case. If M has prime order then it is sufficient that A does not centralize M.

The ordinary (or p-modular) irreducible characters of H are then given by the ordinary (or p-modular) irreducible characters of F and N, the class fusions from the table of N onto the table of G and from the table of G into that of F, and the permutation π that is induced by the action of A on the conjugacy classes of N.

In general, the action of A on the classes of M is not the right thing to look at, one really must consider the action on the relevant characters of $M.G$. For example, take H the quaternion group or the dihedral group of order eight, N a cyclic subgroup of index two, and M the centre of H; here A acts trivially on M, but the relevant fact is that the action of A swaps those two irreducible characters of N that take the value -1 on the involution in M –these are the faithful irreducible characters of N.

If the orders of M and A are coprime then also the power maps of H can be computed from the above data. For each prime p that divides the orders of both M and A, the p-th power map is in general not uniquely determined by these input data. In this case, we can compute the (finitely many) candidates for the character table of H that are described by these data. One possible reason for ambiguities is the existence of several isoclinic but nonisomorphic groups that can arise from the input tables (cf. Section 2.2.4, see Section 2.4.12 for an example).

With the GAP function `PossibleActionsForTypeMGA` (**CTblLib: PossibleActionsForType-MGA**), one can compute the possible orbit structures induced by $G.A$ on the classes of $M.G$, and `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) computes the possible ordinary character tables for a given orbit structure. For constructing the p-modular Brauer table of a group H of the structure $M.G.A$, the GAP function `BrauerTableOfTypeMGA` (**CTblLib: BrauerTableOfTypeMGA**) takes the ordinary character table of H and the p-modular tables of the subgroup $M.G$ and the factor group $G.A$ as its input. The p-modular table of G is not explicitly needed in the construction, it is implicitly given by the class fusions from $M.G$ into $M.G.A$ and from $M.G.A$ onto $G.A$; these class fusions must of course be available.

The GAP Character Table Library contains many tables of groups of the structure $M.G.A$ as described above, which are encoded by references to the tables of the groups $M.G$ and $G.A$, plus the fusion and action information. This reduces the space needed for storing these character tables.

For examples, see Section 2.4.

2.3.2 Character Tables of Groups of the Structure $G.S_3$

Let G be a finite group, and H be an upward extension of G such that the factor group H/G is a Frobenius group $F = KC$ with abelian kernel K and cyclic complement C of prime order c. (Typical cases for F are the symmetric group S_3 on three points and the alternating group A_4 on four points.) Let N and U denote the preimages of K and C under the natural epimorphism from H onto F.
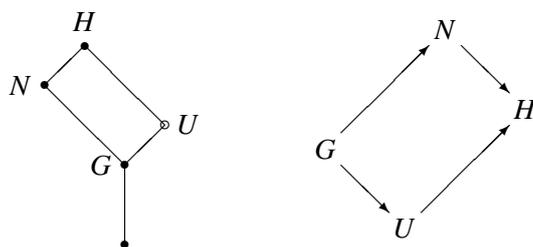
For certain isomorphism types of F, the ordinary (or p-modular) character table of H can be computed from the ordinary (or p-modular) character tables of G, U, and N, the class fusions from the table of G into those of U and N, and the permutation π induced by H on the conjugacy classes of N. This holds for example for $F = S_3$ and in the ordinary case also for $F = A_4$.

Each class of H is either a union of π-orbits or an H-class of $U \setminus G$; the latter classes are in bijection with the U-classes of $U \setminus G$, they are just $|K|$ times larger since the $|K|$ conjugates of U in H are fused. The power maps of H are uniquely determined from the power maps of U and N, because each element in F lies in K or in an F-conjugate of C.

Concerning the computation of the ordinary irreducible characters of H, we could induce the irreducible characters of U and N to H, and then take the union of the irreducible characters among those and the irreducible differences of those. (For the case $F = S_3$, this approach has been described in the Appendix of [HL94].)

The GAP function `CharacterTableOfTypeGS3` (**CTblLib: CharacterTableOfTypeGS3**) proceeds in a different way, which is suitable also for the construction of p-modular character tables of H.

By the facts listed in Section 2.2.7, for an irreducible (Brauer) character χ of N, we have $I_H(\chi)$ equal to either N or H. In the former case, χ induces irreducibly to H. In the latter case, there are extensions $\psi^{(i)}$, $1 \leq i \leq |C|$ (or $|C|_{p'}$), to H, and we have the following possibilities, depending on the restriction χ_G.

If $\chi_G = e\varphi$, for an irreducible character φ of G, then $I_U(\varphi) = U$ holds, hence the $\psi_U^{(i)}$ are $|C|$ (or $|C|_{p'}$) extensions of χ_G to U. Moreover, we have either $e = 1$ or $e^2 = |K|$. In the case $e = 1$, this determines the values of the $\psi^{(i)}$ on the classes of U outside G. In the case $e \neq 1$, we have the problem to combine e extensions of φ to a character of U that extends to H.

(One additional piece of information in the case of ordinary character tables is that the norm of this linear combination equals $1 + (|K| - 1)/|C|$, which determines the $\psi_U^{(i)}$ if $F = A_4 \cong 2^2 : 3$ or $F = 2^3 : 7$ holds; in the former case, the sum of each two out of the three different extensions of φ extends to U; in the latter case, the sum of all different extensions plus one of the extensions extends. Note that for $F = S_3$, the case $e \neq 1$ does not occur.)

The remaining case is that χ_G is not a multiple of an irreducible character of G. Then $\chi_G = \varphi_1 + \varphi_2 + \ldots + \varphi_{|K|}$, for pairwise different irreducible characters φ_i, $1 \leq i \leq |K|$, of G with the property $\varphi_i^N = \chi$. The action of U on G fixes at least one of the φ_i, since $|K| \equiv 1 \bmod |C|$. Without loss of generality, let $I_U(\varphi_1) = U$, and let $\varphi_1^{(i)}$, $1 \leq i \leq |C|$, be the extensions of φ_1 to U. (In fact exactly φ_1 is fixed by U since otherwise $k \in K$ would exist with $\varphi_1^k \neq \varphi_1$ and such that also φ_1^k would be invariant in U; but then φ_1 would be invariant under both C and C^k, which generate F. So each of the $|K|$ constituents is invariant in exactly one of the $|K|$ subgroups of type U above G.)

Then $((\varphi_1^{(i)})^H)_N = \varphi_1^N = \chi$, hence the values of $\psi^{(i)}$ on the classes of $U \setminus G$ are given by those of

$(\varphi_1^{(i)})^H$. (These are exactly the values of $\varphi_1^{(i)}$. So in both cases, we take the values of χ on N, and on the classes of $U \setminus G$ the values of the extensions of the unique extendible constituent of χ_G.)

For examples, see Section 2.5.

2.3.3 Character Tables of Groups of the Structure $G.2^2$

Let G be a finite group, and H be an upward extension of G such that the factor group H/G is a Klein four group. We assume that the ordinary character tables of G and of the three index two subgroups U_1, U_2, and U_3 (of the structures $G.2_1$, $G.2_2$, and $G.2_3$, respectively) of H above G are known, as well as the class fusions of G into these groups. The idea behind the method that is described in this section is that in this situation, there are only few possibilities for the ordinary character table of H.
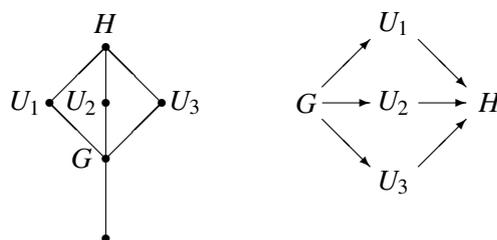
Namely, the action of H on the classes of $G.2_i$ is given by a table automorphism π_i of $G.2_i$, and H realizes compatible choices of such automorphisms π_1, π_2, π_3 in the sense that the orbits of all three π_i on the classes of G inside the groups $G.2_i$ coincide. Furthermore, if $G.2_i$ has n_i conjugacy classes then an action π_i that is a product of f_i disjoint transpositions leads to a character table candidate for $G.2^2$ that has $2n_i - 3f_i$ classes, so also the f_i must be compatible.

Taking the "inner" classes, i.e., the orbit sums of the classes inside G under the π_i, plus the union of the π_i-orbits of the classes of $G.2_i \setminus G$ gives a possibility for the classes of H. Furthermore, the power maps of the groups $G.2_i$ determine the power maps of the candidate table constructed this way.

Concerning the computation of the irreducible characters of H, we consider also the case of p-modular characters tables, where we assume that the ordinary character table of H is already known and the only task is to compute the irreducible p-modular Brauer characters.

Let χ be an irreducible (p-modular Brauer) character of G. By the facts that are listed in Section 2.2.7, there are three possibilities.

1. $I_H(\chi) = G$; then χ^H is irreducible.

2. $I_H(\chi) = G.2_i$ for i one of 1, 2, 3; then $I_{G.2_i}(\chi) = G.2_i$ for this i, so χ extends to $G.2_i$; none of these extensions extends to H (because otherwise χ would be invariant in H), so they induce irreducible characters of H.

3. $I_H(\chi) = H$; then χ extends to each of the three groups $G.2_i$, and either all these extensions induce the same character of H (which vanishes on $H \setminus G$) or they are invariant in H and thus extend to H.

In the latter part of case 3. (except if $p = 2$), the problem is to combine the values of six irreducible characters of the groups $G.2_i$ to four characters of H. This yields essentially two choices, and we try to exclude one possibility by forming scalar products with the 2-nd symmetrizations of the known irreducibles. If several possibilities remain then we get several possible tables.

So we end up with a list of possible character tables of H. The first step is to specify a list of possible triples (π_1, π_2, π_3), using the table automorphisms of the groups $G.2_i$; this can be done using the GAP function `PossibleActionsForTypeGV4` (**CTblLib: PossibleActionsForTypeGV4**). Then the GAP function `PossibleCharacterTablesOfTypeGV4` (**CTblLib: PossibleCharacterTablesOfTypeGV4**) can be used for computing the character table candidates for each given triple of permutations; it may of course happen that some triples of automorphisms are excluded in this second step.

For examples, see Section 2.6.

2.3.4 Character Tables of Groups of the Structure $2^2.G$ (August 2005)

Let G be a finite group, and H be a central extension of G by a Klein four group $Z = \langle z_1, z_2 \rangle$; set $z_3 = z_1 z_2$ and $Z_i = \langle z_i \rangle$, for $1 \leq i \leq 3$. We assume that the ordinary character tables of the three factor groups $2_i.G = H/Z_i$ of H are known, as well as the class fusions from these groups to G. The idea behind the method described in this section is that in this situation, there are only few possibilities for the ordinary character table of H.
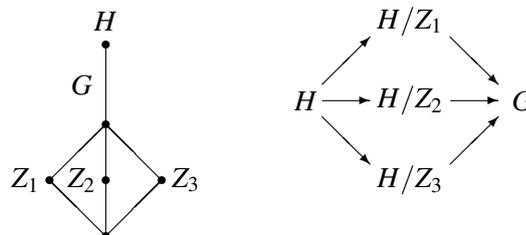
Namely, the irreducible (p-modular) characters of H are exactly the inflations of the irreducible (p-modular) characters of the three factor groups H/Z_i. (Note that for any noncyclic central subgroup C of H and any $\chi \in \mathrm{Irr}(H)$, we have $|\ker(\chi) \cap C| > 1$. To see this, let $N = \ker(\chi)$. Then clearly $|N| > 1$, and χ can be regarded as a faithful irreducible character of H/N. If $N \cap C$ would be trivial then $NC/N \cong C$ would be a noncyclic central subgroup of H/N. This cannot happen by [Isa76, Thm. 2.32 (a)], so the statement can be regarded as an obvious refinement of this theorem.) So all we have to construct is the character table head of H –classes and power maps– and the factor fusions from H to these groups.

For fixed $h \in H$, we consider the question in which H-classes the elements h, hz_1, hz_2, and hz_3 lie. There are three possibilities.

1. The four elements are all conjugate in H. Then in each of the three groups H/Z_i, the two preimages of $hZ \in H/Z$ are conjugate.

2. We are not in case 1. but two of the four elements are conjugate in H, i. e., $g^{-1}hg = hz_i$ for some $g \in H$ and some i; then $g^{-1}hz_jg = hz_iz_j$ for each j, so the four elements lie in exactly two H-classes. This implies that for $i \neq j$, the elements h and hz_j are not H-conjugate, so hZ_i is not conjugate to hz_jZ_i in H/Z_i and hZ_j is conjugate to hz_iZ_j in H/Z_j.

3. The four elements are pairwise nonconjugate in H. Then in each of the three groups H/Z_i, the two preimages of $hZ \in H/Z$ are nonconjugate.

We observe that the question which case actually applies for $h \in H$ can be decided from the three factor fusions from H/Z_i to G. So we attempt to construct the table head of H and the three factor

fusions from H to the groups H/Z_i, as follows. Each class g^G of G yields either one or two or four preimage classes in H.

In case 1., we get one preimage class in H, and have no choice for the factor fusions.

In case 2., we get two preimage classes, there is exactly one group H/Z_i in which g^G has two preimage classes –which are in bijection with the two preimage classes of H– and for the other two groups H/Z_j, the factor fusions from H map the two classes of H to the unique preimage class of g^G. (In the following picture, this is shown for $i = 1$.)
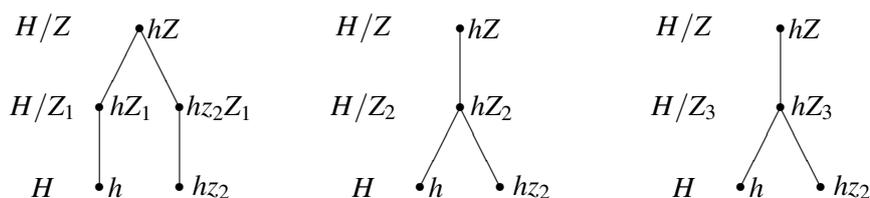
In case 3., the three factor fusions are in general not uniquely determined: We get four classes, which are defined as two pairs of preimages of the two preimages of g^G in H/Z_1 and in H/Z_2 –so we choose the relevant images in the two factor fusions to H/Z_1 and H/Z_2, respectively. Note that the class of h in H is the unique class that maps to the class of hZ_1 in H/Z_1 and to the class of hZ_2 in H/Z_2, and so on, and we define four classes of H via the four possible combinations of image classes in H/Z_1 and H/Z_2 (see the picture below).
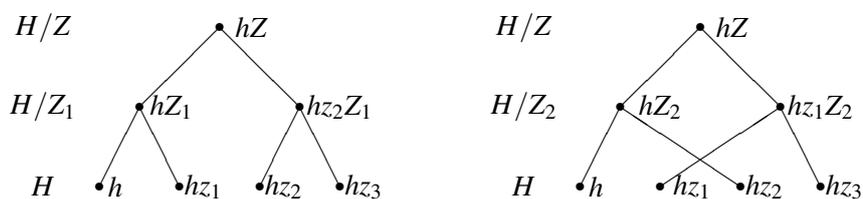
Due to the fact that in general we do not know which of the two preimage classes of g^G in H/Z_3 is the class of hZ_3, there are in general the following *two* possibilities for the fusion from H to H/Z_3.
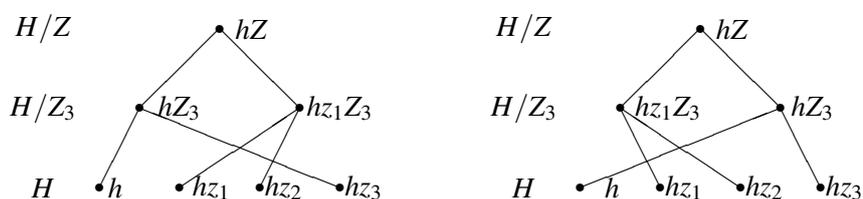
This means that we can inflate the irreducible characters of H/Z_1 and of H/Z_2 to H but that for the inflations of those irreducible characters of H/Z_3 to H that are not characters of G, the values on classes where case 3. applies are determined only up to sign.

The GAP function `PossibleCharacterTablesOfTypeV4G` (**CTblLib: PossibleCharacterTablesOfTypeV4G**) computes the candidates for the table of H from the tables of the groups H/Z_i by

setting up the character table head of H using the class fusions from H/Z_1 and H/Z_2 to G, and then forming the possible class fusions from H to H/Z_3.

If case 3. applies for a class g^G with g of *odd* element order then exactly one preimage class in H has odd element order, and we can identify this class in the groups H/Z_i, which resolves the ambiguity in this situation. More generally, if $g = k^2$ holds for some $k \in G$ then all preimages of k^G in H square to the same class of H, so again this class can be identified. In fact `PossibleCharacterTablesOfTypeV4G` (**CTblLib: PossibleCharacterTablesOfTypeV4G**) checks whether the p-th power maps of the candidate table for H and the p-th power map of H/Z_3 together with the fusion candidate form a commutative diagram.

An additional criterion used by `PossibleCharacterTablesOfTypeV4G` (**CTblLib: PossibleCharacterTablesOfTypeV4G**) is given by the property that the product of two characters inflated from H/Z_1 and H/Z_2, respectively, that are not characters of G is a character of H that contains Z_3 in its kernel, so it is checked whether the scalar products of these characters with all characters that are inflated from H/Z_3 via the candidate fusion are nonnegative integers.

Once the fusions from H to the groups H/Z_i are known, the computation of the irreducible p-modular characters of H from those of the groups H/Z_i is straightforward.

The only open question is why this construction is described in this note. That is, how is it related to table automorphisms?

The answer is that in several interesting cases, the three subgroups Z_1, Z_2, Z_3 are conjugate under an order three automorphism σ, say, of H. In this situation, the three factor groups $2_i.G = H/Z_i$ are isomorphic, and we can describe the input tables and fusions by the character table of $2_1.G$, the factor fusion from this group to G, and the automorphism σ' of G that is induced by σ. Assume that $\sigma(Z_1) = Z_2$ holds, and choose $h \in H$. Then $\sigma(hZ_1) = \sigma(h)Z_2$ is mapped to $\sigma(h)Z = \sigma'(hZ)$ under the factor fusion from $2_2.G$ to G. Let us start with the character table of $2_1.G$, and fix the class fusion to the character table of G. We may choose the identity map as isomorphism from the table of $2_1.G$ to the tables of $2_2.G$ and $2_3.G$, which implies that the class of hZ_1 is identified with the class of hZ_2 and in turn the class fusion from the table of $2_2.G$ to that of G can be chosen as the class fusion from the table of $2_1.G$ followed by the permutation of classes of G induced by σ'; analogously, the fusion from the table of $2_3.G$ is obtained by applying this permutation twice to the class fusion from the table of $2_1.G$.

For examples, see Section 2.7.

2.3.5 p-**Modular Tables of Extensions by** p-**singular Automorphisms**

Let G be a finite group, and H be an upward extension of G by an automorphism of prime order p, say. H induces a table automorphism of the p-modular character table of G; let π denote the corresponding permutation of classes of G. The columns of the p-modular character table of H are given by the orbits of π, and the irreducible Brauer characters of H are exactly the orbit sums of π on the irreducible Brauer characters of G.

Note that for computing the p-modular character table of H from that of G, it is sufficient to know the orbits of π and not π itself. Also the ordinary character table of H is not needed, but since GAP stores Brauer character tables relative to their ordinary tables, we are interested mainly in cases where the ordinary character tables of G and H and the p-modular character table of G are known. Assuming that the class fusion between the ordinary tables of G and H is stored on the table of G, the orbits of the action of H on the p-regular classes of G can be read off from it.

The GAP function `IBrOfExtensionBySingularAutomorphism` (**CTblLib: IBrOfExtension-BySingularAutomorphism**) can be used to compute the p-modular irreducibles of H.

For examples, see Section 2.8.

2.3.6   Character Tables of Subdirect Products of Index Two (July 2007)

Let C_2 denote the cyclic group of order two, let G_1, G_2 be two finite groups, and for $i \in \{1,2\}$, let $\varphi_i \colon G_i \to C_2$ be an epimorphism with kernel H_i. Let G be the subdirect product (pullback) of G_1 and G_2 w.r.t. the epimorphisms φ_i, i.e.,

$$G = \{(g_1, g_2) \in G_1 \times G_2; \varphi_1(g_1) = \varphi_2(g_2)\}.$$

The group G has index two in the direct product $G_1 \times G_2$, and G contains $H_1 \times H_2$ as a subgroup of index two.

In the following, we describe how the ordinary (or p-modular) character table of G can be computed from the ordinary (or p-modular) character tables of the groups G_i and H_i, and the class fusions from H_i to G_i.

(For the case that one of the groups G_i is a cyclic group of order four, an alternative way to construct the character table of G is described in Section 2.2.4. For the case that one of the groups G_i acts fixed point freely on the nontrivial irreducible characters of H_i, an alternative construction is described in Section 2.3.1.)
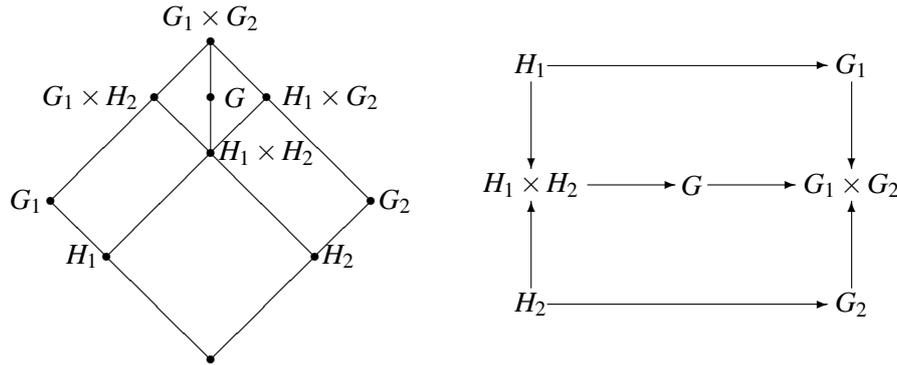
Each conjugacy class of G is either contained in $H_1 \times H_2$ or not. In the former case, let $h_i \in H_i$ and $g_i \in G_i \setminus H_i$; in particular, $(g_1, g_2) \in G$ because both $\varphi_1(g_1)$ and $\varphi_2(g_2)$ are not the identity. There are four possibilities.

1.   If $h_1^{H_1} = h_1^{G_1}$ and $h_2^{H_2} = h_2^{G_2}$ then $(h_1, h_2)^{H_1 \times H_2} = (h_1, h_2)^{G_1 \times G_2}$ holds, hence this class is equal to $(h_1, h_2)^G$.

2.   If $h_1^{H_1} \neq h_1^{G_1}$ and $h_2^{H_2} \neq h_2^{G_2}$ then the four $H_1 \times H_2$-classes with the representatives (h_1, h_2), $(h_1^{g_1}, h_2)$, $(h_1, h_2^{g_2})$, and $(h_1^{g_1}, h_2^{g_2})$ fall into two G-classes, where (h_1, h_2) is G-conjugate with $(h_1^{g_1}, h_2^{g_2})$, and $(h_1^{g_1}, h_2)$ is G-conjugate with $(h_1, h_2^{g_2})$.

3.   If $h_1^{H_1} = h_1^{G_1}$ and $h_2^{H_2} \neq h_2^{G_2}$ then the two $H_1 \times H_2$-classes with the representatives (h_1, h_2) and $(h_1, h_2^{g_2})$ fuse in G; note that there is $\tilde{g}_1 \in C_{G_1}(h_1) \setminus H_1$, so $(\tilde{g}_1, g_2) \in G$ holds.

4.   The case of $h_1^{H_1} \neq h_1^{G_1}$ and $h_2^{H_2} = h_2^{G_2}$ is analogous to case 3.

It remains to deal with the G-classes that are not contained in $H_1 \times H_2$. Each such class is in fact a conjugacy class of $G_1 \times G_2$. Note that two elements $g_1, g_2 \in G_1 \setminus H_1$ are G_1-conjugate if and only if they are H_1-conjugate. (If $g_1^x = g_2$ for $x \in G_1 \setminus H_1$ then $g_1^{g_1x} = g_2$ holds, and $g_1x \in H_1$.) This implies $(g_1, g_2)^{G_1 \times G_2} = (g_1, g_2)^{H_1 \times H_2}$, and thus this class is equal to $(g_1, g_2)^G$.

The (ordinary or p-modular) irreducible characters of G are given by the restrictions χ_G of all those irreducible characters χ of $G_1 \times G_2$ whose restriction to $H_1 \times H_2$ is irreducible, plus the induced characters φ^G, where φ runs over all those irreducible characters of $H_1 \times H_2$ that do not occur as restrictions of characters of $G_1 \times G_2$.

In other words, no irreducible character of $H_1 \times H_2$ has inertia subgroup G inside $G_1 \times G_2$. This can be seen as follows. Let φ be an irreducible character of $H_1 \times H_2$. Then $\varphi = \varphi_1 \cdot \varphi_2$, where φ_1, φ_2 are irreducible characters of $H_1 \times H_2$ with the properties that $H_2 \subseteq \ker(\varphi_1)$ and $H_1 \subseteq \ker(\varphi_2)$. Sloppy speaking, φ_i is an irreducible character of H_i.

There are four possibilities.

1. If φ_1 extends to G_1 and φ_2 extends to G_2 then φ extends to G, so φ has inertia subgroup $G_1 \times G_2$.

2. If φ_1 does not extend to G_1 and φ_2 does not extend to G_2 then $\varphi^{G_1 \times G_2}$ is irreducible, so φ has inertia subgroup $H_1 \times H_2$.

3. If φ_1 extends to G_1 and φ_2 does not extend to G_2 then φ extends to $G_1 \times H_2$ but not to $G_1 \times G_2$, so φ has inertia subgroup $G_1 \times H_2$.

4. The case that φ_1 does not extend to G_1 and φ_2 extends to G_2 is analogous to case 3, φ has inertia subgroup $H_1 \times G_2$.

For examples, see Section 2.9.

2.4 Examples for the Type $M.G.A$

2.4.1 Character Tables of Dihedral Groups

Let $n = 2^k \cdot m$ where k is a nonnegative integer and m is an odd integer, and consider the dihedral group D_{2n} of order $2n$. Let N denote the derived subgroup of D_{2n}.

If $k = 0$ then D_{2n} has the structure $M.G.A$, with $M = N$ and G the trivial group, and A a cyclic group of order two that inverts each element of N and hence acts fixed-point freely on N. The smallest nontrivial example is of course that of $D_6 \cong S_3$.

```
                          ──────── Example ────────
  gap> tblMG:= CharacterTable( "Cyclic", 3 );;
  gap> tblG:= CharacterTable( "Cyclic", 1 );;
  gap> tblGA:= CharacterTable( "Cyclic", 2 );;
  gap> StoreFusion( tblMG, [ 1, 1, 1 ], tblG );
  gap> StoreFusion( tblG, [ 1 ], tblGA );
  gap> elms:= Elements( AutomorphismsOfTable( tblMG ) );
  [ (), (2,3) ]
  gap> orbs:= [ [ 1 ], [ 2, 3 ] ];;
  gap> new:= PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA, orbs,
  >               "S3" );
  [ rec( MGfusMGA := [ 1, 2, 2 ], table := CharacterTable( "S3" ) ) ]
  gap> Display( new[1].table );
  S3
```

```
     2  1  .  1
     3  1  1  .

        1a 3a 2a
    2P  1a 3a 1a
    3P  1a 1a 2a

X.1     1  1  1
X.2     1  1 -1
X.3     2 -1  .
```

If $k > 0$ then D_{2n} has the structure $M.G.A$, with $M = N$ and G a cyclic group of order two such that $M.G$ is cyclic, and A is a cyclic group of order two that inverts each element of $M.G$ and hence acts fixed-point freely on $M.G$. The smallest nontrivial example is of course that of D_8.

```
                        _____ Example _____
 gap> tblMG:= CharacterTable( "Cyclic", 4 );;
 gap> tblG:= CharacterTable( "Cyclic", 2 );;
 gap> tblGA:= CharacterTable( "2^2" );;
 gap> OrdersClassRepresentatives( tblMG );
 [ 1, 4, 2, 4 ]
 gap> StoreFusion( tblMG, [ 1, 2, 1, 2 ], tblG );
 gap> StoreFusion( tblG, [ 1, 2 ], tblGA );
 gap> elms:= Elements( AutomorphismsOfTable( tblMG ) );
 [ (), (2,4) ]
 gap> orbs:= Orbits( Group( elms[2] ), [ 1 ..4 ] );;
 gap> new:= PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA, orbs,
 >              "order8" );
 [ rec( MGfusMGA := [ 1, 2, 3, 2 ],
       table := CharacterTable( "order8" ) ),
   rec( MGfusMGA := [ 1, 2, 3, 2 ],
       table := CharacterTable( "order8" ) ) ]
```

Here we get two possible tables, which are the character tables of the dihedral and the quaternion group of order eight, respectively.

```
                        _____ Example _____
 gap> List( new, x -> OrdersClassRepresentatives( x.table ) );
 [ [ 1, 4, 2, 2, 2 ], [ 1, 4, 2, 4, 4 ] ]
 gap> Display( new[1].table );
 order8

     2  3  2  3  2  2

        1a 4a 2a 2b 2c
    2P  1a 2a 1a 1a 1a

X.1     1  1  1  1  1
X.2     1  1  1 -1 -1
X.3     1 -1  1  1 -1
X.4     1 -1  1 -1  1
X.5     2  . -2  .  .
```

For each $k > 1$ and $m = 1$, we get two possible tables this way, that of the dihedral group of order 2^{k+1} and that of the generalized quaternion group of order 2^{k+1}.

2.4.2 An *M.G.A* Type Example with *M* noncentral in *M.G* (May 2004)

The Sylow 7 normalizer in the symmetric group S_{12} has the structure $7 : 6 \times S_5$, its intersection N with the alternating group A_{12} is of index two, it has the structure $(7 : 3 \times A_5) : 2$.

Let M denote the normal subgroup of order 7 in N, let G denote the normal subgroup of the type $3 \times A_5$ in $F = N/M \cong 3 \times S_5$, and $A = F/G$, the cyclic group of order two. Then N has the structure $M.G.A$, where A acts fixed-point freely on the irreducible characters of $M.G = 7 : 3 \times A_5$ that do not contain M in their kernels, hence the character table of N is determined by the character tables of $M.G$ and F, and the action of A on $M.G$.

Note that in this example, the group M is not central in $M.G$, unlike in most of our examples.
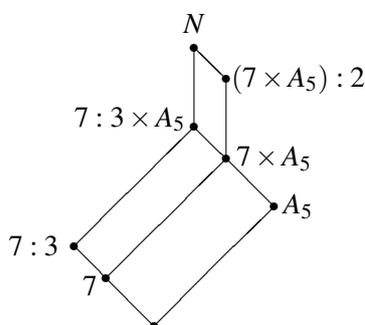

```
                          ┌─ Example ─┐
 gap> tblMG:= CharacterTable( "7:3" ) * CharacterTable( "A5" );;
 gap> nsg:= ClassPositionsOfNormalSubgroups( tblMG );
 [ [ 1 ], [ 1, 6 .. 11 ], [ 1 .. 5 ], [ 1, 6 .. 21 ], [ 1 .. 15 ],
   [ 1 .. 25 ] ]
 gap> List( nsg, x -> Sum( SizesConjugacyClasses( tblMG ){ x } ) );
 [ 1, 7, 60, 21, 420, 1260 ]
 gap> tblG:= tblMG / nsg[2];;
 gap> tblGA:= CharacterTable( "Cyclic", 3 ) * CharacterTable( "A5.2" );;
 gap> GfusGA:= PossibleClassFusions( tblG, tblGA );
 [ [ 1, 2, 3, 4, 4, 8, 9, 10, 11, 11, 15, 16, 17, 18, 18 ],
   [ 1, 2, 3, 4, 4, 15, 16, 17, 18, 18, 8, 9, 10, 11, 11 ] ]
 gap> reps:= RepresentativesFusions( Group(()), GfusGA, tblGA );
 [ [ 1, 2, 3, 4, 4, 8, 9, 10, 11, 11, 15, 16, 17, 18, 18 ] ]
 gap> StoreFusion( tblG, reps[1], tblGA );
 gap> acts:= PossibleActionsForTypeMGA( tblMG, tblG, tblGA );
 [ [ [ 1 ], [ 2 ], [ 3 ], [ 4, 5 ], [ 6, 11 ], [ 7, 12 ], [ 8, 13 ],
       [ 9, 15 ], [ 10, 14 ], [ 16 ], [ 17 ], [ 18 ], [ 19, 20 ],
       [ 21 ], [ 22 ], [ 23 ], [ 24, 25 ] ] ]
 gap> poss:= PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA,
 >               acts[1], "A12N7" );
 [ rec(
       MGfusMGA := [ 1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 5, 6, 7, 9, 8, 10,
           11, 12, 13, 13, 14, 15, 16, 17, 17 ],
       table := CharacterTable( "A12N7" ) ) ]
```

Let us compare the result table with the table of the Sylow 7 normalizer in A_{12}.

```
                            Example
gap> g:= AlternatingGroup( 12 );;
gap> IsRecord( TransformingPermutationsCharacterTables( poss[1].table,
>                CharacterTable( Normalizer( g, SylowSubgroup( g, 7 ) ) ) ) );
true
```

Since July 2007, an alternative way to construct the character table of *N* from other character tables is to exploit its structure as a subdirect product of index two in the group $7 : 6 \times S_5$, see Section 2.3.6.

```
                            Example
gap> tblh1:= CharacterTable( "7:3" );;
gap> tblg1:= CharacterTable( "7:6" );;
gap> tblh2:= CharacterTable( "A5" );;
gap> tblg2:= CharacterTable( "A5.2" );;
gap> subdir:= CharacterTableOfIndexTwoSubdirectProduct( tblh1, tblg1,
>                tblh2, tblg2, "(7:3xA5).2" );;
gap> IsRecord( TransformingPermutationsCharacterTables( poss[1].table,
>                subdir.table ) );
true
```

For storing the table of *N* in the GAP Character Table Library, the construction as a subdirect product is more suitable, since the "auxiliary table" of the direct product $7 : 3 \times A_5$ need not be stored in the library.

2.4.3 Atlas Tables of the Type *M.G.A*

We show the construction of some character tables of groups of the type *M.G.A* that are contained in the GAP Character Table Library. Each entry in the following input list contains the names of the library character tables of *M.G*, *G*, *G.A*, and *M.G.A*.

First we consider the situation where *G* is a simple group or a central extension of a simple group whose character table is shown in the Atlas, and *M* and *A* are cyclic groups such that *M* is central in *M.G*.

In the following cases, the character tables are uniquely determined by the input tables. Note that in each of these cases, |*A*| and |*M*| are coprime.

```
                            Example
gap> listMGA:= [
> [ "3.A6",      "A6",        "A6.2_1",      "3.A6.2_1"      ],
> [ "3.A6",      "A6",        "A6.2_2",      "3.A6.2_2"      ],
> [ "6.A6",      "2.A6",      "2.A6.2_1",    "6.A6.2_1"      ],
> [ "6.A6",      "2.A6",      "2.A6.2_2",    "6.A6.2_2"      ],
> [ "3.A7",      "A7",        "A7.2",        "3.A7.2"        ],
> [ "6.A7",      "2.A7",      "2.A7.2",      "6.A7.2"        ],
> [ "3.L3(4)",   "L3(4)",     "L3(4).2_2",   "3.L3(4).2_2"   ],
> [ "3.L3(4)",   "L3(4)",     "L3(4).2_3",   "3.L3(4).2_3"   ],
> [ "6.L3(4)",   "2.L3(4)",   "2.L3(4).2_2", "6.L3(4).2_2"   ],
> [ "6.L3(4)",   "2.L3(4)",   "2.L3(4).2_3", "6.L3(4).2_3"   ],
> [ "12_1.L3(4)", "4_1.L3(4)", "4_1.L3(4).2_2", "12_1.L3(4).2_2" ],
> [ "12_1.L3(4)", "4_1.L3(4)", "4_1.L3(4).2_3", "12_1.L3(4).2_3" ],
> [ "12_2.L3(4)", "4_2.L3(4)", "4_2.L3(4).2_2", "12_2.L3(4).2_2" ],
> [ "12_2.L3(4)", "4_2.L3(4)", "4_2.L3(4).2_3", "12_2.L3(4).2_3" ],
> [ "3.U3(5)",   "U3(5)",     "U3(5).2",     "3.U3(5).2"     ],
```

```
> [ "3.M22",        "M22",          "M22.2",          "3.M22.2"        ],
> [ "6.M22",        "2.M22",        "2.M22.2",        "6.M22.2"        ],
> [ "12.M22",       "4.M22",        "4.M22.2",        "12.M22.2"       ],
> [ "3.L3(7)",      "L3(7)",        "L3(7).2",        "3.L3(7).2"      ],
> [ "3_1.U4(3)",    "U4(3)",        "U4(3).2_1",      "3_1.U4(3).2_1"  ],
> [ "3_1.U4(3)",    "U4(3)",        "U4(3).2_2'",     "3_1.U4(3).2_2'" ],
> [ "3_2.U4(3)",    "U4(3)",        "U4(3).2_1",      "3_2.U4(3).2_1"  ],
> [ "3_2.U4(3)",    "U4(3)",        "U4(3).2_3'",     "3_2.U4(3).2_3'" ],
> [ "6_1.U4(3)",    "2.U4(3)",      "2.U4(3).2_1",    "6_1.U4(3).2_1"  ],
> [ "6_1.U4(3)",    "2.U4(3)",      "2.U4(3).2_2'",   "6_1.U4(3).2_2'" ],
> [ "6_2.U4(3)",    "2.U4(3)",      "2.U4(3).2_1",    "6_2.U4(3).2_1"  ],
> [ "6_2.U4(3)",    "2.U4(3)",      "2.U4(3).2_3'",   "6_2.U4(3).2_3'" ],
> [ "12_1.U4(3)",   "4.U4(3)",      "4.U4(3).2_1",    "12_1.U4(3).2_1" ],
> [ "12_2.U4(3)",   "4.U4(3)",      "4.U4(3).2_1",    "12_2.U4(3).2_1" ],
> [ "3.G2(3)",      "G2(3)",        "G2(3).2",        "3.G2(3).2"      ],
> [ "3.U3(8)",      "U3(8)",        "U3(8).2",        "3.U3(8).2"      ],
> [ "3.U3(8).3_1",  "U3(8).3_1",    "U3(8).6",        "3.U3(8).6"      ],
> [ "3.J3",         "J3",           "J3.2",           "3.J3.2"         ],
> [ "3.U3(11)",     "U3(11)",       "U3(11).2",       "3.U3(11).2"     ],
> [ "3.McL",        "McL",          "McL.2",          "3.McL.2"        ],
> [ "3.O7(3)",      "O7(3)",        "O7(3).2",        "3.O7(3).2"      ],
> [ "6.O7(3)",      "2.O7(3)",      "2.O7(3).2",      "6.O7(3).2"      ],
> [ "3.U6(2)",      "U6(2)",        "U6(2).2",        "3.U6(2).2"      ],
> [ "6.U6(2)",      "2.U6(2)",      "2.U6(2).2",      "6.U6(2).2"      ],
> [ "3.Suz",        "Suz",          "Suz.2",          "3.Suz.2"        ],
> [ "6.Suz",        "2.Suz",        "2.Suz.2",        "6.Suz.2"        ],
> [ "3.ON",         "ON",           "ON.2",           "3.ON.2"         ],
> [ "3.Fi22",       "Fi22",         "Fi22.2",         "3.Fi22.2"       ],
> [ "6.Fi22",       "2.Fi22",       "2.Fi22.2",       "6.Fi22.2"       ],
> [ "3.2E6(2)",     "2E6(2)",       "2E6(2).2",       "3.2E6(2).2"     ],
> [ "6.2E6(2)",     "2.2E6(2)",     "2.2E6(2).2",     "6.2E6(2).2"     ],
> [ "3.F3+",        "F3+",          "F3+.2",          "3.F3+.2"        ],
> ];;
```

(We need not consider groups $3.U_3(8).6'$ and $3.U_3(8).6'$, see Section 2.4.7.)

Note that the groups of the types $12_1.L_3(4).2_1$ and $12_2.L_3(4).2_1$ have central subgroups of order six, so we cannot choose G equal to $4_1.L_3(4)$ and $4_2.L_3(4)$, respectively, in these cases. See Section 2.4.4 for the construction of these tables.

Also in the following cases, $|A|$ and $|M|$ are coprime, we have $|M| = 3$ and $|A| = 2$. The group $M.G$ has a central subgroup of the type $2^2 \times 3$, and A acts on this group by inverting the elements in the subgroup of order 3 and by swapping two involutions in the Klein four group.

```
──────────── Example ────────────
gap> Append( listMGA, [
> [ "(2^2x3).L3(4)",   "2^2.L3(4)",    "2^2.L3(4).2_2", "(2^2x3).L3(4).2_2" ],
> [ "(2^2x3).L3(4)",   "2^2.L3(4)",    "2^2.L3(4).2_3", "(2^2x3).L3(4).2_3" ],
> [ "(2^2x3).U6(2)",   "2^2.U6(2)",    "2^2.U6(2).2",   "(2^2x3).U6(2).2"   ],
> [ "(2^2x3).2E6(2)",  "2^2.2E6(2)",   "2^2.2E6(2).2",  "(2^2x3).2E6(2).2"  ],
> ] );
```

Additionally, there are a few cases where A has order two, and $G.A$ has a factor group of the type 2^2, and a few cases where M has the type 2^2 and A is of order three and acts transitively on the

involutions in *M*.

```
                          ─── Example ───
gap> Append( listMGA, [
> [ "3.A6.2_3",      "A6.2_3",    "A6.2^2",     "3.A6.2^2"          ],
> [ "3.L3(4).2_1",   "L3(4).2_1", "L3(4).2^2",  "3.L3(4).2^2"       ],
> [ "3_1.U4(3).2_2", "U4(3).2_2", "U4(3).(2^2)_{122}",
>                                               "3_1.U4(3).(2^2)_{122}" ],
> [ "3_2.U4(3).2_3", "U4(3).2_3", "U4(3).(2^2)_{133}",
>                                               "3_2.U4(3).(2^2)_{133}" ],
> [ "3^2.U4(3).2_3'", "3_2.U4(3).2_3'", "3_2.U4(3).(2^2)_{133}",
>                                               "3^2.U4(3).(2^2)_{133}" ],
> [ "2^2.L3(4)",      "L3(4)",     "L3(4).3",    "2^2.L3(4).3"       ],
> [ "(2^2x3).L3(4)",  "3.L3(4)",   "3.L3(4).3",  "(2^2x3).L3(4).3"   ],
> [ "2^2.L3(4).2_1",  "L3(4).2_1", "L3(4).6",    "2^2.L3(4).6"       ],
> [ "2^2.Sz(8)",      "Sz(8)",     "Sz(8).3",    "2^2.Sz(8).3"       ],
> [ "2^2.U6(2)",      "U6(2)",     "U6(2).3",    "2^2.U6(2).3"       ],
> [ "(2^2x3).U6(2)",  "3.U6(2)",   "3.U6(2).3",  "(2^2x3).U6(2).3"   ],
> [ "2^2.O8+(2)",     "O8+(2)",    "O8+(2).3",   "2^2.O8+(2).3"      ],
> [ "2^2.O8+(3)",     "O8+(3)",    "O8+(3).3",   "2^2.O8+(3).3"      ],
> [ "2^2.2E6(2)",     "2E6(2)",    "2E6(2).3",   "2^2.2E6(2).3"      ],
> ] );
```

The constructions of the character tables of groups of the types $4_2.L_3(4).2_3$, $12_2.L_3(4).2_3$, $12_1.U_4(3).2'_2$ and $12_2.U_4(3).2'_3$ is described in Section 2.4.5 and 2.4.6, in these cases the GAP functions return several possible tables.

The construction of the various character table of groups of the types $4_1.L_3(4).2^2$ and $4_2.L_3(4).2^2$ are described in Section 2.6.7.

The following function takes the ordinary character tables of the groups *M.G*, *G*, and *G.A*, a string to be used as the `Identifier` (**Reference: Identifier for tables of marks**) value of the character table of *M.G.A*, and the character table of *M.G.A* that is contained in the GAP Character Table Library; the function first computes the possible actions of *G.A* on the classes of *M.G*, using the function `PossibleActionsForTypeMGA` (**CTblLib: PossibleActionsForTypeMGA**), then computes the union of possible character tables for these actions, and then representatives up to permutation equivalence; if there is only one solution then the result table is compared with the library table.

```
                          ─── Example ───
gap> ConstructOrdinaryMGATable:= function( tblMG, tblG, tblGA, name, lib )
>     local acts, poss, trans;
>
>     acts:= PossibleActionsForTypeMGA( tblMG, tblG, tblGA );
>     poss:= Concatenation( List( acts, pi ->
>             PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA, pi,
>                 name ) ) );
>     poss:= RepresentativesCharacterTables( poss );
>     if Length( poss ) = 1 then
>       # Compare the computed table with the library table.
>       if not IsCharacterTable( lib ) then
>         List( poss, x -> AutomorphismsOfTable( x.table ) );
>         Print( "#I  no library table for ", name, "\n" );
>       else
>         trans:= TransformingPermutationsCharacterTables( poss[1].table,
>                 lib );
```

```
>              if not IsRecord( trans ) then
>                Print( "#E  computed table and library table for ", name,
>                      " differ\n" );
>              fi;
>              # Compare the computed fusion with the stored one.
>              if OnTuples( poss[1].MGfusMGA, trans.columns )
>                    <> GetFusionMap( tblMG, lib ) then
>                Print( "#E  computed and stored fusion for ", name,
>                      " differ\n" );
>              fi;
>           fi;
>        elif Length( poss ) = 0 then
>          Print( "#E  no solution for ", name, "\n" );
>        else
>          Print( "#E  ", Length( poss ), " possibilities for ", name, "\n" );
>        fi;
>        return poss;
>      end;;
```

The following function takes the ordinary character tables of the groups *M.G*, *G.A*, and *M.G.A*, and tries to construct the *p*-modular character tables of *M.G.A* from the *p*-modular character tables of the first two of these tables, for all prime divisors *p* of the order of *M.G.A*. Note that the tables of *G* are not needed in the construction, only the class fusions from *M.G* to *M.G.A* and from *M.G.A* to *G.A* must be stored.

```
                            Example
  gap> ConstructModularMGATables:= function( tblMG, tblGA, ordtblMGA )
  >     local name, poss, p, modtblMG, modtblGA, modtblMGA, modlib, trans;
  >
  >     name:= Identifier( ordtblMGA );
  >     poss:= [];
  >     for p in PrimeDivisors( Size( ordtblMGA ) ) do
  >       modtblMG := tblMG mod p;
  >       modtblGA := tblGA mod p;
  >       if ForAll( [ modtblMG, modtblGA ], IsCharacterTable ) then
  >         modtblMGA:= BrauerTableOfTypeMGA( modtblMG, modtblGA, ordtblMGA );
  >         Add( poss, modtblMGA );
  >         modlib:= ordtblMGA mod p;
  >         if IsCharacterTable( modlib ) then
  >           trans:= TransformingPermutationsCharacterTables( modtblMGA.table,
  >                     modlib );
  >           if not IsRecord( trans ) then
  >             Print( "#E  computed table and library table for ", name,
  >                     " mod ", p, " differ\n" );
  >           fi;
  >         else
  >           AutomorphismsOfTable( modtblMGA.table );
  >           Print( "#I  no library table for ", name, " mod ", p, "\n" );
  >         fi;
  >       else
  >         Print( "#I  not all input tables for ", name, " mod ", p,
  >                 " available\n" );
  >       fi;
```

```
>     od;
>
>     return poss;
>     end;;
```

Now we run the constructions for the cases in the list. Note that in order to avoid conflicts of the class fusions that arise in the construction with the class fusions that are already stored on the library tables, we choose identifiers for the result tables that are different from the identifiers of the library tables.

```
——————————————— Example ———————————————
gap> for  input in listMGA do
>       tblMG := CharacterTable( input[1] );
>       tblG  := CharacterTable( input[2] );
>       tblGA := CharacterTable( input[3] );
>       name  := Concatenation( "new", input[4] );
>       lib   := CharacterTable( input[4] );
>       poss:= ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );
>       if 1 <> Length( poss ) then
>         Print( "#I  ", Length( poss ), " possibilities for ", name, "\n" );
>       elif lib = fail then
>         Print( "#I  no library table for ", input[4], "\n" );
>       else
>         ConstructModularMGATables( tblMG, tblGA, lib );
>       fi;
>   od;
#I  not all input tables for 3.2E6(2).2 mod 2 available
#I  not all input tables for 3.2E6(2).2 mod 3 available
#I  not all input tables for 3.2E6(2).2 mod 5 available
#I  not all input tables for 3.2E6(2).2 mod 7 available
#I  not all input tables for 3.2E6(2).2 mod 11 available
#I  not all input tables for 3.2E6(2).2 mod 13 available
#I  not all input tables for 3.2E6(2).2 mod 17 available
#I  not all input tables for 3.2E6(2).2 mod 19 available
#I  not all input tables for 6.2E6(2).2 mod 2 available
#I  not all input tables for 6.2E6(2).2 mod 3 available
#I  not all input tables for 6.2E6(2).2 mod 5 available
#I  not all input tables for 6.2E6(2).2 mod 7 available
#I  not all input tables for 6.2E6(2).2 mod 11 available
#I  not all input tables for 6.2E6(2).2 mod 13 available
#I  not all input tables for 6.2E6(2).2 mod 17 available
#I  not all input tables for 6.2E6(2).2 mod 19 available
#I  not all input tables for 3.F3+.2 mod 2 available
#I  not all input tables for 3.F3+.2 mod 3 available
#I  not all input tables for 3.F3+.2 mod 5 available
#I  not all input tables for 3.F3+.2 mod 7 available
#I  not all input tables for 3.F3+.2 mod 13 available
#I  not all input tables for 3.F3+.2 mod 17 available
#I  not all input tables for 3.F3+.2 mod 29 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 2 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 3 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 5 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 7 available
```

```
#I  not all input tables for (2^2x3).2E6(2).2 mod 11 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 13 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 17 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 19 available
#I  not all input tables for 3^2.U4(3).(2^2)_{133} mod 2 available
#I  not all input tables for 3^2.U4(3).(2^2)_{133} mod 5 available
#I  not all input tables for 3^2.U4(3).(2^2)_{133} mod 7 available
#I  not all input tables for 2^2.O8+(3).3 mod 5 available
#I  not all input tables for 2^2.O8+(3).3 mod 7 available
#I  not all input tables for 2^2.O8+(3).3 mod 13 available
#I  not all input tables for 2^2.2E6(2).3 mod 2 available
#I  not all input tables for 2^2.2E6(2).3 mod 3 available
#I  not all input tables for 2^2.2E6(2).3 mod 5 available
#I  not all input tables for 2^2.2E6(2).3 mod 7 available
#I  not all input tables for 2^2.2E6(2).3 mod 11 available
#I  not all input tables for 2^2.2E6(2).3 mod 13 available
#I  not all input tables for 2^2.2E6(2).3 mod 17 available
#I  not all input tables for 2^2.2E6(2).3 mod 19 available
```

We do not get any unexpected output, so the character tables in question are determined by the inputs.

Alternative constructions of the character tables of $3.A_6.2^2$, $3.L_3(4).2^2$, and $3_2.U_4(3).(2^2)_{133}$ can be found in Section 2.6.2.

2.4.4   More **Atlas** Tables of the Type $M.G.A$

In the following situations, we have $|A| = 2$, and $|M|$ is a multiple of 2. The result turns out to be unique up to isoclinism, see Section 2.3.1.

First, there are some cases where the centre of $M.G$ is a cyclic group of order four, and $|M| = 2$ holds.

```
──────────────────────────── Example ────────────────────────────
gap> listMGA2:= [
> [ "4_1.L3(4)",  "2.L3(4)",  "2.L3(4).2_1",  "4_1.L3(4).2_1" ],
> [ "4_1.L3(4)",  "2.L3(4)",  "2.L3(4).2_2",  "4_1.L3(4).2_2" ],
> [ "4_2.L3(4)",  "2.L3(4)",  "2.L3(4).2_1",  "4_2.L3(4).2_1" ],
> [ "4.M22",      "2.M22",    "2.M22.2",      "4.M22.2"       ],
> [ "4.U4(3)",    "2.U4(3)",  "2.U4(3).2_2",  "4.U4(3).2_2"   ],
> [ "4.U4(3)",    "2.U4(3)",  "2.U4(3).2_3",  "4.U4(3).2_3"   ],
> ];;
```

Note that the groups $4_1.L3(4).2_3$ and $4_2.L3(4).2_2$ and their isoclinic variants have centres of order four, so they do not appear here. The construction of the character table of $4_2.L3(4).2_3$ is more involved, it is described in Section 2.4.5.

Also in the following cases, we have $|M| = 2$, but the situation is different because $M.G$ has a central subgroup of the type 2^2 containing a unique subgroup of order 2 that is central in $M.G.A$.

```
──────────────────────────── Example ────────────────────────────
gap> Append( listMGA2, [
> [ "2^2.L3(4)",     "2.L3(4)",      "2.L3(4).2_2",           "2^2.L3(4).2_2" ],
> [ "2^2.L3(4)",     "2.L3(4)",      "2.L3(4).2_3",           "2^2.L3(4).2_3" ],
> [ "2^2.L3(4).2_1", "2.L3(4).2_1", "2.L3(4).(2^2)_{123}", "2^2.L3(4).2^2" ],
```

```
>  [ "2^2.O8+(2)",    "2.O8+(2)",    "2.O8+(2).2",          "2^2.O8+(2).2"  ],
>  [ "2^2.U6(2)",     "2.U6(2)",     "2.U6(2).2",           "2^2.U6(2).2"   ],
>  [ "2^2.2E6(2)",    "2.2E6(2)",    "2.2E6(2).2",          "2^2.2E6(2).2"  ],
> ] );
```

Next there are two constructions for $G = 6.L_3(4)$, with $|M| = 12$ and $|A| = 2$. Note that the groups $12_1.L3(4).2_1$ and $12_2.L3(4).2_1$ have central subgroups of the order six, so we cannot use the factor groups $4_1.L3(4).2_1$ and $4_2.L3(4).2_1$, respectively, for the constructions.

```
────────────────── Example ──────────────────
gap> Append( listMGA2, [
> [ "12_1.L3(4)", "6.L3(4)", "6.L3(4).2_1", "12_1.L3(4).2_1" ],
> [ "12_2.L3(4)", "6.L3(4)", "6.L3(4).2_1", "12_2.L3(4).2_1" ],
> ] );
```

Next there are alternative constructions for tables which have been constructed in Section 2.4.3. There we had viewed the groups of the structure 12.*S*.2, for a simple group *S*, as 3.*G*.2 with $G = 4.S$. Here we view these groups as 2.*G*.2 with $G = 6.S$, which means that we do not prescribe the 4.*S*.2 type factor group. So it is not surprising that we get more than one solution, and that the computation of the 2-power map of 12.*S*.2 is more involved. Note that the construction of the character table of $12_2.L_3(4).2_3$ is more involved, it is described in Section 2.4.5.

```
────────────────── Example ──────────────────
gap> Append( listMGA2, [
> [ "12.M22",     "6.M22",     "6.M22.2",       "12.M22.2"       ],
> [ "12_1.L3(4)", "6.L3(4)",   "6.L3(4).2_2",   "12_1.L3(4).2_2" ],
> [ "12_1.U4(3)", "6_1.U4(3)", "6_1.U4(3).2_2", "12_1.U4(3).2_2" ],
> [ "12_2.U4(3)", "6_2.U4(3)", "6_2.U4(3).2_3", "12_2.U4(3).2_3" ],
> ] );
```

Finally, there are alternative constructions for the cases where the group *M*.*G* has a central subgroup of the type $2^2 \times 3$, and *A* acts on this group by inverting the elements in the subgroup of order 3 and by swapping two involutions in the Klein four group.

```
────────────────── Example ──────────────────
gap> Append( listMGA2, [
> [ "(2^2x3).L3(4)",  "6.L3(4)",  "6.L3(4).2_2", "(2^2x3).L3(4).2_2" ],
> [ "(2^2x3).L3(4)",  "6.L3(4)",  "6.L3(4).2_3", "(2^2x3).L3(4).2_3" ],
> [ "(2^2x3).U6(2)",  "6.U6(2)",  "6.U6(2).2",   "(2^2x3).U6(2).2"   ],
> [ "(2^2x3).2E6(2)", "6.2E6(2)", "6.2E6(2).2",  "(2^2x3).2E6(2).2"  ],
> ] );
```

Now we run the constructions for the cases in the list.

```
────────────────── Example ──────────────────
gap> for  input in listMGA2 do
>       tblMG := CharacterTable( input[1] );
>       tblG  := CharacterTable( input[2] );
>       tblGA := CharacterTable( input[3] );
>       name  := Concatenation( "new", input[4] );
>       lib   := CharacterTable( input[4] );
>       poss:= ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );
```

```
>       if Length( poss ) = 2 then
>         iso:= CharacterTableIsoclinic( poss[1].table );
>         if IsRecord( TransformingPermutationsCharacterTables( poss[2].table,
>                          iso ) ) then
>           Unbind( poss[2] );
>         fi;
>       elif Length( poss ) = 1 then
>         Print( "#I  unique up to permutation equivalence: ", name, "\n" );
>       fi;
>       if 1 <> Length( poss ) then
>         Print( "#I  ", Length( poss ), " possibilities for ", name, "\n" );
>       elif lib = fail then
>         Print( "#I  no library table for ", input[4], "\n" );
>       else
>         ConstructModularMGATables( tblMG, tblGA, lib );
>       fi;
>   od;
#E  2 possibilities for new4_1.L3(4).2_1
#E  2 possibilities for new4_1.L3(4).2_2
#E  2 possibilities for new4_2.L3(4).2_1
#E  2 possibilities for new4.M22.2
#E  2 possibilities for new4.U4(3).2_2
#E  2 possibilities for new4.U4(3).2_3
#I  unique up to permutation equivalence: new2^2.L3(4).2_2
#I  unique up to permutation equivalence: new2^2.L3(4).2_3
#I  unique up to permutation equivalence: new2^2.L3(4).2^2
#I  unique up to permutation equivalence: new2^2.O8+(2).2
#I  unique up to permutation equivalence: new2^2.U6(2).2
#I  unique up to permutation equivalence: new2^2.2E6(2).2
#I  not all input tables for 2^2.2E6(2).2 mod 2 available
#I  not all input tables for 2^2.2E6(2).2 mod 3 available
#I  not all input tables for 2^2.2E6(2).2 mod 5 available
#I  not all input tables for 2^2.2E6(2).2 mod 7 available
#E  2 possibilities for new12_1.L3(4).2_1
#E  2 possibilities for new12_2.L3(4).2_1
#E  2 possibilities for new12.M22.2
#E  2 possibilities for new12_1.L3(4).2_2
#E  2 possibilities for new12_1.U4(3).2_2
#E  2 possibilities for new12_2.U4(3).2_3
#I  unique up to permutation equivalence: new(2^2x3).L3(4).2_2
#I  unique up to permutation equivalence: new(2^2x3).L3(4).2_3
#I  unique up to permutation equivalence: new(2^2x3).U6(2).2
#I  unique up to permutation equivalence: new(2^2x3).2E6(2).2
#I  not all input tables for (2^2x3).2E6(2).2 mod 2 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 3 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 5 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 7 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 11 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 13 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 17 available
#I  not all input tables for (2^2x3).2E6(2).2 mod 19 available
```

Again, we do not get any unexpected output, so the character tables in question are determined up to isoclinism by the inputs.

2.4.5 The Character Tables of $4_2.L_3(4).2_3$ and $12_2.L_3(4).2_3$

In the construction of the character table of $M.G.A = 4_2.L_3(4).2_3$ from the tables of $M.G = 4_2.L_3(4)$ and $G.A = 2.L_3(4).2_3$, the action of A on the classes of $M.G$ is uniquely determined, but we get four possible character tables.

```
——————————————————————— Example ———————————————————————
  gap> tblMG := CharacterTable( "4_2.L3(4)" );;
  gap> tblG  := CharacterTable( "2.L3(4)" );;
  gap> tblGA := CharacterTable( "2.L3(4).2_3" );;
  gap> name  := "new4_2.L3(4).2_3";;
  gap> lib   := CharacterTable( "4_2.L3(4).2_3" );;
  gap> poss  := ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );
  #E  4 possibilities for new4_2.L3(4).2_3
  [ rec(
        MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 7, 9, 10, 11, 10, 12,
            12, 13, 14, 15, 14, 16, 17, 18, 17, 19, 20, 21, 22, 19, 22,
            21, 20 ], table := CharacterTable( "new4_2.L3(4).2_3" ) ),
    rec(
        MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 7, 9, 10, 11, 10, 12,
            12, 13, 14, 15, 14, 16, 17, 18, 17, 19, 20, 21, 22, 19, 22,
            21, 20 ], table := CharacterTable( "new4_2.L3(4).2_3" ) ),
    rec(
        MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 7, 9, 10, 11, 10, 12,
            12, 13, 14, 15, 14, 16, 17, 18, 17, 19, 20, 21, 22, 19, 22,
            21, 20 ], table := CharacterTable( "new4_2.L3(4).2_3" ) ),
    rec(
        MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 7, 9, 10, 11, 10, 12,
            12, 13, 14, 15, 14, 16, 17, 18, 17, 19, 20, 21, 22, 19, 22,
            21, 20 ], table := CharacterTable( "new4_2.L3(4).2_3" ) ) ]
```

The centre of $4_2.L_3(4)$ is inverted by the action of the outer automorphism, so the existence of *two* possible tables can be expected because two isoclinic groups of the type $4_2.L_3(4).2_3$ exist, see Section 2.2.6.

Indeed the result consists of two pairs of isoclinic tables, so we have to decide which pair of tables belongs to the groups of the type $4_2.L_3(4).2_3$.

```
——————————————————————— Example ———————————————————————
  gap> IsRecord( TransformingPermutationsCharacterTables( poss[1].table,
  >                 CharacterTableIsoclinic( poss[4].table ) ) );
  true
  gap> IsRecord( TransformingPermutationsCharacterTables( poss[2].table,
  >                 CharacterTableIsoclinic( poss[3].table ) ) );
  true
```

The possible tables differ only w.r.t. the 2-power map and perhaps the element orders. The Atlas prints the table of the split extension of $M.G$, this table is one of the first two possibilities.

```
─────────────────── Example ───────────────────
gap> List( poss, x -> PowerMap( x.table, 2 ) );
[ [ 1, 3, 1, 1, 3, 6, 8, 6, 4, 4, 4, 5, 16, 18, 16, 13, 15, 13, 19,
      21, 19, 21, 1, 1, 6, 6, 9, 9, 11, 11, 16, 16, 13, 13 ],
  [ 1, 3, 1, 1, 3, 6, 8, 6, 4, 4, 4, 5, 16, 18, 16, 13, 15, 13, 19,
      21, 19, 21, 1, 1, 6, 6, 11, 11, 9, 9, 16, 16, 13, 13 ],
  [ 1, 3, 1, 1, 3, 6, 8, 6, 4, 4, 4, 5, 16, 18, 16, 13, 15, 13, 19,
      21, 19, 21, 3, 3, 8, 8, 9, 9, 11, 11, 18, 18, 15, 15 ],
  [ 1, 3, 1, 1, 3, 6, 8, 6, 4, 4, 4, 5, 16, 18, 16, 13, 15, 13, 19,
      21, 19, 21, 3, 3, 8, 8, 11, 11, 9, 9, 18, 18, 15, 15 ] ]
```

The 2-power map is not determined by the irreducible characters (and by the 2-power map of the factor group $2.L_3(4).2_3$). We determine this map using the embedding of $4_2.L_3(4).2_3$ into $4.U_4(3).2_3$. Note that $L_3(4).2_3$ is a maximal subgroup of $U_4(3).2_3$ (see [CCN+85, p. 52]), and that the subgroup $L_3(4)$ of $U_4(3)$ lifts to $4_2.L_3(4)$ in $4.U_4(3)$ because no embedding of $L_3(4)$, $2.L_3(4)$, or $4_1.L_3(4)$ into $4.U_4(3)$ is possible.

```
─────────────────── Example ───────────────────
gap> PossiblePowerMaps( poss[1].table, 2 );
[ [ 1, 3, 1, 1, 3, 6, 8, 6, 4, 4, 4, 5, 16, 18, 16, 13, 15, 13, 19,
      21, 19, 21, 1, 1, 6, 6, 11, 11, 9, 9, 16, 16, 13, 13 ],
  [ 1, 3, 1, 1, 3, 6, 8, 6, 4, 4, 4, 5, 16, 18, 16, 13, 15, 13, 19,
      21, 19, 21, 1, 1, 6, 6, 9, 9, 11, 11, 16, 16, 13, 13 ] ]
gap> t:= CharacterTable( "4.U4(3)" );;
gap> List( [ "L3(4)", "2.L3(4)", "4_1.L3(4)", "4_2.L3(4)" ], name ->
>           Length( PossibleClassFusions( CharacterTable( name ), t ) ) );
[ 0, 0, 0, 4 ]
```

So the split extension $4_2.L_3(4).2_3$ of $4_2.L_3(4)$ is a subgroup of the split extension $4.U_4(3).2_3$ of $4.U_4(3)$, and only one of the two possible tables of $4_2.L_3(4).2_3$ admits a class fusion into the Atlas table of $4.U_3(4).2_3$; the construction of the latter table is shown in Section 2.4.3.

```
─────────────────── Example ───────────────────
gap> t2:= CharacterTable( "4.U4(3).2_3" );;
gap> List( poss, x -> Length( PossibleClassFusions( x.table, t2 ) ) );
[ 0, 16, 0, 0 ]
```

I do not know a character theoretic argument that would disprove the existence of a group whose character table is the other candidate (or its isoclinic variant). For example, the table passes the tests from Section 2.4.17.

(It is straightforward to compute all extensions of $4_2.L_3(4)$ by an automorphism of order two. The extensions with 34 conjugacy classes belong to the second candidate and its isoclinic variant.)

The correct table is the one that is contained in the GAP Character Table Library.

```
─────────────────── Example ───────────────────
gap> IsRecord( TransformingPermutationsCharacterTables( poss[2].table,
>                 lib ) );
true
gap> ConstructModularMGATables( tblMG, tblGA, lib );;
```

In the construction of the character table of $12_2.L_3(4).2_3$, the same ambiguity arises. We resolve it using the fact that $4_2.L_3(4).2_3$ occurs as a factor group, modulo the unique normal subgroup of order three.

```
━━━━━━━━━━━━━━━━━━ Example ━━━━━━━━━━━━━━━━━━
 gap> tblMG := CharacterTable( "12_2.L3(4)" );;
 gap> tblG  := CharacterTable( "6.L3(4)" );;
 gap> tblGA := CharacterTable( "6.L3(4).2_3" );;
 gap> name  := "new12_2.L3(4).2_3";;
 gap> lib   := CharacterTable( "12_2.L3(4).2_3" );;
 gap> poss  := ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );;
 #E  4 possibilities for new12_2.L3(4).2_3
 gap> Length( poss );
 4
 gap> nsg:= ClassPositionsOfNormalSubgroups( poss[1].table );
 [ [ 1 ], [ 1, 5 ], [ 1, 7 ], [ 1, 4 .. 7 ], [ 1, 3 .. 7 ],
   [ 1 .. 7 ], [ 1 .. 50 ], [ 1 .. 62 ] ]
 gap> List( nsg, x -> Sum( SizesConjugacyClasses( poss[1].table ){ x } ) );
 [ 1, 3, 2, 4, 6, 12, 241920, 483840 ]
 gap> factlib:= CharacterTable( "4_2.L3(4).2_3" );;
 gap> List( poss, x -> IsRecord( TransformingPermutationsCharacterTables(
 >                       x.table / [ 1, 5 ], factlib ) ) );
 [ false, true, false, false ]
 gap> IsRecord( TransformingPermutationsCharacterTables( poss[2].table,
 >                 lib ) );
 true
 gap> ConstructModularMGATables( tblMG, tblGA, lib );;
```

2.4.6 The Character Tables of $12_1.U_4(3).2'_2$ and $12_2.U_4(3).2'_3$ (December 2015)

In the construction of the character table of $M.G.A = 12_1.U_4(3).2'_2$ from the tables of $M.G = 12_1.U_4(3)$ and $G.A = 2.U_4(3).2'_2$, the action of A on the classes of $M.G$ is uniquely determined, but we get two possible character tables.

   (Note that the groups $2.U_4(3).2_2$ and $2.U_4(3).2'_2$ are isomorphic, but we have to take the latter one because the stored factor fusion from $12_1.U_4(3)$ to $2.U_4(3)$ must be combined with the class fusion from $2.U_4(3)$ to $2.U_4(3).2'_2$; using the library table of $2.U_4(3).2_2$ would be technically more involved.)

```
━━━━━━━━━━━━━━━━━━ Example ━━━━━━━━━━━━━━━━━━
 gap> tblMG := CharacterTable( "12_1.U4(3)" );;
 gap> tblG  := CharacterTable( "2.U4(3)" );;
 gap> tblGA := CharacterTable( "2.U4(3).2_2'" );;
 gap> name  := "new12_1.U4(3).2_2'";;
 gap> lib   := CharacterTable( "12_1.U4(3).2_2'" );;
 gap> poss  := ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );;
 #E  2 possibilities for new12_1.U4(3).2_2'
 gap> ConstructModularMGATables( tblMG, tblGA, lib );;
```

   This is not surprising, the two tables involve the two isoclinic variants of $4.U_4(3).2'_2$ (which is isomorphic with $4.U_4(3).2_2$) as tables of factor groups. The irreducible characters of the two tables are equal, only the 2-power map and the element orders are different.

```
━━━━━━━━━━━━━━━━━━ Example ━━━━━━━━━━━━━━━━━━
 gap> Irr( poss[1].table ) = Irr( poss[2].table );
 true
 gap> iso:= CharacterTableIsoclinic( poss[1].table );;
 gap> TransformingPermutationsCharacterTables( iso, poss[2].table );
```

```
  rec( columns := (), group := <permutation group with 5 generators>,
    rows := () )
```

The same phenomenon occurs in the construction of the character table of $M.G.A = 12_2.U_4(3).2'_3$ from the tables of $M.G = 12_2.U_4(3)$ and $G.A = 2.U_4(3).2'_3$.

```
─────────────────────────────── Example ───────────────────────────────
 gap> tblMG := CharacterTable( "12_2.U4(3)" );;
 gap> tblG  := CharacterTable( "2.U4(3)" );;
 gap> tblGA := CharacterTable( "2.U4(3).2_3'" );;
 gap> name  := "new12_2.U4(3).2_3'";;
 gap> lib   := CharacterTable( "12_2.U4(3).2_3'" );;
 gap> poss  := ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );;
 #E  2 possibilities for new12_2.U4(3).2_3'
 gap> ConstructModularMGATables( tblMG, tblGA, lib );;
 gap> iso:= CharacterTableIsoclinic( poss[1].table );;
 gap> TransformingPermutationsCharacterTables( iso, poss[2].table );
 rec( columns := (), group := <permutation group with 8 generators>,
    rows := () )
```

2.4.7   Groups of the Structures $3.U_3(8).3_1$ and $3.U_3(8).6$ (February 2017)

The list of Improvements to the Atlas of Finite Groups [BN95] states the following, concerning the group $G = U_3(8)$.

"There is a unique group of type $3.G.6$ which contains the group of type $3.G.3$ shown. But the (unique) groups of type $3.G.6'$ and $3.G.6''$ contain not this $3.G.3$ but its *isoclines*."

In this section we will show that this statement is not correct, in the sense that the three isoclinic variants of groups of the structure $3.U_3(8).3_1$ are in fact isomorphic.

As a consequence, there is a unique group of the structure $3.U_3(8).6$, up to isomorphism. Note that otherwise the strange situation of nonisomorphic groups $3.G.6$, $3.G.6'$, and $3.G.6''$ would happen, which would be also not isoclinic because their centres are trivial.

A group of the structure $3.U_3(8).3_1$ can be obtained as the semidirect product G, say, of the group $SU(3,8)$ with the automorphism of the field with 64 elements that raises each field element to its fourth power. Note that the semidirect product of $SU(3,8)$ with the field automorphism that squares each field element yields a group of the structure $3.U_3(8).6$.

First we create a permutation representation of G.

```
─────────────────────────────── Example ───────────────────────────────
 gap> s:= SU(3,8);;
 gap> gens:= GeneratorsOfGroup( s );;
 gap> imgs1:= List( gens, m -> List( m, v -> List( v, x -> x^4 ) ) );;
 gap> imgs2:= List( gens, m -> List( m, v -> List( v, x -> x^16 ) ) );;
 gap> f:= GF(64);;
 gap> mats:= List( gens, m -> IdentityMat( 9, f ) );;
 gap> for i in [ 1 .. Length( gens ) ] do
 >      mats[i]{ [ 1 .. 3 ] }{ [ 1 .. 3 ] }:= gens[i];
 >      mats[i]{ [ 4 .. 6 ] }{ [ 4 .. 6 ] }:= imgs1[i];
 >      mats[i]{ [ 7 .. 9 ] }{ [ 7 .. 9 ] }:= imgs2[i];
 >    od;
 gap> fieldaut:= NullMat( 9, 9, f );;
 gap> fieldaut{ [ 4 .. 6 ] }{ [ 1 .. 3 ] }:= IdentityMat( 3, f );;
```

```
gap> fieldaut{ [ 7 .. 9 ] }{ [ 4 .. 6 ] }:= IdentityMat( 3, f );;
gap> fieldaut{ [ 1 .. 3 ] }{ [ 7 .. 9 ] }:= IdentityMat( 3, f );;
gap> v:= [ 1, 0, 0, 1, 0, 0, 1, 0, 0 ] * One( f );;
gap> g:= Group( Concatenation( mats, [ fieldaut ] ) );;
gap> orb:= Orbit( g, v );;
gap> Length( orb );
32319
gap> act:= Action( g, orb );;
gap> Size( act ) = 3 * Size( s );
true
gap> sm:= SmallerDegreePermutationRepresentation( act );;
gap> NrMovedPoints( Image( sm ) );
4617
gap> g:= Image( sm );;
```

The next step is the construction of the central product of G and a cyclic group of order nine, of the structure $3.(3 \times U_3(8).3_1)$. We could try to create the factor group of $9 \times 3.U_3(8).3_1$ modulo a diagonal subgroup of order three, by just applying the / operation. Since GAP would need too much time for that, and since we know better in which situation we are, we create the desired action directly on suitable sets on pairs.

```
——————————————— Example ———————————————
gap> c:= CyclicGroup( IsPermGroup, 9 );;
gap> dp:= DirectProduct( g, c );;
gap> u:= Image( Embedding( dp, 1 ) );;
gap> c:= Image( Embedding( dp, 2 ) );;
gap> c3:= c.1^3;
(4618,4621,4624)(4619,4622,4625)(4620,4623,4626)
gap> z:= Centre( u );;
gap> Size( z );  Length( GeneratorsOfGroup( z ) );
3
1
gap> diag:= Subgroup( dp, [ c3 * z.1 ] );;
gap> orb:= Orbit( dp, [ 1, 4618 ], OnPairs );;
gap> Length( orb );
41553
gap> orb:= Set( orb );;
gap> orbs:= List( OrbitsDomain( diag, orb, OnSets ), Set );;
gap> Length( orbs );
13851
gap> cp:= Action( dp, orbs, OnSetsSets );;
gap> Size( cp );
148925952
```

The three isoclinic variants of the structure $3.U_3(8).3_1$ appear as subgroups of index three in this central product. (The fourth subgroup of index three is of course a central product of the structure $3.(3 \times U_3(8))$.)

```
——————————————— Example ———————————————
gap> der:= DerivedSubgroup( cp );;
gap> Index( cp, der );
9
```

```
gap> inter:= IntermediateSubgroups( cp, der ).subgroups;;
gap> z:= Centre( cp );;
gap> Size( z );
9
gap> inter:= Filtered( inter, x -> not IsSubset( x, z ) );;
gap> List( inter, Size );
[ 49641984, 49641984, 49641984 ]
```

Finally, we check that the three groups are isomorphic.

```
――――――― Example ―――――――
gap> IsomorphismGroups( inter[1], inter[2] ) <> fail;
true
gap> IsomorphismGroups( inter[1], inter[3] ) <> fail;
true
```

Remark:

An indication that the groups might be isomorphic is the fact that their character tables are equivalent, which can be shown much easier, as follows.

```
――――――― Example ―――――――
gap> t1:= CharacterTable( "3.U3(8).3_1" );;
gap> t2:= CharacterTableIsoclinic( t1, rec( k:= 1 ) );;
gap> t3:= CharacterTableIsoclinic( t1, rec( k:= 2 ) );;
gap> TransformingPermutationsCharacterTables( t1, t2 ) <> fail;
true
gap> TransformingPermutationsCharacterTables( t1, t3 ) <> fail;
true
```

2.4.8   The Character Table of $(2^2 \times F_4(2)) : 2 < B$ (March 2003)

The sporadic simple group B contains a maximal subgroup $\overline{N}$ of the type $(2^2 \times F_4(2)) : 2$, which is the normalizer of a 2C element $\overline{x}$ in B (see [CCN+85, p. 217]).

We will see below that the normal Klein four group V in $\overline{N}$ contains two 2A elements in B. The 2A centralizer in B, a group of the structure $2.{}^2E_6(2).2$, contains maximal subgroups of the type $2^2 \times F_4(2)$. So the two 2A type subgroups C_1, C_2 in V are conjugate in $\overline{N}$, and $Z = \langle x \rangle$ is the centre of $\overline{N}$.
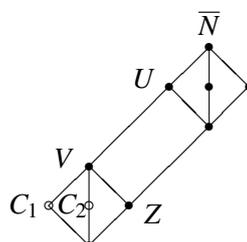
We start with computing the class fusion of the $2^2 \times F_4(2)$ type subgroup U of $\overline{N}$ into B; in order to speed this up, we first compute the class fusion of the $F_4(2)$ subgroup of U into B (which is unique), and use it and the stored embedding into U for prescribing an approximation of the desired class fusion. Additionally, we prescribe (without loss of generality) that the *first* involution class in V is mapped to the class 2C of B.

```
                           ─── Example ───
gap> f42:= CharacterTable( "F4(2)" );;
gap> v4:= CharacterTable( "2^2" );;
gap> dp:= v4 * f42;
CharacterTable( "V4xF4(2)" )
gap> b:= CharacterTable( "B" );;
gap> f42fusb:= PossibleClassFusions( f42, b );;
gap> Length( f42fusb );
1
gap> f42fusdp:= GetFusionMap( f42, dp );;
gap> comp:= CompositionMaps( f42fusb[1], InverseMap( f42fusdp ) );
[ 1, 3, 3, 3, 5, 6, 6, 7, 9, 9, 9, 9, 14, 14, 13, 13, 10, 14, 14, 12,
  14, 17, 15, 18, 22, 22, 22, 22, 26, 26, 22, 22, 27, 27, 28, 31, 31,
  39, 39, 36, 36, 33, 33, 39, 39, 35, 41, 42, 47, 47, 49, 49, 49, 58,
  58, 56, 56, 66, 66, 66, 66, 58, 58, 66, 66, 69, 69, 60, 72, 72, 75,
  79, 79, 81, 81, 85, 86, 83, 83, 91, 91, 94, 94, 104, 104, 109, 109,
  116, 116, 114, 114, 132, 132, 140, 140 ]
gap> v4fusdp:= GetFusionMap( v4, dp );
[ 1, 96 .. 286 ]
gap> comp[ v4fusdp[2] ]:= 4;;
gap> dpfusb:= PossibleClassFusions( dp, b, rec( fusionmap:= comp ) );;
gap> Length( dpfusb );
4
gap> Set( dpfusb, x -> x{ v4fusdp } );
[ [ 1, 4, 2, 2 ] ]
```

As announced above, we see that *V* contains two 2A involutions.

Set $G = U/Z$, $M.G = U$, and $G.A = \overline{N}/Z$. The latter group is the direct product of $F_4(2).2$ and a cyclic group of order 2. Next we compute the class fusion from *G* into *G.A*.

```
                           ─── Example ───
gap> tblG:= dp / v4fusdp{ [ 1, 2 ] };;
gap> tblMG:= dp;;
gap> c2:= CharacterTable( "Cyclic", 2 );;
gap> tblGA:= c2 * CharacterTable( "F4(2).2" );
CharacterTable( "C2xF4(2).2" )
gap> GfusGA:= PossibleClassFusions( tblG, tblGA );;
gap> Length( GfusGA );
4
gap> Length( RepresentativesFusions( tblG, GfusGA, tblGA ) );
1
```

In principle, we have to be careful which of these equivalent maps we choose, since the underlying symmetries may be broken in the central extension $M.G \to G$, for which we choose the default factor fusion.

However, in this situation the fusion *G* into *G.A* is unique already up to table automorphisms of the table of *G.A*, so we are free to choose one map.

```
                           ─── Example ───
gap> Length( RepresentativesFusions( Group( () ), GfusGA, tblGA ) );
1
gap> StoreFusion( tblG, GfusGA[1], tblGA );
```

The tables involved determine the character table of $M.G.A \cong \overline{N}$ uniquely.

```
—————————————— Example ——————————————
gap> elms:= PossibleActionsForTypeMGA( tblMG, tblG, tblGA );;
gap> Length( elms );
1
gap> poss:= PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA, elms[1],
>              "(2^2xF4(2)):2" );;
gap> Length( poss );
1
gap> tblMGA:= poss[1].table;;
```

Finally, we compare the table we constructed with the one that is contained in the GAP Character Table Library.

```
—————————————— Example ——————————————
gap> IsRecord( TransformingPermutationsCharacterTables( tblMGA,
>              CharacterTable( "(2^2xF4(2)):2" ) ) );
true
```

2.4.9   The Character Table of $2.(S_3 \times Fi_{22}.2) < 2.B$ (**March 2003**)

The sporadic simple group B contains a maximal subgroup $\overline{M}$ of type $S_3 \times Fi_{22}.2$. In order to compute the character table of its preimage M in the Schur cover $2.B$, we first analyse the structure of M and then describe the construction of the character table from known character tables.

Let Z denote the centre of $2.B$. We start with $\overline{M} = M/Z$. Its class fusion into B is uniquely determined by the character tables.

```
—————————————— Example ——————————————
gap> s3:= CharacterTable( "Dihedral", 6 );;
gap> fi222:= CharacterTable( "Fi22.2" );;
gap> tblMbar:= s3 * fi222;;
gap> b:= CharacterTable( "B" );;
gap> Mbarfusb:= PossibleClassFusions( tblMbar, b );;
gap> Length( Mbarfusb );
1
```

The subgroup of type Fi_{22} lifts to the double cover $2.Fi_{22}$ (that is, a group that is *not* a direct product $2 \times Fi_{22}$) in $2.B$ since $2.B$ admits no class fusion from Fi_{22}.

```
—————————————— Example ——————————————
gap> 2b:= CharacterTable( "2.B" );;
gap> PossibleClassFusions( CharacterTable( "Fi22" ), 2b );
[  ]
```

So the preimage of $Fi_{22}.2$ is one of the two nonisomorphic but isoclinic groups of type $2.Fi_{22}.2$, and we have to decide which one really occurs. For that, we consider the subgroup of type $3 \times Fi_{22}.2$ in B, which is a 3A centralizer in B. Its preimage has the structure $3 \times 2.Fi_{22}.2$ because the preimage of the central group of order 3 is a cyclic group of order 6 and thus contains a normal complement of the $2.Fi_{22}$ type subgroup. And a class fusion into $2.B$ is possible only from the direct product containing the $2.Fi_{22}.2$ group that is printed in the Atlas.

```
─────────────────────── Example ───────────────────────
gap> c3:= CharacterTable( "Cyclic", 3 );;
gap> 2fi222:= CharacterTable( "2.Fi22.2" );;
gap> PossibleClassFusions( c3 * CharacterTableIsoclinic( 2fi222 ), 2b );
[  ]
```

Next we note that the involutions in the normal subgroup $\overline{S}$ of type S_3 in $\overline{M}$ lift to involutions in $2.B$.

```
─────────────────────── Example ───────────────────────
gap> s3inMbar:= GetFusionMap( s3, tblMbar );
[ 1, 113 .. 225 ]
gap> s3inb:= Mbarfusb[1]{ s3inMbar };
[ 1, 6, 2 ]
gap> 2bfusb:= GetFusionMap( 2b, b );;
gap> 2s3in2B:= InverseMap( 2bfusb ){ s3inb };
[ [ 1, 2 ], [ 8, 9 ], 3 ]
gap> CompositionMaps( OrdersClassRepresentatives( 2b ), 2s3in2B );
[ [ 1, 2 ], [ 3, 6 ], 2 ]
```

Thus the preimage S of $\overline{S}$ contains elements of order 6 but no elements of order 4, which implies that S is a direct product $2 \times S_3$.

The two complements C_1, C_2 of Z in S are normal in the preimage N of $\overline{N} = S_3 \times Fi_{22}$, which is thus of type $S_3 \times 2.Fi_{22}$. However, they are conjugate under the action of $2.Fi_{22}.2$, as no class fusion from $S_3 \times 2.Fi_{22}.2$ into $2.B$ is possible.

```
─────────────────────── Example ───────────────────────
gap> PossibleClassFusions( s3 * 2fi222, 2b );
[  ]
```

(More specifically, the classes of element order 36 in $2.Fi_{22}.2$ have centralizer orders 36 and 72, so their centralizer orders in $S_3 \times 2.Fi_{22}.2$ are 216 and 432; but the centralizers of order 36 elements in $2.B$ have centralizer order at most 216.)

Now let us see how the character table of M can be constructed.

Let Y denote the normal subgroup of order 3 in M, and U its centralizer in M, which has index 2 in M. Then the character table of M is determined by the tables of M/Y, U, $U/Y \cong 2.Fi_{22}.2$, and the action of M on the classes of U.

As for M/Y, consider the normal subgroup $N = N_M(C_1)$ of index 2 in M. In particular, S/Y is central in N/Y but not in M/Y, so the character table of M/Y is determined by the tables of $M/(YZ)$, $N/Y \cong 2 \times 2.Fi_{22}$, $N/(YZ) \cong 2 \times Fi_{22}$, and the action of M/Y on the classes of N/Y.

Thus we proceed in two steps, starting with the computation of the character table of M/Y, for which we choose the name according to the structure $2^2.Fi_{22}.2$.
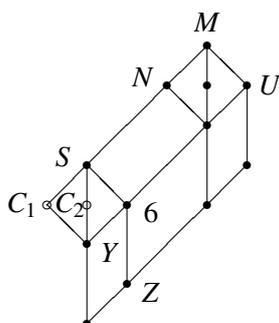
```
───────────────────── Example ─────────────────────
gap> c2:= CharacterTable( "Cyclic", 2 );;
gap> 2fi22:= CharacterTable( "2.Fi22" );;
gap> tblNmodY:= c2 * 2fi22;;
gap> centre:= GetFusionMap( 2fi22, tblNmodY ){
>                 ClassPositionsOfCentre( 2fi22 ) };
[ 1, 2 ]
gap> tblNmod6:= tblNmodY / centre;;
gap> tblMmod6:= c2 * fi222;;
gap> fus:= PossibleClassFusions( tblNmod6, tblMmod6 );;
gap> Length( fus );
1
gap> StoreFusion( tblNmod6, fus[1], tblMmod6 );
gap> elms:= PossibleActionsForTypeMGA( tblNmodY, tblNmod6, tblMmod6 );;
gap> Length( elms );
1
gap> poss:= PossibleCharacterTablesOfTypeMGA( tblNmodY, tblNmod6, tblMmod6,
>               elms[1], "2^2.Fi22.2" );;
gap> Length( poss );
1
gap> tblMmodY:= poss[1].table;
CharacterTable( "2^2.Fi22.2" )
```

So we found a unique solution for the character table of M/Y. Now we compute the table of M. For that, we have to specify the class fusion of U/Y into M/Y; it is unique up to table automorphisms of M/Y.

```
───────────────────── Example ─────────────────────
gap> tblU:= c3 * 2fi222;;
gap> tblUmodY:= tblU / GetFusionMap( c3, tblU );;
gap> fus:= PossibleClassFusions( tblUmodY, tblMmodY );;
gap> Length( RepresentativesFusions( Group( () ), fus, tblMmodY ) );
1
gap> StoreFusion( tblUmodY, fus[1], tblMmodY );
gap> elms:= PossibleActionsForTypeMGA( tblU, tblUmodY, tblMmodY );;
gap> Length( elms );
1
gap> poss:= PossibleCharacterTablesOfTypeMGA( tblU, tblUmodY, tblMmodY,
>               elms[1], "(S3x2.Fi22).2" );;
gap> Length( poss );
1
gap> tblM:= poss[1].table;
CharacterTable( "(S3x2.Fi22).2" )
gap> mfus2b:= PossibleClassFusions( tblM, 2b );;
gap> Length( RepresentativesFusions( tblM, mfus2b, 2b ) );
1
```

We did not construct M as a central extension of $\overline{M}$, so we verify that the tables fit together; note that this way we get also the class fusion from M onto $\overline{M}$.

```
───────────────────── Example ─────────────────────
gap> Irr( tblM / ClassPositionsOfCentre( tblM ) ) = Irr( tblMbar );
true
```

Finally, we compare the table we constructed with the one that is contained in the GAP Character Table Library.

```
                              Example
  gap> IsRecord( TransformingPermutationsCharacterTables( tblM,
  >                 CharacterTable( "(S3x2.Fi22).2" ) ) );
  true
```

2.4.10   The Character Table of $(2 \times 2.Fi_{22}) : 2 < Fi_{24}$ (November 2008)

The automorphism group Fi_{24} of the sporadic simple group Fi'_{24} contains a maximal subgroup N of the type $(2 \times 2.Fi_{22}) : 2$, whose intersection with Fi'_{24} is $2.Fi_{22}.2$ (see [CCN+85, p. 207]).

The normal Klein four group V in N contains two 2C elements in Fi_{24}, because the 2C centralizer in Fi_{24}, a group of the structure $2 \times Fi_{23}$, contains maximal subgroups of the type $2 \times 2.Fi_{22}$, and so the two 2C type subgroups C_1, C_2 in V are conjugate in N, and $Z = Z(N)$ is the centre of $N \cap Fi'_{24}$.
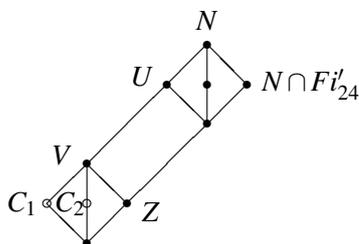
With $U = C_N(C_1)$, a group of the type $2 \times 2.Fi_{22}$, we set $G = U/Z$, $M.G = U$, and $G.A = N/Z$. The latter group is the direct product of $Fi_{22}.2$ and a cyclic group of order 2.

This is exactly the situation of the construction of the character table of the group that is called $2^2.Fi_{22}.2$ in Section 2.4.9, where this group occurs as "M/Y". Since the character table is uniquely determined by the input data, it is the table we are interested in here.

So all we have to do is to compute the class fusion from this table into that of Fi_{24}.

```
                              Example
  gap> fi24:= CharacterTable( "Fi24" );;
  gap> t:= CharacterTable( "2^2.Fi22.2" );;
  gap> fus:= PossibleClassFusions( t, fi24 );;
  gap> Length( fus );
  4
  gap> Length( RepresentativesFusions( t, fus, fi24 ) );
  1
```

(It should be noted that we did not need the character table of the $2.Fi_{22}.2$ type subgroup of N in the above construction, only the tables of $2.Fi_{22}$ and $Fi_{22}.2$ were used.)

The fact that the character table of a factor of a subgroup of $2.B$ occurs as the character table of a subgroup of Fi_{24} is not a coincidence. In fact, the groups $3.Fi_{24}$ and $2.B$ are subgroups of the Monster group M, and the subgroup $U = 2.(S_3 \times Fi_{22}.2)$ of $2.B$ normalizes an element of order three. The full normalizer of this element in M is $3.Fi_{24}$, which means that we have established U as a (maximal) subgroup of $3.Fi_{24}$. Note that we have constructed the character table of U in Section 2.4.9.

Let us compute the class fusion of U into $3.Fi_{24}$.

```
                              Example
  gap> t:= CharacterTable( "(S3x2.Fi22).2" );;
  gap> 3fi24:= CharacterTable( "3.Fi24" );;
```

```
gap> fus:= PossibleClassFusions( t, 3fi24 );;
gap> Length( fus );
16
gap> Length( RepresentativesFusions( t, fus, 3fi24 ) );
1
gap> GetFusionMap( t, 3fi24 ) in fus;
true
```

Moreover, U turns out to be the full normalizer of a 6A element in M,

———————————————— Example ————————————————
```
gap> m:= CharacterTable( "M" );;
gap> tfusm:= PossibleClassFusions( t, m );;
gap> Length( tfusm );
4
gap> Length( RepresentativesFusions( t, tfusm, m ) );
1
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>        x -> Sum( SizesConjugacyClasses( t ){ x } ) = 6 );
[ [ 1, 2, 142, 143 ] ]
gap> Set( tfusm, x -> x{ nsg[1] } );
[ [ 1, 2, 4, 13 ] ]
gap> OrdersClassRepresentatives( t ){ nsg[1] };
[ 1, 2, 3, 6 ]
gap> PowerMap( m, -1 )[13];
13
gap> Size( t ) = 2 * SizesCentralizers( m )[13];
true
```

(Thus U is also the full normalizer of an element of order six in $2.B$ and in $3.Fi_{24}$.)

2.4.11   The Character Table of $S_3 \times 2.U_4(3).2_2 \leq 2.Fi_{22}$ (September 2002)

The sporadic simple Fischer group Fi_{22} contains a maximal subgroup $\overline{M}$ of type $S_3 \times U_4(3).2_2$ (see [CCN+85, p. 163]). We claim that the preimage M of $\overline{M}$ in the central extension $2.Fi_{22}$ has the structure $S_3 \times 2.U_4(3).2_2$, where the factor of type $2.U_4(3).2_2$ is the one printed in the Atlas.

For that, we first note that the normal subgroup $\overline{S}$ of type S_3 in $\overline{M}$ lifts to a group S which has the structure $2 \times S_3$. This follows from the fact that all involutions in Fi_{22} lift to involutions in $2.Fi_{22}$ or, equivalently, the central involution in $2.Fi_{22}$ is not a square.
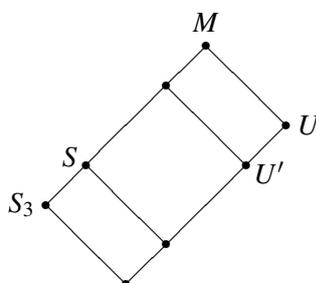
———————————————— Example ————————————————
```
gap> 2Fi22:= CharacterTable( "2.Fi22" );;
gap> ClassPositionsOfCentre( 2Fi22 );
```

```
  [ 1, 2 ]
gap> 2 in PowerMap( 2Fi22, 2 );
false
```

Second, the normal subgroup $\overline{U} \cong U_4(3).2_2$ of Fi_{22} lifts to a nonsplit extension U in $2.Fi_{22}$, since $2.Fi_{22}$ contains no $U_4(3)$ type subgroup. Furthermore, U is the $2.U_4(3).2_2$ type group printed in the Atlas because the isoclinic variant does not admit a class fusion into $2.Fi_{22}$.

```
————————————————— Example —————————————————
gap> PossibleClassFusions( CharacterTable( "U4(3)" ), 2Fi22 );
[ ]
gap> tblU:= CharacterTable( "2.U4(3).2_2" );;
gap> iso:= CharacterTableIsoclinic( tblU );
CharacterTable( "Isoclinic(2.U4(3).2_2)" )
gap> PossibleClassFusions( iso, 2Fi22 );
[  ]
```

Now there are just two possibilities. Either the two S_3 type subgroups in S are normal in M (and thus M is the direct product of any such S_3 with the preimage of the $U_4(3).2_2$ type subgroup), or they are conjugate in M.

Suppose we are in the latter situation, let z be a generator of the centre of $2.Fi_{22}$, and let τ, σ be an involution and an order three element respectively, in one of the S_3 type subgroups.

Each element $g \in U \setminus U'$ conjugates τ to an involution in the other S_3 type subgroup of S, so $g^{-1}\tau g = \tau\sigma^i z$ for some $i \in \{0, 1, 2\}$. Furthermore, it is possible to choose g as an involution.

```
————————————————— Example —————————————————
gap> derpos:= ClassPositionsOfDerivedSubgroup( tblU );;
gap> outer:= Difference( [ 1 .. NrConjugacyClasses( tblU ) ], derpos );;
gap> 2 in OrdersClassRepresentatives( tblU ){ outer };
true
```

With this choice, $(g\tau)^2 = \tau\sigma^i z\tau = \sigma^{-i}z$ holds, which means that $(g\tau)^3$ squares to z. As we have seen above, this is impossible, hence M is a direct product, as claimed.

The class fusion of M into $2.Fi_{22}$ is determined by the character tables, up to table automorphisms.

```
————————————————— Example —————————————————
gap> tblM:= CharacterTable( "Dihedral", 6 ) * tblU;;
gap> fus:= PossibleClassFusions( tblM, 2Fi22 );;
gap> Length( RepresentativesFusions( tblM, fus, 2Fi22 ) );
1
gap> IsRecord( TransformingPermutationsCharacterTables( tblM,
>                 CharacterTable( "2.Fi22M8" ) ) );
true
```

2.4.12 The Character Table of $4.HS.2 \leq HN.2$ (May 2002)

The maximal subgroup U of type $2.HS.2$ in the sporadic simple group HN extends to a group N of structure $4.HS.2$ in the automorphism group $HN.2$ of HN (see [CCN$^+$85, p. 166]).

N is the normalizer of a 4D element $g \in HN.2 \setminus HN$. The centralizer C of g is of type $4.HS$, which is the central product of $2.HS$ and the cyclic group $\langle g \rangle$ of order 4. We have $Z = Z(N) = \langle g^2 \rangle$. Since $U/Z \cong HS.2$ is a complement of $\langle g \rangle/Z$ in N/Z, the factor group N/Z is a direct product of $HS.2$ and a cyclic group of order 2.
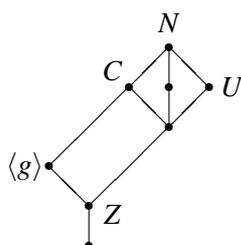
Thus N has the structure $2.G.2$, the normal subgroup $2.G$ being C, the factor group $G.2$ being $2 \times HS.2$, and G being $2 \times HS$. Each element in $N \setminus C$ inverts g, so N acts fixed point freely on the faithful irreducible characters of C. Hence we can use `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) for constructing the character table of N from the tables of C and N/Z and the action of N on the classes of C.

We start with the table of the central product C. It can be viewed as an isoclinic table of the direct product of $2.HS$ and a cyclic group of order 2, see 2.2.4.

```
――――――――――――――――――― Example ―――――――――――――――――――
 gap> c2:= CharacterTable( "Cyclic", 2 );;
 gap> tblC:= CharacterTableIsoclinic( CharacterTable( "2.HS" ) * c2 );;
```

The table of G is given as that of the factor group by the unique normal subgroup of C that consists of two conjugacy classes.

```
――――――――――――――――――― Example ―――――――――――――――――――
 gap> ord2:= Filtered( ClassPositionsOfNormalSubgroups( tblC ),
 >                  x -> Length( x ) = 2 );
 [ [ 1, 3 ] ]
 gap> tblCbar:= tblC / ord2[1];;
```

Finally, we construct the table of the extension $G.2$ and the class fusion of G into this table (which is uniquely determined by the character tables).

```
――――――――――――――――――― Example ―――――――――――――――――――
 gap> tblNbar:= CharacterTable( "HS.2" ) * c2;;
 gap> fus:= PossibleClassFusions( tblCbar, tblNbar );
 [ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
       19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 29, 30, 31, 32,
       33, 34, 35, 36, 35, 36, 37, 38, 39, 40, 41, 42, 41, 42 ] ]
 gap> StoreFusion( tblCbar, fus[1], tblNbar );
```

Now we compute the table automorphisms of the table of C that are compatible with the extension N; we get two solutions.

```
――――――――――――――――――― Example ―――――――――――――――――――
 gap> elms:= PossibleActionsForTypeMGA( tblC, tblCbar, tblNbar );
 [ [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6, 8 ], [ 7 ], [ 9 ], [ 10 ],
       [ 11 ], [ 12, 14 ], [ 13 ], [ 15 ], [ 16, 18 ], [ 17 ], [ 19 ],
       [ 20 ], [ 21 ], [ 22 ], [ 23 ], [ 24, 26 ], [ 25 ], [ 27 ],
       [ 28, 30 ], [ 29 ], [ 31 ], [ 32, 34 ], [ 33 ], [ 35 ],
       [ 36, 38 ], [ 37 ], [ 39 ], [ 40, 42 ], [ 41 ], [ 43 ],
       [ 44, 46 ], [ 45 ], [ 47 ], [ 48, 50 ], [ 49 ], [ 51, 53 ],
       [ 52, 54 ], [ 55 ], [ 56, 58 ], [ 57 ], [ 59 ], [ 60 ],
       [ 61, 65 ], [ 62, 68 ], [ 63, 67 ], [ 64, 66 ], [ 69 ],
       [ 70, 72 ], [ 71 ], [ 73 ], [ 74, 76 ], [ 75 ], [ 77, 81 ],
```

```
        [ 78, 84 ], [ 79, 83 ], [ 80, 82 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6, 8 ], [ 7 ], [ 9 ], [ 10 ],
      [ 11 ], [ 12, 14 ], [ 13 ], [ 15, 17 ], [ 16 ], [ 18 ], [ 19 ],
      [ 20 ], [ 21 ], [ 22 ], [ 23 ], [ 24, 26 ], [ 25 ], [ 27 ],
      [ 28, 30 ], [ 29 ], [ 31 ], [ 32, 34 ], [ 33 ], [ 35, 37 ],
      [ 36 ], [ 38 ], [ 39 ], [ 40, 42 ], [ 41 ], [ 43 ], [ 44, 46 ],
      [ 45 ], [ 47, 49 ], [ 48 ], [ 50 ], [ 51, 53 ], [ 52, 54 ],
      [ 55 ], [ 56, 58 ], [ 57 ], [ 59 ], [ 60 ], [ 61, 65 ],
      [ 62, 68 ], [ 63, 67 ], [ 64, 66 ], [ 69, 71 ], [ 70 ], [ 72 ],
      [ 73 ], [ 74, 76 ], [ 75 ], [ 77, 83 ], [ 78, 82 ], [ 79, 81 ],
      [ 80, 84 ] ] ]
```

We compute the possible character tables arising from these two actions.

```
——————————————— Example ———————————————
gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
>                  tblC, tblCbar, tblNbar, pi, "4.HS.2" ) );;
gap> List( poss, Length );
[ 0, 2 ]
```

So one of the two table automorphisms turned out to be impossible; the reason is that the corresponding "character table" would not admit a 2–power map. (Alternatively, we could exclude this action on C by the fact that it is not compatible with the action of $2.HS.2$ on its subgroup $2.HS$, which occurs here as the restriction of the action of N on C to that of U on $C \cap U$.)

The other table automorphism leads to two possible character tables. This is not surprising since N contains a subgroup of type $2.HS.2$, and the above setup does not determine which of the two isoclinism types of this group occurs. Let us look at the possible class fusions from these tables into that of $HN.2$:

```
——————————————— Example ———————————————
gap> result:= poss[2];;
gap> hn2:= CharacterTable( "HN.2" );;
gap> possfus:= List( result, r -> PossibleClassFusions( r.table, hn2 ) );;
gap> List( possfus, Length );
[ 32, 0 ]
gap> RepresentativesFusions( result[1].table, possfus[1], hn2 );
[ [ 1, 46, 2, 2, 47, 3, 7, 45, 4, 58, 13, 6, 46, 47, 6, 47, 7, 48,
      10, 62, 20, 9, 63, 21, 12, 64, 24, 27, 49, 50, 13, 59, 14, 16,
      70, 30, 18, 53, 52, 17, 54, 20, 65, 22, 36, 56, 26, 76, 39, 77,
      28, 59, 58, 31, 78, 41, 34, 62, 35, 65, 2, 45, 3, 45, 6, 48, 7,
      47, 17, 54, 13, 49, 13, 50, 14, 50, 18, 53, 18, 52, 21, 56, 25,
      57, 27, 59, 30, 60, 44, 72, 34, 66, 35, 66, 41, 71 ] ]
```

Only one of the candidates admits an embedding, and the class fusion is unique up to table automorphisms. So we are done.

Finally, we compare the table we have constructed with the one that is contained in the GAP Character Table Library.

```
——————————————— Example ———————————————
gap> libtbl:= CharacterTable( "4.HS.2" );;
gap> IsRecord( TransformingPermutationsCharacterTables( result[1].table,
>                  libtbl ) );
true
```

(The following paragraphs have been added in May 2006.)

The Brauer tables of $N = 2.G.2$ can be constructed as in Section 2.4.3. Note that the Brauer tables of $C = 2.G$ and of $N/Z = G.2$ are automatically available because the ordinary tables constructed above arose as a direct product and as an isoclinic table of a direct product, and the GAP Character Table Library contains the Brauer tables of the direct factors involved.

```
──────── Example ────────
gap> StoreFusion( tblC, result[1].MGfusMGA, result[1].table );
gap> ForAll( PrimeDivisors( Size( result[1].table ) ),
>           p -> IsRecord( TransformingPermutationsCharacterTables(
>               BrauerTableOfTypeMGA( tblC mod p, tblNbar mod p,
>                   result[1].table ).table, libtbl mod p ) ) );
true
```

Here it is advantageous that the Brauer table of $C/Z = G$ is not needed in the construction, since GAP does not know how to compute the p-modular table of the ordinary table of G constructed above. Of course we have $G \cong 2 \times HS$, and the p-modular table of HS is known, but in the construction of the table of G as a factor of the table of $2.G$, the information is missing that the nonsolvable simple direct factor of $2.G$ corresponds to the library table of HS.

2.4.13 The Character Tables of $4.A_6.2_3$, $12.A_6.2_3$, and $4.L_2(25).2_3$

For the "broken box" cases in the Atlas (see [CCN+85, p. xxiv]), the character tables can be constructed with the $M.G.A$ construction method from Section 2.3.1. (The situation with $9.U_3(8).3_3$ is more complicated, this group will be considered in Section 2.4.16.)

The group $N = 4.A_6.2_3$ (see [CCN+85, p. 5]) can be described as an upward extension of the normal subgroup $C \cong 4.A_6$ –which is a central product of $U = 2.A_6$ and a cyclic group $\langle g \rangle$ of order 4– by a cyclic group of order 2, such that the factor group of N by the central subgroup $Z = \langle g^2 \rangle$ of order 2 is isomorphic to a subdirect product $\overline{N}$ of $M_{10} = A_6.2_3$ and a cyclic group of order 4 and that N acts nontrivially on its normal subgroup $\langle g \rangle$.
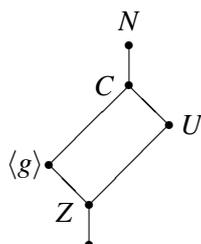
Thus N has the structure $2.G.2$, with $2.G = C$ and $G.2 = \overline{N}$. These two groups are isoclinic variants of $2 \times 2.A_6$ and of $2 \times M_{10}$, respectively. Each element in $N \setminus C$ inverts g, so it acts fixed point freely on the faithful irreducible characters of C. Hence we can use `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) for constructing the character table of N from the tables of C and N/Z and the action of N on the classes of C.

```
──────── Example ────────
gap> c2:= CharacterTable( "Cyclic", 2 );;
gap> 2a6:= CharacterTable( "2.A6" );;
gap> tblC:= CharacterTableIsoclinic( 2a6 * c2 );;
gap> ord2:= Filtered( ClassPositionsOfNormalSubgroups( tblC ),
```

```
>                    x -> Length( x ) = 2 );
[ [ 1, 3 ] ]
gap> tblG:= tblC / ord2[1];;
gap> tblNbar:= CharacterTableIsoclinic( CharacterTable( "A6.2_3" ) * c2 );;
gap> fus:= PossibleClassFusions( tblG, tblNbar );
[ [ 1, 2, 3, 4, 5, 6, 5, 6, 7, 8, 9, 10, 9, 10 ] ]
gap> StoreFusion( tblG, fus[1], tblNbar );
gap> elms:= PossibleActionsForTypeMGA( tblC, tblG, tblNbar );
[ [ [ 1 ], [ 2 ], [ 3 ], [ 4 ], [ 5 ], [ 6 ], [ 7, 11 ], [ 8, 12 ],
      [ 9, 13 ], [ 10, 14 ], [ 15, 17 ], [ 16, 18 ], [ 19, 23 ],
      [ 20, 24 ], [ 21, 25 ], [ 22, 26 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6 ], [ 7, 11 ], [ 8, 14 ],
      [ 9, 13 ], [ 10, 12 ], [ 15 ], [ 16, 18 ], [ 17 ], [ 19, 23 ],
      [ 20, 26 ], [ 21, 25 ], [ 22, 24 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6 ], [ 7, 11 ], [ 8, 14 ],
      [ 9, 13 ], [ 10, 12 ], [ 15, 17 ], [ 16 ], [ 18 ], [ 19, 23 ],
      [ 20, 26 ], [ 21, 25 ], [ 22, 24 ] ] ]
gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
>                    tblC, tblG, tblNbar, pi, "4.A6.2_3" ) );
[ [  ], [  ],
  [
      rec(
          MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 9, 6, 9, 8, 7, 10,
              11, 10, 12, 13, 14, 15, 16, 13, 16, 15, 14 ],
          table := CharacterTable( "4.A6.2_3" ) ) ] ]
```

So we get a unique solution. It coincides with the character table of $4.A_6.2_3$ that is stored in the GAP Character Table Library.

```
                             ── Example ──
gap> t:= poss[3][1].table;;
gap> IsRecord( TransformingPermutationsCharacterTables( t,
>                    CharacterTable( "4.A6.2_3" ) ) );
true
```

Note that the first two candidates for the action lead to tables that do not admit a 2–power map. In fact the 2–power map of the character table of $4.A_6.2_3$ is not uniquely determined by the matrix of character values. However, the 2–power map is unique up to automorphisms of this matrix; the function `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) takes this into account, and returns only representatives, in this case one table.

As is mentioned in the Atlas (see [CCN+85, Section 6.7]), the group $\Gamma L(2,9)$ contains subgroups of the structure $4.A_6.2_3$. We can find them as follows.

```
                             ── Example ──
gap> g:= GammaL(2,9);;
gap> phi:= IsomorphismPermGroup( g );;
gap> img:= Image( phi );;
gap> der:= DerivedSubgroup( img );;
gap> derder:= DerivedSubgroup( der );;
gap> Index( img, derder );
16
gap> inter:= Filtered( IntermediateSubgroups( img, derder ).subgroups,
```

```
>                     s -> Size( s ) = 4 * Size( derder ) and
>                          IsCyclic( CommutatorFactorGroup( s ) ) and
>                          Size( Centre( s ) ) = 2 );;
gap> Length( inter );
2
gap> ForAll( inter, x -> IsConjugate( img, inter[1], x ) );
true
gap> IsRecord( TransformingPermutationsCharacterTables( t,
>                 CharacterTable( inter[1] ) ) );
true
```

The Atlas states in [CCN$^+$85, Section 6.7] that there is a group of the structure $2^2.A_6.2_3$ that is isoclinic with $4.A_6.2_3$. We construct also the character table of the $2^2.A_6.2_3$ type group with the $M.G.A$ construction method from Section 2.3.1.

The group $N = 2^2.A_6.2_3$ can be described as an upward extension of the normal subgroup $C \cong 2 \times 2.A_6$ by a cyclic group of order 2, such that the factor group of N by the central subgroup Z of order 2 that is contained in $U = C' \cong 2.A_6$ is isomorphic to a subdirect product $\overline{N}$ of $M_{10} = A_6.2_3$ and a cyclic group of order 4 and that N acts nontrivially on the centre of C, which is a Klein four group.
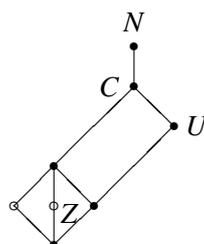
Thus N has the structure $2.G.2$, with $2.G = C$ and $G.2 = \overline{N}$. These latter group is an isoclinic variant of $2 \times M_{10}$, as in the construction of $4.A_6.2_3$. Each element in $N \setminus C$ swaps the two involutions in $Z(C) \setminus Z$, so it acts fixed point freely on those irreducible characters of C whose kernels do not contain Z. Hence we can use `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) for constructing the character table of N from the tables of C and N/Z and the action of N on the classes of C.

```
                              ──── Example ────
gap> tblC:= 2a6 * c2;;
gap> z:= GetFusionMap( 2a6, tblC ){ ClassPositionsOfCentre( 2a6 ) };
[ 1, 3 ]
gap> tblG:= tblC / z;;
gap> tblNbar:= CharacterTableIsoclinic( CharacterTable( "A6.2_3" ) * c2 );;
gap> fus:= PossibleClassFusions( tblG, tblNbar );
[ [ 1, 2, 3, 4, 5, 6, 5, 6, 7, 8, 9, 10, 9, 10 ] ]
gap> StoreFusion( tblG, fus[1], tblNbar );
gap> elms:= PossibleActionsForTypeMGA( tblC, tblG, tblNbar );
[ [ [ 1 ], [ 2 ], [ 3 ], [ 4 ], [ 5 ], [ 6 ], [ 7, 11 ], [ 8, 12 ],
      [ 9, 13 ], [ 10, 14 ], [ 15, 17 ], [ 16, 18 ], [ 19, 23 ],
      [ 20, 24 ], [ 21, 25 ], [ 22, 26 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6 ], [ 7, 11 ], [ 8, 14 ],
      [ 9, 13 ], [ 10, 12 ], [ 15 ], [ 16, 18 ], [ 17 ], [ 19, 23 ],
      [ 20, 26 ], [ 21, 25 ], [ 22, 24 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6 ], [ 7, 11 ], [ 8, 14 ],
```

```
           [ 9, 13 ], [ 10, 12 ], [ 15, 17 ], [ 16 ], [ 18 ], [ 19, 23 ],
           [ 20, 26 ], [ 21, 25 ], [ 22, 24 ] ] ]
gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
>                 tblC, tblG, tblNbar, pi, "2^2.A6.2_3" ) );
[ [  ], [  ],
  [
      rec(
          MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 9, 6, 9, 8, 7, 10,
              11, 10, 12, 13, 14, 15, 16, 13, 16, 15, 14 ],
          table := CharacterTable( "2^2.A6.2_3" ) ) ] ]
```

So we get a unique solution.

The group $N = 12.A_6.2_3$ (see [CCN+85, p. 5]) can be described as an upward extension of the normal subgroup $C \cong 12.A_6$ –which is a central product of $U = 6.A_6$ and a cyclic group $\langle g \rangle$ of order 4– by a cyclic group of order 2, such that the factor group of N by the central subgroup $Z = \langle g^2 \rangle$ of order 2 is isomorphic to a subdirect product $\overline{N}$ of $3.M_{10} = 3.A_6.2_3$ and a cyclic group of order 4 and that N acts nontrivially on its normal subgroup $\langle g \rangle$.

Note that N has a central subgroup Y, say, of order 3, so the situation here differs from that for groups of the type $12.G.2$ with G one of $L_3(4)$, $U_4(3)$, where the action on the normal subgroup of order three is nontrivial.
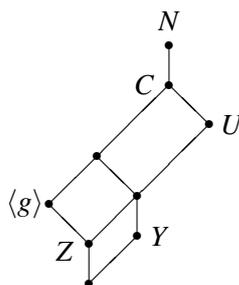
Thus N has the structure $2.G.2$, with $2.G = C$ and $G.2 = \overline{N}$. These two groups are isoclinic variants of $2 \times 6.A_6$ and of $2 \times 3.M_{10}$, respectively. Each element in $N \setminus C$ inverts g, so it acts fixed point freely on the faithful irreducible characters of C. Hence we can use `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) for constructing the character table of N from the tables of C and N/Z and the action of N on the classes of C.

```
──────────── Example ────────────
gap> c2:= CharacterTable( "Cyclic", 2 );;
gap> tblC:= CharacterTableIsoclinic( CharacterTable( "6.A6" ) * c2 );;
gap> ord2:= Filtered( ClassPositionsOfNormalSubgroups( tblC ),
>               x -> Length( x ) = 2 );
[ [ 1, 7 ] ]
gap> tblG:= tblC / ord2[1];;
gap> tblNbar:= CharacterTableIsoclinic( CharacterTable( "3.A6.2_3" ) * c2 );;
gap> fus:= PossibleClassFusions( tblG, tblNbar );
[ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 13, 14, 15, 16,
      17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 21, 22, 23, 24, 25, 26 ]
    ,
  [ 1, 2, 5, 6, 3, 4, 7, 8, 11, 12, 9, 10, 13, 14, 13, 14, 15, 16,
      19, 20, 17, 18, 21, 22, 25, 26, 23, 24, 21, 22, 25, 26, 23, 24
      ] ]
```

```
gap> rep:= RepresentativesFusions( Group( () ), fus, tblNbar );
[ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 13, 14, 15, 16,
      17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 21, 22, 23, 24, 25, 26
    ] ]
gap> StoreFusion( tblG, rep[1], tblNbar );
gap> elms:= PossibleActionsForTypeMGA( tblC, tblG, tblNbar );
[ [ [ 1 ], [ 2 ], [ 3 ], [ 4 ], [ 5 ], [ 6 ], [ 7 ], [ 8 ], [ 9 ],
      [ 10 ], [ 11 ], [ 12 ], [ 13 ], [ 14 ], [ 15 ], [ 16 ], [ 17 ],
      [ 18 ], [ 19, 23 ], [ 20, 24 ], [ 21, 25 ], [ 22, 26 ],
      [ 27, 33 ], [ 28, 34 ], [ 29, 35 ], [ 30, 36 ], [ 31, 37 ],
      [ 32, 38 ], [ 39, 51 ], [ 40, 52 ], [ 41, 53 ], [ 42, 54 ],
      [ 43, 55 ], [ 44, 56 ], [ 45, 57 ], [ 46, 58 ], [ 47, 59 ],
      [ 48, 60 ], [ 49, 61 ], [ 50, 62 ] ],
  [ [ 1 ], [ 2, 8 ], [ 3 ], [ 4, 10 ], [ 5 ], [ 6, 12 ], [ 7 ],
      [ 9 ], [ 11 ], [ 13 ], [ 14 ], [ 15 ], [ 16 ], [ 17 ], [ 18 ],
      [ 19, 23 ], [ 20, 26 ], [ 21, 25 ], [ 22, 24 ], [ 27 ],
      [ 28, 34 ], [ 29 ], [ 30, 36 ], [ 31 ], [ 32, 38 ], [ 33 ],
      [ 35 ], [ 37 ], [ 39, 51 ], [ 40, 58 ], [ 41, 53 ], [ 42, 60 ],
      [ 43, 55 ], [ 44, 62 ], [ 45, 57 ], [ 46, 52 ], [ 47, 59 ],
      [ 48, 54 ], [ 49, 61 ], [ 50, 56 ] ],
  [ [ 1 ], [ 2, 8 ], [ 3 ], [ 4, 10 ], [ 5 ], [ 6, 12 ], [ 7 ],
      [ 9 ], [ 11 ], [ 13 ], [ 14 ], [ 15 ], [ 16 ], [ 17 ], [ 18 ],
      [ 19, 23 ], [ 20, 26 ], [ 21, 25 ], [ 22, 24 ], [ 27, 33 ],
      [ 28 ], [ 29, 35 ], [ 30 ], [ 31, 37 ], [ 32 ], [ 34 ], [ 36 ],
      [ 38 ], [ 39, 51 ], [ 40, 58 ], [ 41, 53 ], [ 42, 60 ],
      [ 43, 55 ], [ 44, 62 ], [ 45, 57 ], [ 46, 52 ], [ 47, 59 ],
      [ 48, 54 ], [ 49, 61 ], [ 50, 56 ] ] ]
gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
>                  tblC, tblG, tblNbar, pi, "12.A6.2_3" ) );
[ [ ], [ ],
  [
      rec(
          MGfusMGA := [ 1, 2, 3, 4, 5, 6, 7, 2, 8, 4, 9, 6, 10, 11, 12,
              13, 14, 15, 16, 17, 18, 19, 16, 19, 18, 17, 20, 21, 22,
              23, 24, 25, 20, 26, 22, 27, 24, 28, 29, 30, 31, 32, 33,
              34, 35, 36, 37, 38, 39, 40, 29, 36, 31, 38, 33, 40, 35,
              30, 37, 32, 39, 34 ],
          table := CharacterTable( "12.A6.2_3" ) ) ] ]
```

So we get again a unique solution. It coincides with the character table that is stored in the GAP Character Table Library.

```
  _____ Example _____
gap> IsRecord( TransformingPermutationsCharacterTables( poss[3][1].table,
>                  CharacterTable( "12.A6.2_3" ) ) );
true
```

The construction of the character table of $4.L_2(25).2_3$ is analogous to that of the table of $4.A_6.2_3$. We get a unique table that coincides with the table in the GAP library.

```
  _____ Example _____
gap> c2:= CharacterTable( "Cyclic", 2 );;
gap> tblC:= CharacterTableIsoclinic( CharacterTable( "2.L2(25)" ) * c2 );;
```

```
gap> ord2:= Filtered( ClassPositionsOfNormalSubgroups( tblC ),
>                 x -> Length( x ) = 2 );
[ [ 1, 3 ] ]
gap> tblG:= tblC / ord2[1];;
gap> tblNbar:= CharacterTableIsoclinic( CharacterTable( "L2(25).2_3" ) * c2 );;
gap> fus:= PossibleClassFusions( tblG, tblNbar );
[ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 10, 11, 12, 13, 14, 13, 14, 15,
      16, 15, 16, 17, 18, 17, 18, 19, 20, 19, 20 ],
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 10, 11, 12, 13, 14, 13, 14, 17,
      18, 17, 18, 19, 20, 19, 20, 15, 16, 15, 16 ],
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 10, 11, 12, 13, 14, 13, 14, 19,
      20, 19, 20, 15, 16, 15, 16, 17, 18, 17, 18 ] ]
gap> rep:= RepresentativesFusions( Group( () ), fus, tblNbar );
[ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 9, 10, 11, 12, 13, 14, 13, 14, 15,
      16, 15, 16, 17, 18, 17, 18, 19, 20, 19, 20 ] ]
gap> StoreFusion( tblG, rep[1], tblNbar );
gap> elms:= PossibleActionsForTypeMGA( tblC, tblG, tblNbar );
[ [ [ 1 ], [ 2 ], [ 3 ], [ 4 ], [ 5 ], [ 6 ], [ 7 ], [ 8 ], [ 9 ],
      [ 10 ], [ 11, 13 ], [ 12, 14 ], [ 15, 19 ], [ 16, 20 ],
      [ 17, 21 ], [ 18, 22 ], [ 23, 25 ], [ 24, 26 ], [ 27, 33 ],
      [ 28, 34 ], [ 29, 31 ], [ 30, 32 ], [ 35, 39 ], [ 36, 40 ],
      [ 37, 41 ], [ 38, 42 ], [ 43, 47 ], [ 44, 48 ], [ 45, 49 ],
      [ 46, 50 ], [ 51, 55 ], [ 52, 56 ], [ 53, 57 ], [ 54, 58 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6 ], [ 7 ], [ 8, 10 ], [ 9 ],
      [ 11 ], [ 12, 14 ], [ 13 ], [ 15, 19 ], [ 16, 22 ], [ 17, 21 ],
      [ 18, 20 ], [ 23, 25 ], [ 24 ], [ 26 ], [ 27, 31 ], [ 28, 34 ],
      [ 29, 33 ], [ 30, 32 ], [ 35, 39 ], [ 36, 42 ], [ 37, 41 ],
      [ 38, 40 ], [ 43, 47 ], [ 44, 50 ], [ 45, 49 ], [ 46, 48 ],
      [ 51, 55 ], [ 52, 58 ], [ 53, 57 ], [ 54, 56 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5 ], [ 6 ], [ 7 ], [ 8, 10 ], [ 9 ],
      [ 11, 13 ], [ 12 ], [ 14 ], [ 15, 19 ], [ 16, 22 ], [ 17, 21 ],
      [ 18, 20 ], [ 23, 25 ], [ 24 ], [ 26 ], [ 27, 33 ], [ 28, 32 ],
      [ 29, 31 ], [ 30, 34 ], [ 35, 39 ], [ 36, 42 ], [ 37, 41 ],
      [ 38, 40 ], [ 43, 47 ], [ 44, 50 ], [ 45, 49 ], [ 46, 48 ],
      [ 51, 55 ], [ 52, 58 ], [ 53, 57 ], [ 54, 56 ] ] ]
gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
>                 tblC, tblG, tblNbar, pi, "4.L2(25).2_3" ) );
[ [ ], [ ],
  [
      rec(
          MGfusMGA := [ 1, 2, 3, 2, 4, 5, 6, 7, 8, 7, 9, 10, 9, 11, 12,
              13, 14, 15, 12, 15, 14, 13, 16, 17, 16, 18, 19, 20, 21,
              22, 21, 20, 19, 22, 23, 24, 25, 26, 23, 26, 25, 24, 27,
              28, 29, 30, 27, 30, 29, 28, 31, 32, 33, 34, 31, 34, 33,
              32 ], table := CharacterTable( "4.L2(25).2_3" ) ) ] ]
gap> IsRecord( TransformingPermutationsCharacterTables( poss[3][1].table,
>                 CharacterTable( "4.L2(25).2_3" ) ) );
true
```

Note that the group $\Gamma L(2, 25)$ does *not* contain subgroups of the structure $4.L_2(25).2_3$, since $\Gamma L(2, 25)$ acts on its subgroup of scalar matrices via mapping each element to its fifth power, thus the central subgroup of order four in $\mathrm{GL}(2, 25)$ is central also in $\Gamma L(2, 25)$.

```
——————————————— Example ———————————————
gap> g:= GammaL(2,25);;
gap> phi:= IsomorphismPermGroup( g );;
gap> img:= Image( phi );;
gap> der:= DerivedSubgroup( img );;
gap> derder:= DerivedSubgroup( der );;
gap> Index( img, derder );
48
gap> inter:= Filtered( IntermediateSubgroups( img, derder ).subgroups,
>                 s -> Size( s ) = 4 * Size( derder ) and
>                      IsCyclic( CommutatorFactorGroup( s ) ) and
>                      Size( Centre( s ) ) = 2 );
[  ]
```

In order to construct a representation of a group of the structure $4.L_2(25).2_3$, we can use the function `CyclicExtensions` from the GAP package GrpConst. We start from the index two subgroup $4.L_2(25)$, which is a central product of $SL(2,25)$ and a cyclic group of order four, and find exactly one upwards extension by a cyclic group of order two, up to isomorphism, with the required properties.

```
——————————————— Example ———————————————
gap> c:= Centralizer( img, derder );;
gap> Size( c );  IsCyclic( c );
24
true
gap> cgen:= MinimalGeneratingSet( c );;
gap> four:= cgen[1]^6;;
gap> s:= ClosureGroup( derder, four );;
gap> LoadPackage( "GrpConst", false );
true
gap> filt:= Filtered( CyclicExtensions( s, 2 ),
>               x -> Size( Centre( x ) ) = 2 and
>                    IsCyclic( CommutatorFactorGroup( x ) ) );;
gap> Length( filt );
2
gap> IsomorphismGroups( filt[1], filt[2] ) <> fail;
true
```

The character table of this group coincides with the library table.

```
——————————————— Example ———————————————
gap> TransformingPermutationsCharacterTables( CharacterTable( filt[1] ),
>        CharacterTable( "4.L2(25).2_3" ) ) <> fail;
true
```

2.4.14   The Character Table of $4.L_2(49).2_3$ (December 2020)

The character tables of the simple group $L_2(49)$ and of its extensions do not appear in the Atlas of Finite Groups [CCN+85], but they may be regarded as Atlas tables because a data file in the format used to produce the Atlas has been available for a long time, as is stated in [JLPW95, Appendix 2].

Analogous to $L_2(9) \cong A_6$ and $L_2(25)$, see Section 2.4.13, the Atlas map for $G = L_2(49)$ shows a "broken box", since there is no group of the form $2.G.2_3$, and a group of the structure $4.G.2_3$ can

be considered instead, which has a normal subgroup isomorphic with $2.(2 \times G)$ and a factor group isomorphic with $(2 \times G).2_3$, see Section 2.4.13. Having its character table available has the effect that the functions `DisplayAtlasMap` (**CTblLib: DisplayAtlasMap for the name of a simple group**) and `BrowseAtlasTable` (**CTblLib: BrowseAtlasTable**) work with input `"L2(49)"`.

We construct the character table of $4.L_2(49).2_3$ in the same way as for the extensions of $L_2(9)$ and $L_2(25)$. There is a unique solution.

```
                           ──── Example ────
  gap> c2:= CharacterTable( "Cyclic", 2 );;
  gap> 2l:= CharacterTable( "2.L2(49)" );;
  gap> tblC:= CharacterTableIsoclinic( 2l * c2 );;
  gap> ord2:= Filtered( ClassPositionsOfNormalSubgroups( tblC ),
  >                x -> Length( x ) = 2 );
  [ [ 1, 3 ] ]
  gap> tblG:= tblC / ord2[1];;
  gap> tblNbar:= CharacterTableIsoclinic(
  >                CharacterTable( "L2(49).2_3" ) * c2 );;
  gap> fus:= PossibleClassFusions( tblG, tblNbar );;
  gap> Length( fus );
  10
  gap> StoreFusion( tblG, fus[1], tblNbar );
  gap> elms:= PossibleActionsForTypeMGA( tblC, tblG, tblNbar );;
  gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
  >                tblC, tblG, tblNbar, pi, "4.L2(49).2_3" ) );;
  gap> List( poss, Length );
  [ 0, 0, 1 ]
  gap> t:= poss[3][1].table;
  CharacterTable( "4.L2(49).2_3" )
```

Analogous to the situation with $L_2(9)$, a group of the desired structure can be found inside the semilinear group $\Gamma L(2, 49)$. In fact, there is a unique class of subgroups in $\Gamma L(2, 49)$ that contain $SL(2, 49) \cong 2.G$, have the right order, have cyclic commutator factor group, and centre of order 2.

```
                           ──── Example ────
  gap> g:= GammaL(2,49);;
  gap> phi:= IsomorphismPermGroup( g );;
  gap> img:= Image( phi );;
  gap> der:= DerivedSubgroup( img );;
  gap> derder:= DerivedSubgroup( der );;
  gap> Index( img, derder );
  96
  gap> inter:= Filtered( IntermediateSubgroups( img, derder ).subgroups,
  >                s -> Size( s ) = 4 * Size( derder ) and
  >                     IsCyclic( CommutatorFactorGroup( s ) ) and
  >                     Size( Centre( s ) ) = 2 );;
  gap> Length( inter );
  4
  gap> ForAll( inter, x -> IsConjugate( img, inter[1], x ) );
  true
```

The character tables of these groups coincide with the table constructed above, and with the library table.

```
┌─────────────────────────── Example ───────────────────────────┐
  gap> TransformingPermutationsCharacterTables( t,
  >         CharacterTable( inter[1] ) ) ) <> fail;
  true
  gap> TransformingPermutationsCharacterTables( t,
  >         CharacterTable( "4.L2(49).2_3" ) ) ) <> fail;
  true
└────────────────────────────────────────────────────────────────┘
```

2.4.15   The Character Table of $4.L_2(81).2_3$ (December 2020)

We start with the character-theoretic construction of this table, analogous to the cases of $L_2(9)$, $L_2(25)$, $L_2(49)$.

```
┌─────────────────────────── Example ───────────────────────────┐
  gap> c2:= CharacterTable( "Cyclic", 2 );;
  gap> 2l:= CharacterTable( "2.L2(81)" );;
  gap> tblC:= CharacterTableIsoclinic( 2l * c2 );;
  gap> ord2:= Filtered( ClassPositionsOfNormalSubgroups( tblC ),
  >               x -> Length( x ) = 2 );
  [ [ 1, 3 ] ]
  gap> tblG:= tblC / ord2[1];;
  gap> tblNbar:= CharacterTableIsoclinic(
  >                 CharacterTable( "L2(81).2_3" ) * c2 );;
  gap> fus:= PossibleClassFusions( tblG, tblNbar );;
  gap> Length( fus );
  40
  gap> fusreps:= RepresentativesFusions( tblG, fus, tblNbar );;
  gap> Length( fusreps );
  1
  gap> StoreFusion( tblG, fusreps[1], tblNbar );
  gap> elms:= PossibleActionsForTypeMGA( tblC, tblG, tblNbar );;
  gap> poss:= List( elms, pi -> PossibleCharacterTablesOfTypeMGA(
  >               tblC, tblG, tblNbar, pi, "4.L2(81).2_3" ) );;
  gap> List( poss, Length );
  [ 0, 0, 1 ]
  gap> TransformingPermutationsCharacterTables( poss[3][1].table,
  >         CharacterTable( "4.L2(81).2_3" ) ) ) <> fail;
  true
└────────────────────────────────────────────────────────────────┘
```

Like in the case of $L_2(25)$, there are *no* $4.L_2(81).2_3$ type subgroups in $\Gamma L(2,81)$.

```
┌─────────────────────────── Example ───────────────────────────┐
  gap> g:= GammaL(2,81);;
  gap> phi:= IsomorphismPermGroup( g );;
  gap> img:= Image( phi );;
  gap> der:= DerivedSubgroup( img );;
  gap> derder:= DerivedSubgroup( der );;
  gap> Index( img, derder );
  320
  gap> inter:= Filtered( IntermediateSubgroups( img, derder ).subgroups,
  >               s -> Size( s ) = 4 * Size( derder ) and
  >                    IsCyclic( CommutatorFactorGroup( s ) ) and
  >                    Size( Centre( s ) ) = 2 );;
└────────────────────────────────────────────────────────────────┘
```

```
gap> ForAll( inter, x -> IsConjugate( img, inter[1], x ) );
true
gap> NrConjugacyClasses( inter[1] );
52
gap> NrConjugacyClasses( CharacterTable( "4.L2(81).2_3" ) );
112
```

The subgroups of $\Gamma L(2, 81)$ constructed above have the structure $2.L_2(81).4_1$.

```
───────────── Example ─────────────
gap> t:= CharacterTable( "2.L2(81).4_1" );;
gap> NrConjugacyClasses( t );
52
gap> TransformingPermutationsCharacterTables( t,
>        CharacterTable( inter[1] ) ) <> fail;
true
```

Like in the case of $L_2(25)$, we can construct a group with the structure $4.L_2(81).2_3$ via the function `CyclicExtensions` from the GAP package GrpConst.

```
───────────── Example ─────────────
gap> c:= Centralizer( img, derder );;
gap> Size( c );  IsCyclic( c );
80
true
gap> cgen:= MinimalGeneratingSet( c );;
gap> four:= cgen[1]^20;;
gap> s:= ClosureGroup( derder, four );;
gap> LoadPackage( "GrpConst", false );
true
gap> filt:= Filtered( CyclicExtensions( s, 2 ),
>               x -> Size( Centre( x ) ) = 2 and
>                    IsCyclic( CommutatorFactorGroup( x ) ) );;
gap> Length( filt );
2
gap> IsomorphismGroups( filt[1], filt[2] ) <> fail;
true
gap> TransformingPermutationsCharacterTables( CharacterTable( filt[1] ),
>        CharacterTable( "4.L2(81).2_3" ) ) <> fail;
true
```

2.4.16   The Character Table of $9.U_3(8).3_3$ (March 2017)

The group that is called $9.U_3(8).3_3$ in the Atlas of Finite Groups occurs as a subgroup of $\Gamma U(3, 8)$. Note that $GU(3, 8)$ has the structure $3.(3 \times U_3(8)).3_2$ (see [CCN+85, p. 66]), and extending the subgroup $C = 3.(3 \times U_3(8))$ by the product of an element outside C with the field automorphism of order three of $GF(64)$ yields a group N of the structure $3.(3 \times U_3(8)).3_3$ whose centre has order three.

The character table of N can be constructed with the $M.G.A$ construction method from Section 2.3.1. The situation is similar to that with $4.A_6.2_3$, see Section 2.4.13, in particular the situation is described by the same picture that is shown for $4.A_6.2_3$ in this section, just the subgroups Z and $\langle g \rangle$ have the orders three and nine, respectively, and C has index three in N.

The normal subgroup $C \cong 9.U_3(8)$ is a central product of $U = 3.U_3(8)$ and a cyclic group $\langle g \rangle$ of order 9, and the factor group of N by the central subgroup $Z = \langle g^3 \rangle$ of order 3 is isomorphic to a subdirect product $\overline{N}$ of $U_3(8).3_3$ and a cyclic group of order 9, such that N acts nontrivially on its normal subgroup $\langle g \rangle$.

Thus N has the structure $3.G.3$, with $3.G = C$ and $G.3 = \overline{N}$. Each element in $N \setminus C$ raises g to its fourth or seventh power, so it acts fixed point freely on the faithful irreducible characters of C. Hence we can use `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfType-MGA**) for constructing the character table of N from the tables of C and N/Z and the action of N on the classes of C.

Since we want to construct also *Brauer tables* of N, we have to choose the class fusion that describes the embedding of C/Z into $\overline{N}$ compatibly with the known Brauer tables of $U_3(8)$ and $U_3(8).3_3$. Note that the 2-modular tables of these groups impose additional conditions on the class fusion.

```
───────────── Example ─────────────
  gap> s:= CharacterTable( "U3(8)" );;
  gap> s3:= CharacterTable( "U3(8).3_3" );;
  gap> poss:= PossibleClassFusions( s, s3 );;
  gap> Length( poss );
  4
  gap> Length( RepresentativesFusions( s, poss, s3 ) );
  1
  gap> smod2:= s mod 2;;
  gap> s3mod2:= s3 mod 2;;
  gap> good:= [];;  modmap:= 0;;
  gap> for map in poss do
  >       modmap:= CompositionMaps( InverseMap( GetFusionMap( s3mod2, s3 ) ),
  >                    CompositionMaps( map, GetFusionMap( smod2, s ) ) );
  >       rest:= List( Irr( s3mod2 ), x -> x{ modmap } );
  >       if not fail in Decomposition( Irr( smod2 ), rest, "nonnegative" ) then
  >         Add( good, map );
  >       fi;
  >    od;
  gap> Length( good );
  2
```

The class fusion from $U_3(8)$ to $U_3(8).3_3$ is determined up to complex conjugation by the 2-modular Brauer tables. We choose the fusion that is stored on the library tables.

```
───────────── Example ─────────────
  gap> good[2] = CompositionMaps( PowerMap( s3, -1 ), good[1] );
  true
  gap> GetFusionMap( s, s3 ) in good;
  true
  gap> sfuss3:= GetFusionMap( s, s3 );;
```

In the next step, we construct the character tables of $C/Z \cong U_3(8) \times 3$ and $N/Z \cong (U_3(8) \times 3).3_3$, and those class fusions between the two tables that are compatible with the fusion between the factors that was chosen above (w. r. t. the stored factor fusions).

In order not to leave out some candidates, we have to consider also the table of N/Z that is obtained from the "other" construction as an isoclinic table of $3 \times U_3(8).3_3$.

(This may look complicated. It would perhaps be more natural to construct the ordinary tables first, by considering the possible fusions, and later to adjust the choices to the conditions that are imposed by the Brauer tables. However, the technical complications of that construction would not be smaller in the end.)

We get four candidates, two for each of the two tables of N/Z.

```
––––––––––––––––––––––– Example –––––––––––––––––––––––
gap> c3:= CharacterTable( "Cyclic", 3 );;
gap> tblG:= s * c3;;
gap> dp:= s3 * c3;;
gap> tblGA1:= CharacterTableIsoclinic( dp, rec( k:= 1 ) );;
gap> tblGA2:= CharacterTableIsoclinic( dp, rec( k:= 2 ) );;
gap> good:= [];;
gap> tblGmod2:= tblG mod 2;;
gap> for tblGA in [ tblGA1, tblGA2 ] do
>        tblGAmod2:= tblGA mod 2;
>        for map in PossibleClassFusions( tblG, tblGA ) do
>          modmap:= CompositionMaps(
>               InverseMap( GetFusionMap( tblGAmod2, tblGA ) ),
>               CompositionMaps( map, GetFusionMap( tblGmod2, tblG ) ) );
>          rest:= List( Irr( tblGAmod2 ), x -> x{ modmap } );
>          if not fail in Decomposition( Irr( tblGmod2 ), rest,
>                            "nonnegative" ) and
>            CompositionMaps( GetFusionMap( tblGA, s3 ), map ) =
>            CompositionMaps( sfuss3, GetFusionMap( tblG, s ) ) then
>            Add( good, [ tblGA, map ] );
>          fi;
>        od;
>     od;
gap> List( good, x -> x[1] );
[ CharacterTable( "Isoclinic(U3(8).3_3xC3,1)" ),
  CharacterTable( "Isoclinic(U3(8).3_3xC3,1)" ),
  CharacterTable( "Isoclinic(U3(8).3_3xC3,2)" ),
  CharacterTable( "Isoclinic(U3(8).3_3xC3,2)" ) ]
```

The character table of C can be constructed with `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**) from the character table of $3 \times 3.U_3(8)$. (Here we need to consider only one variant of the table.)

```
––––––––––––––––––––––– Example –––––––––––––––––––––––
gap> 3s:= CharacterTable( "3.U3(8)" );;
gap> dp:= 3s * c3;;
gap> tblMG:= CharacterTableIsoclinic( dp );;
```

The construction of this table does not automatically yield a factor fusion to the table of C/Z. We form the relevant factor table, which has the same ordering of irreducible characters, and use the factor fusion to this table.

```
––––––––––––––––––––––– Example –––––––––––––––––––––––
gap> GetFusionMap( tblMG, tblG );
fail
gap> cen:= ClassPositionsOfCentre( tblMG );
[ 1, 2, 3, 4, 5, 6, 7, 8, 9 ]
```

```
gap> OrdersClassRepresentatives( tblMG ){ cen };
[ 1, 9, 9, 3, 9, 9, 3, 9, 9 ]
gap> facttbl:= tblMG / [ 1, 4, 7 ];;
gap> tr:= TransformingPermutationsCharacterTables( facttbl, tblG );;
gap> tr.rows;  tr.columns;
()
()
gap> StoreFusion( tblMG, GetFusionMap( tblMG, facttbl ), tblG );
```

Now we compute the orbits of the possible actions of *N* on the classes of *C*, and the resulting candidates for the character table of *N*.

```
───────────── Example ─────────────
gap> posstbls:= [];;
gap> for pair in good do
>       tblGA:= pair[1];
>       GfusGA:= pair[2];
>       tblG:= s * c3;
>       StoreFusion( tblG, GfusGA, tblGA );
>       for pi in PossibleActionsForTypeMGA( tblMG, tblG, tblGA ) do
>         for cand in PossibleCharacterTablesOfTypeMGA(
>                         tblMG, tblG, tblGA, pi, "test" ) do
>           Add( posstbls, [ tblGA, cand ] );
>         od;
>       od;
>     od;
gap> Length( posstbls );
32
```

Now we discard all those candidates that are not compatible with the 2-modular character tables.

```
───────────── Example ─────────────
gap> compatible:= [];;  r:= 0;;  modr:= 0;;
gap> for pair in posstbls do
>       tblGA:= pair[1];
>       r:= pair[2];
>       comp:= ComputedClassFusions( tblMG );
>       pos:= PositionProperty( comp, x -> x.name = Identifier( r.table ) );
>       if pos = fail then
>         StoreFusion( tblMG, r.MGfusMGA, r.table );
>       else
>         comp[ pos ]:= ShallowCopy( comp[ pos ] );
>         comp[ pos ].map:= r.MGfusMGA;
>       fi;
>       Unbind( ComputedBrauerTables( tblMG )[2] );
>       modr:= BrauerTableOfTypeMGA( tblMG mod 2, tblGA mod 2, r.table );
>       rest:= List( Irr( modr.table ), x -> x{ modr.MGfusMGA } );
>       dec:= Decomposition( Irr( tblMG mod 2 ), rest, "nonnegative" );
>       if not fail in dec then
>         Add( compatible, pair );
>       fi;
>     od;
gap> Length( compatible );
8
```

The remaining candidates fall into two equivalence classes.

``` Example
gap> tbls:= [];;
gap> for pair in compatible do
>        if ForAll( tbls, t -> TransformingPermutationsCharacterTables(
>                                t, pair[2].table ) = fail ) then
>          Add( tbls, pair[2].table );
>        fi;
>      od;
gap> Length( tbls );
2
```

The two tables can be distinguished by their element orders one contains the element order 54 and the other does not or by their 4th power maps the classes of element order 171 in one table are not fixed by the 4th power map, the corresponding classes in the other table are fixed.

``` Example
gap> Set( OrdersClassRepresentatives( tbls[1] ) );
[ 1, 2, 3, 4, 6, 7, 9, 12, 18, 19, 21, 27, 36, 54, 57, 63, 171 ]
gap> Set( OrdersClassRepresentatives( tbls[2] ) );
[ 1, 2, 3, 4, 6, 7, 9, 12, 18, 19, 21, 27, 36, 57, 63, 171 ]
gap> pos171:= Positions( OrdersClassRepresentatives( tbls[1] ), 171 );;
gap> pow4:= PowerMap( tbls[1], 4 );;
gap> ForAny( [ 1 .. Length( pos171 ) ],
>            i -> pos171[i] = pow4[ pos171[i] ] );
false
gap> pos171:= Positions( OrdersClassRepresentatives( tbls[2] ), 171 );;
gap> PowerMap( tbls[2], 4 ){ pos171 } = pos171;
true
```

Thus we can use the group *N* to decide which table is correct. For that, we construct a permutation representation of *N*.

``` Example
gap> gu:= GU(3,8);;
gap> orbs:= OrbitsDomain( gu, Elements( GF(64)^3 ) );;
gap> List( orbs, Length );
[ 1, 32319, 32832, 32832, 32832, 32832, 32832, 32832 ]
gap> orb:= SortedList( First( orbs, x -> Length( x ) = 32319 ) );;
gap> actgu:= Action( gu, orb, OnRight );;
gap> Size( actgu ) = Size( gu );
true
gap> cen:= Centre( actgu );;
gap> Size( cen );
9
gap> u:= ClosureGroup( DerivedSubgroup( actgu ), cen );;
gap> aut:= v -> List( v, x -> x^4 );;
gap> pi:= PermList( List( orb, v -> PositionSorted( orb, aut( v ) ) ) );;
gap> outer:= First( GeneratorsOfGroup( actgu ), x -> not x in u );;
gap> g:= ClosureGroup( u, pi * outer );;
```

Before we perform computations with the group, we reduce the degree of the representation by a factor of 7.

```
——————————————— Example ———————————————
gap> g:= Group( SmallGeneratingSet( g ) );;
gap> allbl:= AllBlocks( g );;
gap> List( allbl, Length );
[ 3, 21, 63, 9, 7 ]
gap> orb:= Orbit( g, First( allbl, x -> Length( x ) = 7 ), OnSets );;
gap> act:= Action( g, orb, OnSets );;
gap> Size( act ) = Size( g );
true
gap> NrMovedPoints( act );
4617
```

Now we test whether an element of order 171 in *N* is conjugate in *N* to its fourth power.

```
——————————————— Example ———————————————
gap> repeat x:= PseudoRandom( act ); until Order( x ) = 171;
gap> IsConjugate( act, x, x^4 );
true
```

This means that the second of the candidate tables constructed above is the right one. The character table with the identifier `"9.U3(8).3_3"` in the character table library is equivalent to this table.

```
——————————————— Example ———————————————
gap> lib:= CharacterTable( "9.U3(8).3_3" );;
gap> IsRecord( TransformingPermutationsCharacterTables( tbls[2], lib ) );
true
```

GAP's currently available methods for the automatic computation of character tables would require too much space when called with this permutation group. Using interactive methods, one can compute the character table with GAP. The table obtained this way is equivalent to the library character table with the identifier `"9.U3(8).3_3"`.

I do not know how to disprove the other candidate with character-theoretic arguments. Thus this table provides an example of a pseudo character table, see Section 2.4.17.

2.4.17 Pseudo Character Tables of the Type *M*.*G*.*A* (May 2004)

With the construction method for character tables of groups of the type *M*.*G*.*A*, one can construct tables that have many properties of character tables but that are not character tables of groups, cf. [Gag86]. For example, the group $3.A_6.2_3$ has a *central* subgroup of order 3, so it is not of the type *M*.*G*.*A* with fixed-point free action on the faithful characters of *M*.*G*.

However, if we apply the "*M*.*G*.*A* construction" to the groups $M.G = 3.A_6$, $G = A_6$, and $G.A = A_6.2_3$ then we get a (in this case unique) result.

```
——————————————— Example ———————————————
gap> tblMG := CharacterTable( "3.A6" );;
gap> tblG  := CharacterTable( "A6" );;
gap> tblGA := CharacterTable( "A6.2_3" );;
gap> elms:= PossibleActionsForTypeMGA( tblMG, tblG, tblGA );
[ [ [ 1 ], [ 2, 3 ], [ 4 ], [ 5, 6 ], [ 7, 8 ], [ 9 ], [ 10, 11 ],
      [ 12, 15 ], [ 13, 17 ], [ 14, 16 ] ] ]
gap> poss:= PossibleCharacterTablesOfTypeMGA(
>                 tblMG, tblG, tblGA, elms[1], "pseudo" );
```

```
[ rec(
      MGfusMGA := [ 1, 2, 2, 3, 4, 4, 5, 5, 6, 7, 7, 8, 9, 10, 8, 10,
         9 ], table := CharacterTable( "pseudo" ) ) ]
```

Such a table automatically satisfies the orthogonality relations, and the tensor product of two "irreducible characters" of which at least one is a row from *G.A* decomposes into a sum of the "irreducible characters", where the coefficients are nonnegative integers.

In this example, any tensor product decomposes with nonnegative integral coefficients, n-th symmetrizations of "irreducible characters" decompose, for $n \leq 5$, and the "class multiplication coefficients" are nonnegative integers.

```
────────────── Example ──────────────
gap> pseudo:= poss[1].table;
CharacterTable( "pseudo" )
gap> Display( pseudo );
pseudo

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

        1a  3a 2a 6a 3b 4a 12a 5a 15a 15b 4b 8a 8b
     2P 1a  3a 1a 3a 3b 2a  6a 5a 15a 15b 2a 4a 4a
     3P 1a  1a 2a 2a 1a 4a  4a 5a  5a  5a 4b 8a 8b
     5P 1a  3a 2a 6a 3b 4a 12a 1a  3a  3a 4b 8b 8a

X.1      1   1  1  1  1  1   1  1   1   1  1  1  1
X.2      1   1  1  1  1  1   1  1   1   1 -1 -1 -1
X.3     10  10  2  2  1 -2  -2  .   .   .  .  .  .
X.4     16  16  .  . -2  .   .  1   1   1  .  .  .
X.5      9   9  1  1  .  1   1 -1  -1  -1  1 -1 -1
X.6      9   9  1  1  .  1   1 -1  -1  -1 -1  1  1
X.7     10  10 -2 -2  1  .   .  .   .   .  .  B -B
X.8     10  10 -2 -2  1  .   .  .   .   .  . -B  B
X.9      6  -3 -2  1  .  2  -1  1   A  /A  .  .  .
X.10     6  -3 -2  1  .  2  -1  1  /A   A  .  .  .
X.11    12  -6  4 -2  .  .   .  2  -1  -1  .  .  .
X.12    18  -9  2 -1  .  2  -1 -2   1   1  .  .  .
X.13    30 -15 -2  1  . -2   1  .   .   .  .  .  .

A = -E(15)-E(15)^2-E(15)^4-E(15)^8
  = (-1-Sqrt(-15))/2 = -1-b15
B = E(8)+E(8)^3
  = Sqrt(-2) = i2
gap> IsInternallyConsistent( pseudo );
true
gap> irr:= Irr( pseudo );;
gap> test:= Concatenation( List( [ 2 .. 5 ],
>                n -> Symmetrizations( pseudo, irr, n ) ) );;
gap> Append( test, Set( Tensored( irr, irr ) ) );
gap> fail in Decomposition( irr, test, "nonnegative" );
false
```

```
gap> if ForAny( Tuples( [ 1 .. NrConjugacyClasses( pseudo ) ], 3 ),
>       t -> not ClassMultiplicationCoefficient( pseudo, t[1], t[2], t[3] )
>               in NonnegativeIntegers ) then
>     Error( "contradiction" );
> fi;
```

I do not know a character-theoretic argument for showing that this table is *not* the character table of a group, but we can use the following group-theoretic argument. Suppose that the group G, say, has the above character table. Then G has a unique composition series with factors of the orders 3, 360, and 2, respectively. Let N denote the normal subgroup of order 3 in G. The factor group $F = G/N$ is an automorphic extension of A_6, and according to [CCN$^+$85, p. 4] it is isomorphic with $M_{10} = A_6.2_3$ and has Sylow 3 normalizers of the structure $3^2 : Q_8$. Since the Sylow 3 subgroup of G is a self-centralizing nonabelian group of order 3^3 and of exponent 3, the Sylow 3 normalizers in G have the structure $3_+^{1+2} : Q_8$, but the Q_8 type subgroups of $\mathrm{Aut}(3_+^{1+2})$ act trivially on the centre of 3_+^{1+2}, contrary to the situation in the above table.

In general, this construction need not produce tables for which all symmetrizations of irreducible characters decompose properly. For example, applying `PossibleCharacterTablesOfTypeMGA` (**CTblLib: PossibleCharacterTablesOfTypeMGA**) to the case $M.G = 3.L_3(4)$ and $G.A = L_3(4).2_1$ does not yield a table because the function suppresses tables that do not admit p-th power maps, for prime divisors p of the order of $M.G.A$, and in this case no compatible 2-power map exists.

```
——————————————— Example ———————————————
gap> tblMG := CharacterTable( "3.L3(4)" );;
gap> tblG  := CharacterTable( "L3(4)" );;
gap> tblGA := CharacterTable( "L3(4).2_1" );;
gap> elms:= PossibleActionsForTypeMGA( tblMG, tblG, tblGA );
[ [ [ 1 ], [ 2, 3 ], [ 4 ], [ 5, 6 ], [ 7 ], [ 8 ], [ 9, 10 ],
      [ 11 ], [ 12, 13 ], [ 14 ], [ 15, 16 ], [ 17, 20 ], [ 18, 22 ],
      [ 19, 21 ], [ 23, 26 ], [ 24, 28 ], [ 25, 27 ] ] ]
gap> PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA, elms[1], "?" );
[ ]
```

Also, it may happen that already `PossibleActionsForTypeMGA` (**CTblLib: PossibleActions-ForTypeMGA**) returns an empty list. Examples are $M.G = 3_1.U_4(3)$, $G.A = U_4(3).2_2$ and $M.G = 3_2.U_4(3)$, $G.A = U_4(3).2_3$.

```
——————————————— Example ———————————————
gap> tblG  := CharacterTable( "U4(3)" );;
gap> tblMG := CharacterTable( "3_1.U4(3)" );;
gap> tblGA := CharacterTable( "U4(3).2_2" );;
gap> PossibleActionsForTypeMGA( tblMG, tblG, tblGA );
[ ]
gap> tblMG:= CharacterTable( "3_2.U4(3)" );;
gap> tblGA:= CharacterTable( "U4(3).2_3" );;
gap> PossibleActionsForTypeMGA( tblMG, tblG, tblGA );
[ ]
```

Also the sections 2.4.5 and 2.4.16 provide examples of pseudo character tables. If one does not use the arguments about Brauer tables then the latter section presents in fact several pseudo character tables.

2.4.18   Some Extra-ordinary p-Modular Tables of the Type $M.G.A$ (September 2005)

For a group $M.G.A$ in the sense of Section 2.3.1 such that *not* all ordinary irreducible characters χ have the property that M is contained in the kernel of χ or χ is induced from $M.G$, it may happen that there are primes p such that all irreducible p-modular characters have this property. This happens if and only if the preimages in $M.G.A$ of each p-regular conjugacy class in $G.A \setminus G$ form one conjugacy class.

The following function can be used to decide whether this situation applies to a character table in the GAP Character Table Library; here we assume that for the library table of a group with the structure $M.G.A$, the class fusions from $M.G$ and to $G.A$ are stored.

```
                                ─── Example ───
 gap> FindExtraordinaryCase:= function( tblMGA )
 >     local result, der, nsg, tblMGAclasses, orders, tblMG,
 >           tblMGfustblMGA, tblMGclasses, pos, M, Mimg, tblMGAfustblGA, tblGA,
 >           outer, inv, filt, other, primes, p;
 >     result:= [];
 >     der:= ClassPositionsOfDerivedSubgroup( tblMGA );
 >     nsg:= ClassPositionsOfNormalSubgroups( tblMGA );
 >     tblMGAclasses:= SizesConjugacyClasses( tblMGA );
 >     orders:= OrdersClassRepresentatives( tblMGA );
 >     if Length( der ) < NrConjugacyClasses( tblMGA ) then
 >       # Look for tables of normal subgroups of the form $M.G$.
 >       for tblMG in Filtered( List( NamesOfFusionSources( tblMGA ),
 >                                    CharacterTable ), x -> x <> fail ) do
 >         tblMGfustblMGA:= GetFusionMap( tblMG, tblMGA );
 >         tblMGclasses:= SizesConjugacyClasses( tblMG );
 >         pos:= Position( nsg, Set( tblMGfustblMGA ) );
 >         if pos <> fail and
 >            Size( tblMG ) = Sum( tblMGAclasses{ nsg[ pos ] } ) then
 >           # Look for normal subgroups of the form $M$.
 >           for M in Difference( ClassPositionsOfNormalSubgroups( tblMG ),
 >                       [ [ 1 ], [ 1 .. NrConjugacyClasses( tblMG ) ] ] ) do
 >             Mimg:= Set( tblMGfustblMGA{ M } );
 >             if Sum( tblMGAclasses{ Mimg } ) = Sum( tblMGclasses{ M } ) then
 >               tblMGAfustblGA:= First( ComputedClassFusions( tblMGA ),
 >                   r -> ClassPositionsOfKernel( r.map ) = Mimg );
 >               if tblMGAfustblGA <> fail then
 >                 tblGA:= CharacterTable( tblMGAfustblGA.name );
 >                 tblMGAfustblGA:= tblMGAfustblGA.map;
 >                 outer:= Difference( [ 1 .. NrConjugacyClasses( tblGA ) ],
 >                     CompositionMaps( tblMGAfustblGA, tblMGfustblMGA ) );
 >                 inv:= InverseMap( tblMGAfustblGA ){ outer };
 >                 filt:= Flat( Filtered( inv, IsList ) );
 >                 if not IsEmpty( filt ) then
 >                   other:= Filtered( inv, IsInt );
 >                   primes:= Filtered( PrimeDivisors( Size( tblMGA ) ),
 >                       p -> ForAll( orders{ filt }, x -> x mod p = 0 )
 >                            and ForAny( orders{ other }, x -> x mod p <> 0 ) );
 >                   for p in primes do
 >                     Add( result, [ Identifier( tblMG ),
 >                                    Identifier( tblMGA ),
 >                                    Identifier( tblGA ), p ] );
```

```
>                   od;
>                 fi;
>               fi;
>             fi;
>           od;
>         fi;
>       od;
>     fi;
>     return result;
> end;;
```

Let us list the tables which are found by this function.

```
                              Example
gap> cases:= [];;
gap> for name in AllCharacterTableNames( IsDuplicateTable, false ) do
>        Append( cases, FindExtraordinaryCase( CharacterTable( name ) ) );
>     od;
gap> for i in Set( cases ) do
>        Print( i, "\n" );
>     od;
[ "2.A6", "2.A6.2_1", "A6.2_1", 3 ]
[ "2.Fi22", "2.Fi22.2", "Fi22.2", 3 ]
[ "2.L2(25)", "2.L2(25).2_2", "L2(25).2_2", 5 ]
[ "2.L2(49)", "2.L2(49).2_2", "L2(49).2_2", 7 ]
[ "2.L2(81)", "2.L2(81).2_1", "L2(81).2_1", 3 ]
[ "2.L2(81)", "2.L2(81).4_1", "L2(81).4_1", 3 ]
[ "2.L2(81).2_1", "2.L2(81).4_1", "L2(81).4_1", 3 ]
[ "2.L4(3)", "2.L4(3).2_2", "L4(3).2_2", 3 ]
[ "2.L4(3)", "2.L4(3).2_3", "L4(3).2_3", 3 ]
[ "2.S3", "2.D12", "S3x2", 3 ]
[ "2.U4(3).2_1", "2.U4(3).(2^2)_{12*2*}", "U4(3).(2^2)_{122}", 3 ]
[ "2.U4(3).2_1", "2.U4(3).(2^2)_{122}", "U4(3).(2^2)_{122}", 3 ]
[ "2.U4(3).2_1", "2.U4(3).(2^2)_{13*3*}", "U4(3).(2^2)_{133}", 3 ]
[ "2.U4(3).2_1", "2.U4(3).(2^2)_{133}", "U4(3).(2^2)_{133}", 3 ]
[ "3.U3(8)", "3.U3(8).3_1", "U3(8).3_1", 2 ]
[ "3.U3(8)", "3.U3(8).6", "U3(8).6", 2 ]
[ "3.U3(8)", "3.U3(8).6", "U3(8).6", 3 ]
[ "3.U3(8).2", "3.U3(8).6", "U3(8).6", 2 ]
[ "3^2:8", "2.A8N3", "s3wrs2", 3 ]
[ "5^(1+2):8:4", "2.HS.2N5", "HS.2N5", 5 ]
[ "6.A6", "6.A6.2_1", "3.A6.2_1", 3 ]
[ "6.A6", "6.A6.2_1", "A6.2_1", 3 ]
[ "6.Fi22", "6.Fi22.2", "3.Fi22.2", 3 ]
[ "6.Fi22", "6.Fi22.2", "Fi22.2", 3 ]
[ "Isoclinic(2.U4(3).2_1)", "2.U4(3).(2^2)_{1*2*2}",
  "U4(3).(2^2)_{122}", 3 ]
[ "Isoclinic(2.U4(3).2_1)", "2.U4(3).(2^2)_{1*3*3}",
  "U4(3).(2^2)_{133}", 3 ]
[ "bd10", "2.D20", "D20", 5 ]
```

The smallest example in this list is $2.A_6.2_1$, the double cover of the symmetric group on six points. The 3-modular table of this group looks as follows.

```
                            ___ Example ___
gap> Display( CharacterTable( "2.A6.2_1" ) mod 3 );
2.A6.2_1mod3


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


        1a  2a 4a 8a 5a 10a 2b 4b 8b
     2P 1a  1a 2a 4a 5a  5a 1a 2a 4a
     3P 1a  2a 4a 8a 5a 10a 2b 4b 8b
     5P 1a  2a 4a 8a 1a  2a 2b 4b 8b


X.1     1   1  1  1  1   1  1  1  1
X.2     1   1  1  1  1   1 -1 -1 -1
X.3     6   6 -2  2  1   1  .  .  .
X.4     4   4  . -2 -1  -1  2 -2  .
X.5     4   4  . -2 -1  -1 -2  2  .
X.6     9   9  1  1 -1  -1  3  3 -1
X.7     9   9  1  1 -1  -1 -3 -3  1
X.8     4  -4  .  . -1   1  .  .  .
X.9    12 -12  .  .  2  -2  .  .  .
```

We see that the two faithful irreducible characters vanish on the three classes outside $2.A_6$.

For the groups in the above list, the function BrauerTableOfTypeMGA (**CTblLib: BrauerTable-OfTypeMGA**) can be used to construct the p-modular tables of $M.G.A$ from the tables of $M.G$ and $G.A$, for the given special primes p. The computations can be performed as follows.

```
                            ___ Example ___
gap> for input in cases do
>        p:= input[4];
>        modtblMG:=  CharacterTable( input[1] ) mod p;
>        ordtblMGA:= CharacterTable( input[2] );
>        modtblGA:=  CharacterTable( input[3] ) mod p;
>        name:= Concatenation( Identifier( ordtblMGA ), " mod ", String(p) );
>        if ForAll( [ modtblMG, modtblGA ], IsCharacterTable ) then
>          poss:= BrauerTableOfTypeMGA( modtblMG, modtblGA, ordtblMGA );
>          modlib:= ordtblMGA mod p;
>          if IsCharacterTable( modlib ) then
>            trans:= TransformingPermutationsCharacterTables( poss.table,
>                         modlib );
>            if not IsRecord( trans ) then
>              Print( "#E  computed table and library table for ", name,
>                     " differ\n" );
>            fi;
>          else
>            Print( "#I  no library table for ", name, "\n" );
>          fi;
>        else
>          Print( "#I  not all input tables for ", name, " available\n" );
>        fi;
>    od;
#I  not all input tables for 2.L2(49).2_2 mod 7 available
```

```
#I  not all input tables for 2.L2(81).2_1 mod 3 available
#I  not all input tables for 2.L2(81).4_1 mod 3 available
#I  not all input tables for 2.L2(81).4_1 mod 3 available
```

The examples $2.A_6.2_1$, $2.L_2(25).2_2$, and $2.L_2(49).2_2$ belong to the infinite series of semiliniear groups $\Sigma L(2, p^2)$, for odd primes p. All groups in this series have the property that all faithful irreducible characters vanish on the p-regular classes outside $SL(2, p^2)$. (Cf. Section 2.2.6 for another property of the groups in this series.)

2.5 Examples for the Type $G.S_3$

2.5.1 Small Examples

The symmetric group S_4 on four points has the form $G.S_3$ where G is the Klein four group V_4, $G.2$ is the dihedral group D_8 of order 8, and $G.3$ is the alternating group A_4. The trivial character of A_4 extends twofold to S_4, in the same way as the trivial character of V_4 extends to the dihedral group. The nontrivial linear characters of A_4 induce irreducibly to S_4. The irreducible degree three character of A_4 is induced from any of the three nontrivial linear characters of V_4, it extends to S_4 in the same way as the unique constituent of the restriction to V_4 that is invariant in the chosen D_8 extends to D_8.

```
─────────────────────────────── Example ───────────────────────────────
gap> c2:= CharacterTable( "Cyclic", 2 );;
gap> t:= c2 * c2;;
gap> tC:= CharacterTable( "Dihedral", 8 );;
gap> tK:= CharacterTable( "Alternating", 4 );;
gap> tfustC:= PossibleClassFusions( t, tC );
[ [ 1, 3, 4, 4 ], [ 1, 3, 5, 5 ], [ 1, 4, 3, 4 ], [ 1, 4, 4, 3 ],
  [ 1, 5, 3, 5 ], [ 1, 5, 5, 3 ] ]
gap> StoreFusion( t, tfustC[1], tC );
gap> tfustK:= PossibleClassFusions( t, tK );
[ [ 1, 2, 2, 2 ] ]
gap> StoreFusion( t, tfustK[1], tK );
gap> elms:= PossibleActionsForTypeGS3( t, tC, tK );
[ (3,4) ]
gap> new:= CharacterTableOfTypeGS3( t, tC, tK, elms[1], "S4" );
rec( table := CharacterTable( "S4" ),
  tblCfustblKC := [ 1, 4, 2, 2, 5 ], tblKfustblKC := [ 1, 2, 3, 3 ] )
gap> Display( new.table );
S4

     2  3  3  .  2  2
     3  1  .  1  .  .

        1a 2a 3a 4a 2b
     2P 1a 1a 3a 2a 1a
     3P 1a 2a 1a 4a 2b

X.1     1  1  1  1  1
X.2     1  1  1 -1 -1
X.3     3 -1  .  1 -1
```

```
  X.4      3 -1  . -1  1
  X.5      2  2 -1  .  .
```

The case $e > 1$ occurs in the following example. We choose G the cyclic group of order two, $G.C$ the cyclic group of order six, $G.K$ the quaternion group of order eight, and construct the character table of $G.F = SL_2(3)$, with $F \cong A_4$.

We get three extensions of the trivial character of $G.K$ to $G.F$, a degree three character induced from the nontrivial linear characters of $G.K$, and three extensions of the irreducible degree 2 character of $G.K$.

```
                              Example
 gap> t:= CharacterTable( "Cyclic", 2 );;
 gap> tC:= CharacterTable( "Cyclic", 6 );;
 gap> tK:= CharacterTable( "Quaternionic", 8 );;
 gap> tfustC:= PossibleClassFusions( t, tC );
 [ [ 1, 4 ] ]
 gap> StoreFusion( t, tfustC[1], tC );
 gap> tfustK:= PossibleClassFusions( t, tK );
 [ [ 1, 3 ] ]
 gap> StoreFusion( t, tfustK[1], tK );
 gap> elms:= PossibleActionsForTypeGS3( t, tC, tK );
 [ (2,5,4) ]
 gap> new:= CharacterTableOfTypeGS3( t, tC, tK, elms[1], "SL(2,3)" );
 rec( table := CharacterTable( "SL(2,3)" ),
   tblCfustblKC := [ 1, 4, 5, 3, 6, 7 ],
   tblKfustblKC := [ 1, 2, 3, 2, 2 ] )
 gap> Display( new.table );
 SL(2,3)

     2  3  2  3  1   1   1  1
     3  1  .  1  1   1   1  1

        1a 4a 2a 6a  3a  3b 6b
     2P 1a 2a 1a 3a  3b  3a 3b
     3P 1a 4a 2a 2a  1a  1a 2a

 X.1     1  1  1  1   1   1  1
 X.2     1  1  1  A  /A   A /A
 X.3     1  1  1 /A   A  /A  A
 X.4     3 -1  3  .   .   .  .
 X.5     2  . -2 /A  -A -/A  A
 X.6     2  . -2  1  -1  -1  1
 X.7     2  . -2  A -/A  -A /A

 A = E(3)
   = (-1+Sqrt(-3))/2 = b3
```

2.5.2 Atlas Tables of the Type $G.S_3$

We demonstrate the construction of all those ordinary and modular character tables in the GAP Character Table Library that are of the type $G.S_3$ where G is a simple group or a central extension of a simple group whose character table is contained in the Atlas. Here is the list of Identifier (**Reference:**

Identifier for tables of marks) values needed for accessing the input tables and the known library tables corresponding to the output.

```
───────────────────────── Example ─────────────────────────
gap> listGS3:= [
> [ "U3(5)",        "U3(5).2",        "U3(5).3",        "U3(5).S3"        ],
> [ "3.U3(5)",      "3.U3(5).2",      "3.U3(5).3",      "3.U3(5).S3"      ],
> [ "L3(4)",        "L3(4).2_2",      "L3(4).3",        "L3(4).3.2_2"     ],
> [ "L3(4)",        "L3(4).2_3",      "L3(4).3",        "L3(4).3.2_3"     ],
> [ "3.L3(4)",      "3.L3(4).2_2",    "3.L3(4).3",      "3.L3(4).3.2_2"   ],
> [ "3.L3(4)",      "3.L3(4).2_3",    "3.L3(4).3",      "3.L3(4).3.2_3"   ],
> [ "2^2.L3(4)",    "2^2.L3(4).2_2",  "2^2.L3(4).3",    "2^2.L3(4).3.2_2" ],
> [ "2^2.L3(4)",    "2^2.L3(4).2_3",  "2^2.L3(4).3",    "2^2.L3(4).3.2_3" ],
> [ "U6(2)",        "U6(2).2",        "U6(2).3",        "U6(2).3.2"       ],
> [ "3.U6(2)",      "3.U6(2).2",      "3.U6(2).3",      "3.U6(2).3.2"     ],
> [ "2^2.U6(2)",    "2^2.U6(2).2",    "2^2.U6(2).3",    "2^2.U6(2).3.2"   ],
> [ "O8+(2)",       "O8+(2).2",       "O8+(2).3",       "O8+(2).3.2"      ],
> [ "2^2.O8+(2)",   "2^2.O8+(2).2",   "2^2.O8+(2).3",   "2^2.O8+(2).3.2"  ],
> [ "L3(7)",        "L3(7).2",        "L3(7).3",        "L3(7).S3"        ],
> [ "3.L3(7)",      "3.L3(7).2",      "3.L3(7).3",      "3.L3(7).S3"      ],
> [ "U3(8)",        "U3(8).2",        "U3(8).3_2",      "U3(8).S3"        ],
> [ "3.U3(8)",      "3.U3(8).2",      "3.U3(8).3_2",    "3.U3(8).S3"      ],
> [ "U3(11)",       "U3(11).2",       "U3(11).3",       "U3(11).S3"       ],
> [ "3.U3(11)",     "3.U3(11).2",     "3.U3(11).3",     "3.U3(11).S3"     ],
> [ "O8+(3)",       "O8+(3).2_2",     "O8+(3).3",       "O8+(3).S3"       ],
> [ "2E6(2)",       "2E6(2).2",       "2E6(2).3",       "2E6(2).S3"       ],
> [ "2^2.2E6(2)",   "2^2.2E6(2).2",   "2^2.2E6(2).3",   "2^2.2E6(2).S3"   ],
> ];;
```

(For G one of $L_3(4)$, $U_6(2)$, $O_8^+(2)$, and $^2E_6(2)$, the tables of $2^2.G$, $2^2.G.2$, and $2^2.G.3$ can be constructed with the methods described in Section 2.3.4 and Section 2.3.1, respectively.)

Analogously, the automorphism groups of $L_3(4)$, $U_3(8)$, and $O_8^+(3)$ have factor groups isomorphic with S_3; in these cases, we choose $G = L_3(4).2_1$, $G = U_3(8).3_1$, and $G = O_8^+(3).2_{111}^2$, respectively.

```
───────────────────────── Example ─────────────────────────
gap> Append( listGS3, [
> [ "L3(4).2_1",          "L3(4).2^2",     "L3(4).6",      "L3(4).D12"      ],
> [ "2^2.L3(4).2_1",      "2^2.L3(4).2^2", "2^2.L3(4).6",  "2^2.L3(4).D12"  ],
> [ "U3(8).3_1",          "U3(8).6",       "U3(8).3^2",    "U3(8).(S3x3)"   ],
> [ "O8+(3).(2^2)_{111}", "O8+(3).D8",     "O8+(3).A4",    "O8+(3).S4"      ],
> ] );
```

In all these cases, the required table automorphism of $G.3$ is uniquely determined. We first compute the ordinary character table of $G.S_3$ and then the p-modular tables, for all prime divisors p of the order of G such that the GAP Character Table Library contains the necessary p-modular input tables.

In each case, we compare the computed character tables with the ones in the GAP Character Table Library. Note that in order to avoid conflicts of the class fusions that arise in the construction with the class fusions that are already stored on the library tables, we choose identifiers for the result tables that are different from the identifiers of the library tables.

```
───────────────────────── Example ─────────────────────────
gap> ProcessGS3Example:= function( t, tC, tK, identifier, pi )
>     local tF, lib, trans, p, tmodp, tCmodp, tKmodp, modtF;
```

```
>
>     tF:= CharacterTableOfTypeGS3( t, tC, tK, pi,
>              Concatenation( identifier, "new" ) );
>     lib:= CharacterTable( identifier );
>     if lib <> fail then
>       trans:= TransformingPermutationsCharacterTables( tF.table, lib );
>       if not IsRecord( trans ) then
>         Print( "#E  computed table and library table for '", identifier,
>                "' differ\n" );
>       fi;
>     else
>       Print( "#I  no library table for '", identifier, "'\n" );
>     fi;
>     StoreFusion( tC, tF.tblCfustblKC, tF.table );
>     StoreFusion( tK, tF.tblKfustblKC, tF.table );
>     for p in PrimeDivisors( Size( t ) ) do
>       tmodp := t  mod p;
>       tCmodp:= tC mod p;
>       tKmodp:= tK mod p;
>       if IsCharacterTable( tmodp ) and
>          IsCharacterTable( tCmodp ) and
>          IsCharacterTable( tKmodp ) then
>         modtF:= CharacterTableOfTypeGS3( tmodp, tCmodp, tKmodp,
>                   tF.table,
>                   Concatenation(  identifier, "mod", String( p ) ) );
>         if   Length( Irr( modtF.table ) ) <>
>              Length( Irr( modtF.table )[1] ) then
>           Print( "#E  nonsquare result table for '",
>                  identifier, " mod ", p, "'\n" );
>         elif lib <> fail and IsCharacterTable( lib mod p ) then
>           trans:= TransformingPermutationsCharacterTables( modtF.table,
>                                                    lib mod p );
>           if not IsRecord( trans ) then
>             Print( "#E  computed table and library table for '",
>                    identifier, " mod ", p, "' differ\n" );
>           fi;
>         else
>           Print( "#I  no library table for '", identifier, " mod ",
>                  p, "'\n" );
>         fi;
>       else
>         Print( "#I  not all inputs available for '", identifier,
>                " mod ", p, "'\n" );
>       fi;
>     od;
>   end;;
```

Now we call the function for the examples in the list.

```
────────────────────── Example ──────────────────────
gap> for input in listGS3 do
>       t := CharacterTable( input[1] );
>       tC:= CharacterTable( input[2] );
```

```
>        tK:= CharacterTable( input[3] );
>        identifier:= input[4];
>        elms:= PossibleActionsForTypeGS3( t, tC, tK );
>        if Length( elms ) = 1 then
>          ProcessGS3Example( t, tC, tK, identifier, elms[1] );
>        else
>          Print( "#I  ", Length( elms ), " actions possible for '",
>                 identifier, "'\n" );
>        fi;
>    od;
#I  not all inputs available for 'O8+(3).S3 mod 3'
#I  not all inputs available for '2E6(2).S3 mod 2'
#I  not all inputs available for '2E6(2).S3 mod 3'
#I  not all inputs available for '2E6(2).S3 mod 5'
#I  not all inputs available for '2E6(2).S3 mod 7'
#I  not all inputs available for '2E6(2).S3 mod 11'
#I  not all inputs available for '2E6(2).S3 mod 13'
#I  not all inputs available for '2E6(2).S3 mod 17'
#I  not all inputs available for '2E6(2).S3 mod 19'
#I  not all inputs available for '2^2.2E6(2).S3 mod 2'
#I  not all inputs available for '2^2.2E6(2).S3 mod 3'
#I  not all inputs available for '2^2.2E6(2).S3 mod 5'
#I  not all inputs available for '2^2.2E6(2).S3 mod 7'
#I  not all inputs available for '2^2.2E6(2).S3 mod 11'
#I  not all inputs available for '2^2.2E6(2).S3 mod 13'
#I  not all inputs available for '2^2.2E6(2).S3 mod 17'
#I  not all inputs available for '2^2.2E6(2).S3 mod 19'
#I  not all inputs available for 'U3(8).(S3x3) mod 2'
#I  not all inputs available for 'U3(8).(S3x3) mod 19'
#I  not all inputs available for 'O8+(3).S4 mod 3'
```

Also the ordinary character table of the automorphic extension of the simple Atlas group $O_8^+(3)$ by A_4 can be constructed with the same approach. Here we get four possible permutations, which lead to essentially the same character table.

```
                          ____ Example ____
 gap> input:= [ "O8+(3)", "O8+(3).3", "O8+(3).(2^2)_{111}", "O8+(3).A4" ];;
 gap> t := CharacterTable( input[1] );;
 gap> tC:= CharacterTable( input[2] );;
 gap> tK:= CharacterTable( input[3] );;
 gap> identifier:= input[4];;
 gap> elms:= PossibleActionsForTypeGS3( t, tC, tK );;
 gap> Length( elms );
 4
 gap> differ:= MovedPoints( Group( List( elms, x -> x / elms[1] ) ) );;
 gap> List( elms, x -> RestrictedPerm( x, differ ) );
 [ (118,216,169)(119,217,170)(120,218,167)(121,219,168),
   (118,216,170)(119,217,169)(120,219,168)(121,218,167),
   (118,217,169)(119,216,170)(120,218,168)(121,219,167),
   (118,217,170)(119,216,169)(120,219,167)(121,218,168) ]
 gap> poss:= List( elms, pi -> CharacterTableOfTypeGS3( t, tC, tK, pi,
 >            Concatenation( identifier, "new" ) ) );;
 gap> lib:= CharacterTable( identifier );;
```

```
gap> ForAll( poss, r -> IsRecord(
>         TransformingPermutationsCharacterTables( r.table, lib ) ) );
true
```

Also the construction of the p-modular tables of $O_8^+(3).A_4$ works.

```
———————————— Example ————————————
gap> ProcessGS3Example( t, tC, tK, identifier, elms[1] );
#I  not all inputs available for 'O8+(3).A4 mod 3'
```

2.6   Examples for the Type $G.2^2$

2.6.1   The Character Table of $A_6.2^2$

As the first example, we consider the automorphism group $\mathrm{Aut}(A_6) \cong A_6.2^2$ of the alternating group A_6 on six points.

In this case, the triple of actions on the subgroups $A_6.2_i$ is uniquely determined by the condition on the number of conjugacy classes in Section 2.3.3.

```
———————————— Example ————————————
gap> tblG:= CharacterTable( "A6" );;
gap> tblsG2:= List( [ "A6.2_1", "A6.2_2", "A6.2_3" ], CharacterTable );;
gap> List( tblsG2, NrConjugacyClasses );
[ 11, 11, 8 ]
gap> possact:= List( tblsG2, x -> Filtered( Elements(
>         AutomorphismsOfTable( x ) ), y -> Order( y ) <= 2 ) );
[ [ (), (3,4)(7,8)(10,11) ],
  [ (), (8,9), (5,6)(10,11), (5,6)(8,9)(10,11) ], [ (), (7,8) ] ]
```

Note that $n_1 = n_2$ implies $f_1 = f_2$, and $n_1 - n_3 = 3$ implies $f_1 - f_3 = 2$, so we get $f_1 = 3$ and $f_3 = 1$, and $A_6.2^2$ has $2 \cdot 11 - 3 \cdot 3 = 2 \cdot 8 - 3 \cdot 1 = 13$ classes.

(The compatibility on the classes inside A_6 yields only that the classes 3 and 4 of $A_6.2_1 \cong S_6$ must be fused in $A_6.2^2$, as well as the classes 5 and 6 of $A_6.2_2 \cong \mathrm{PGL}(2,9)$.)

```
———————————— Example ————————————
gap> List( tblsG2, x -> GetFusionMap( tblG, x ) );
[ [ 1, 2, 3, 4, 5, 6, 6 ], [ 1, 2, 3, 3, 4, 5, 6 ],
  [ 1, 2, 3, 3, 4, 5, 5 ] ]
```

These arguments are used by the GAP function `PossibleActionsForTypeGV4` (**CTblLib: PossibleActionsForTypeGV4**), which returns the list of all possible triples of permutations such that the i-th permutation describes the action of $A_6.2^2$ on the classes of $A_6.2_i$.

```
———————————— Example ————————————
gap> acts:= PossibleActionsForTypeGV4( tblG, tblsG2 );
[ [ (3,4)(7,8)(10,11), (5,6)(8,9)(10,11), (7,8) ] ]
```

For the given actions, the GAP function `PossibleCharacterTablesOfTypeGV4` (**CTblLib: PossibleCharacterTablesOfTypeGV4**) then computes the possibilities for the character table of $A_6.2^2$; in this case, the result is unique.

```
──────────────────── Example ────────────────────
gap> poss:= PossibleCharacterTablesOfTypeGV4( tblG, tblsG2, acts[1],
>             "A6.2^2" );
[ rec(
      G2fusGV4 := [ [ 1, 2, 3, 3, 4, 5, 6, 6, 7, 8, 8 ],
          [ 1, 2, 3, 4, 5, 5, 9, 10, 10, 11, 11 ],
          [ 1, 2, 3, 4, 5, 12, 13, 13 ] ],
      table := CharacterTable( "A6.2^2" ) ) ]
gap> IsRecord( TransformingPermutationsCharacterTables( poss[1].table,
>                 CharacterTable( "A6.2^2" ) ) );
true
```

Finally, possible *p*-modular tables can be computed from the *p*-modular input tables and the ordinary table of $A_6.2^2$; here we show this for $p = 3$.

```
──────────────────── Example ────────────────────
gap> PossibleCharacterTablesOfTypeGV4( tblG mod 3,
>       List( tblsG2, t -> t mod 3 ), poss[1].table );
[ rec(
      G2fusGV4 :=
        [ [ 1, 2, 3, 4, 5, 5, 6 ], [ 1, 2, 3, 4, 4, 7, 8, 8, 9, 9 ],
          [ 1, 2, 3, 4, 10, 11, 11 ] ],
      table := BrauerTable( "A6.2^2", 3 ) ) ]
```

2.6.2 **Atlas** Tables of the Type $G.2^2$ – **Easy Cases**

We demonstrate the construction of all those ordinary and modular character tables in the GAP Character Table Library that are of the type $G.2^2$ where G is a simple group or a central extension of a simple group whose character table is contained in the Atlas. Here is the list of Identifier (**Reference: Identifier for tables of marks**) values needed for accessing the input tables and the result tables.

(The construction of the character table of $O_8^+(3).2_{111}^2$ is more involved and will be described in Section 2.6.10. The construction of the character tables of groups of the type $2.L_3(4).2^2$ and $6.L_3(4).2^2$ is described in the sections 2.6.4 and 2.6.5, respectively. The construction of the character tables of groups of the type $2.U_4(3).2^2$ is described in Section 2.6.6.)

```
──────────────────── Example ────────────────────
gap> listGV4:= [
> [ "A6",       "A6.2_1",       "A6.2_2",       "A6.2_3",       "A6.2^2"      ],
> [ "3.A6",     "3.A6.2_1",     "3.A6.2_2",     "3.A6.2_3",     "3.A6.2^2"    ],
> [ "L2(25)",   "L2(25).2_1",   "L2(25).2_2",   "L2(25).2_3",   "L2(25).2^2"  ],
> [ "L3(4)",    "L3(4).2_1",    "L3(4).2_2",    "L3(4).2_3",    "L3(4).2^2"   ],
> [ "2^2.L3(4)", "2^2.L3(4).2_1", "2^2.L3(4).2_2", "2^2.L3(4).2_3",
>                                                   "2^2.L3(4).2^2" ],
> [ "3.L3(4)",  "3.L3(4).2_1",  "3.L3(4).2_2",  "3.L3(4).2_3",  "3.L3(4).2^2" ],
> [ "U4(3)",    "U4(3).2_1",    "U4(3).2_2",    "U4(3).2_2'",
>                                                   "U4(3).(2^2)_{122}" ],
> [ "U4(3)",    "U4(3).2_1",    "U4(3).2_3",    "U4(3).2_3'",
>                                                   "U4(3).(2^2)_{133}" ],
> [ "3_1.U4(3)", "3_1.U4(3).2_1", "3_1.U4(3).2_2", "3_1.U4(3).2_2'",
>                                                   "3_1.U4(3).(2^2)_{122}" ],
> [ "3_2.U4(3)", "3_2.U4(3).2_1", "3_2.U4(3).2_3", "3_2.U4(3).2_3'",
>                                                   "3_2.U4(3).(2^2)_{133}" ],
```

```
> [ "L2(49)",   "L2(49).2_1", "L2(49).2_2", "L2(49).2_3", "L2(49).2^2" ],
> [ "L2(81)",   "L2(81).2_1", "L2(81).2_2", "L2(81).2_3", "L2(81).2^2" ],
> [ "L3(9)",    "L3(9).2_1",  "L3(9).2_2",  "L3(9).2_3",  "L3(9).2^2"  ],
> [ "O8+(3)",   "O8+(3).2_1", "O8+(3).2_2", "O8+(3).2_2'",
>                                            "O8+(3).(2^2)_{122}" ],
> [ "O8-(3)",   "O8-(3).2_1", "O8-(3).2_2", "O8-(3).2_3", "O8-(3).2^2" ],
> ];;
```

Analogously, the automorphism groups $L_3(4).D_{12}$ of $L_3(4)$ and $U_4(3).D_8$ of $U_4(3)$, and the subgroup $O_8^+(3).D_8$ of the automorphism group $O_8^+(3).S_4$ have factor groups that are isomorphic with 2^2; in these cases, we choose $G = L_3(4).3$, $G = U_4(3).2_1$, and $G = O_8^+(3).2_1$, respectively.

Also the group $2^2.L_3(4).D_{12}$ has a factor group isomorphic with 2^2. Note that the character tables of $L_3(4).D_{12}$ and $2^2.L_3(4).D_{12}$ have been constructed already in Section 2.5.2.

The automorphism groups of $L_4(4)$ and $U_4(5)$ have the structure $L_4(4).2^2$ and $U_4(5).2^2$, respectively; their tables are contained in the GAP Character Table Library but not in the Atlas.

```
─────────────────────────── Example ───────────────────────────
gap> Append( listGV4, [
> [ "L3(4).3", "L3(4).6",     "L3(4).3.2_2", "L3(4).3.2_3", "L3(4).D12"  ],
> [ "2^2.L3(4).3", "2^2.L3(4).6", "2^2.L3(4).3.2_2", "2^2.L3(4).3.2_3",
>                                            "2^2.L3(4).D12" ],
> [ "U4(3).2_1", "U4(3).4", "U4(3).(2^2)_{122}", "U4(3).(2^2)_{133}",
>                                            "U4(3).D8" ],
> [ "O8+(3).2_1", "O8+(3).(2^2)_{111}", "O8+(3).(2^2)_{122}", "O8+(3).4",
>                                            "O8+(3).D8" ],
> [ "L4(4)",    "L4(4).2_1",  "L4(4).2_2",  "L4(4).2_3",  "L4(4).2^2"  ],
> [ "U4(5)",    "U4(5).2_1",  "U4(5).2_2",  "U4(5).2_3",  "U4(5).2^2"  ],
> ] );
```

Now we proceed in two steps, the computation of the possible ordinary character tables from the ordinary tables of the relevant subgroups, and then the computation of the Brauer tables from the Brauer tables of the relevant subgroups and from the ordinary table of the group.

The following function first computes the possible triples of actions on the subgroups $G.2_i$, using the function `PossibleActionsForTypeGV4` (**CTblLib: PossibleActionsForTypeGV4**). Then the union of the candidate tables for these actions is computed, this list is returned in the end. and representatives of classes of permutation equivalent candidates are inspected further with consistency checks. If there is a unique solution up to permutation equivalence, this table is compared with the one that is contained in the GAP Character Table Library.

```
─────────────────────────── Example ───────────────────────────
gap> ConstructOrdinaryGV4Table:= function( tblG, tblsG2, name, lib )
>       local acts, nam, poss, reps, i, trans;
>
>       # Compute the possible actions for the ordinary tables.
>       acts:= PossibleActionsForTypeGV4( tblG, tblsG2 );
>       # Compute the possible ordinary tables for the given actions.
>       nam:= Concatenation( "new", name );
>       poss:= Concatenation( List( acts, triple ->
>           PossibleCharacterTablesOfTypeGV4( tblG, tblsG2, triple, nam ) ) );
>       # Test the possibilities for permutation equivalence.
>       reps:= RepresentativesCharacterTables( poss );
```

```
>       if 1 < Length( reps ) then
>         Print( "#I  ", name, ": ", Length( reps ),
>               " equivalence classes\n" );
>       elif Length( reps ) = 0 then
>         Print( "#E  ", name, ": no solution\n" );
>       else
>         # Compare the computed table with the library table.
>         if not IsCharacterTable( lib ) then
>           Print( "#I  no library table for ", name, "\n" );
>           PrintToLib( name, poss[1].table );
>           for i in [ 1 .. 3 ] do
>             Print( LibraryFusion( tblsG2[i],
>                        rec( name:= name, map:= poss[1].G2fusGV4[i] ) ) );
>           od;
>         else
>           trans:= TransformingPermutationsCharacterTables( poss[1].table,
>                       lib );
>           if not IsRecord( trans ) then
>             Print( "#E  computed table and library table for ", name,
>                   " differ\n" );
>           fi;
>           # Compare the computed fusions with the stored ones.
>           if List( poss[1].G2fusGV4, x -> OnTuples( x, trans.columns ) )
>                   <> List( tblsG2, x -> GetFusionMap( x, lib ) ) then
>             Print( "#E  computed and stored fusions for ", name,
>                   " differ\n" );
>           fi;
>         fi;
>       fi;
>       return poss;
>     end;;
```

The following function computes, for all those prime divisors p of the group order in question such that the p-modular Brauer tables of the subgroups $G.2_i$ are available, the possible p-modular Brauer tables. If the solution is unique up to permutation equivalence, it is compared with the table that is contained in the GAP Character Table Library.

It may happen (even in the case that the ordinary character table is unique up to permutation equivalence) that some candidates for the ordinary character table are excluded due to information provided by some p-modular table. In this case, a message is printed, and the ordinary character table from the GAP Character Table Library is checked again under the additional restrictions.

```
────────── Example ──────────
gap> ConstructModularGV4Tables:= function( tblG, tblsG2, ordposs,
>                                           ordlibtblGV4 )
>     local name, modposs, primes, checkordinary, i, record, p, tmodp,
>           t2modp, poss, modlib, trans, reps;
>
>     if not IsCharacterTable( ordlibtblGV4 ) then
>       Print( "#I  no ordinary library table ...\n" );
>       return [];
>     fi;
>     name:= Identifier( ordlibtblGV4 );
```

```
>        modposs:= List( ordposs, x -> [] );
>        primes:= ShallowCopy( PrimeDivisors( Size( tblG ) ) );
>        ordposs:= ShallowCopy( ordposs );
>        checkordinary:= false;
>        for i in [ 1 .. Length( ordposs ) ] do
>          record:= ordposs[i];
>          for p in primes do
>            tmodp := tblG  mod p;
>            t2modp:= List( tblsG2, t2 -> t2 mod p );
>            if IsCharacterTable( tmodp ) and
>               ForAll( t2modp, IsCharacterTable ) then
>              poss:= PossibleCharacterTablesOfTypeGV4( tmodp, t2modp,
>                       record.table, record.G2fusGV4 );
>              poss:= RepresentativesCharacterTables( poss );
>              if   Length( poss ) = 0 then
>                Print( "#I  excluded cand. ", i, " (out of ",
>                       Length( ordposs ), ") for ", name, " by ", p,
>                       "-mod. table\n" );
>                Unbind( ordposs[i] );
>                Unbind( modposs[i] );
>                checkordinary:= true;
>                break;
>              elif Length( poss ) = 1 then
>                # Compare the computed table with the library table.
>                modlib:= ordlibtblGV4 mod p;
>                if IsCharacterTable( modlib ) then
>                  trans:= TransformingPermutationsCharacterTables(
>                            poss[1].table, modlib );
>                  if not IsRecord( trans ) then
>                    Print( "#E  computed table and library table for ",
>                           name, " mod ", p, " differ\n" );
>                  fi;
>                else
>                  Print( "#I  no library table for ",
>                         name, " mod ", p, "\n" );
>                  PrintToLib( name, poss[1].table );
>                fi;
>              else
>                Print( "#I  ", name, " mod ", p, ": ", Length( poss ),
>                       " equivalence classes\n" );
>              fi;
>              Add( modposs[i], poss );
>            elif i = 1 then
>              Print( "#I  not all input tables for ", name, " mod ", p,
>                     " available\n" );
>              primes:= Difference( primes, [ p ] );
>            fi;
>          od;
>        od;
>        if checkordinary then
>          # Test whether the ordinary table is admissible.
>          ordposs:= Compacted( ordposs );
```

```
>         modposs:= Compacted( modposs );
>         reps:= RepresentativesCharacterTables( ordposs );
>         if 1 < Length( reps ) then
>           Print( "#I  ", name, ": ", Length( reps ),
>                 " equivalence classes (ord. table)\n" );
>         elif Length( reps ) = 0 then
>           Print( "#E  ", name, ": no solution (ord. table)\n" );
>         else
>           # Compare the computed table with the library table.
>           trans:= TransformingPermutationsCharacterTables(
>                     ordposs[1].table, ordlibtblGV4 );
>           if not IsRecord( trans ) then
>             Print( "#E  computed table and library table for ", name,
>                   " differ\n" );
>           fi;
>           # Compare the computed fusions with the stored ones.
>           if List( ordposs[1].G2fusGV4, x -> OnTuples( x, trans.columns ) )
>                 <> List( tblsG2, x -> GetFusionMap( x, ordlibtblGV4 ) ) then
>             Print( "#E  computed and stored fusions for ", name,
>                   " differ\n" );
>           fi;
>         fi;
>       fi;
>       return rec( ordinary:= ordposs, modular:= modposs );
>     end;;
```

Finally, here is the loop over the list of tables.

```
                              Example
gap> for input in listGV4 do
>       tblG   := CharacterTable( input[1] );
>       tblsG2 := List( input{ [ 2 .. 4 ] }, CharacterTable );
>       lib    := CharacterTable( input[5] );
>       poss   := ConstructOrdinaryGV4Table( tblG, tblsG2, input[5], lib );
>       ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
>    od;
#I  excluded cand. 2 (out of 2) for L3(4).2^2 by 3-mod. table
#I  excluded cand. 2 (out of 8) for 2^2.L3(4).2^2 by 7-mod. table
#I  excluded cand. 3 (out of 8) for 2^2.L3(4).2^2 by 5-mod. table
#I  excluded cand. 4 (out of 8) for 2^2.L3(4).2^2 by 5-mod. table
#I  excluded cand. 5 (out of 8) for 2^2.L3(4).2^2 by 5-mod. table
#I  excluded cand. 6 (out of 8) for 2^2.L3(4).2^2 by 5-mod. table
#I  excluded cand. 7 (out of 8) for 2^2.L3(4).2^2 by 7-mod. table
#I  excluded cand. 2 (out of 2) for 3.L3(4).2^2 by 3-mod. table
#I  excluded cand. 2 (out of 2) for L3(9).2^2 by 7-mod. table
#I  not all input tables for O8+(3).(2^2)_{122} mod 3 available
#I  not all input tables for O8-(3).2^2 mod 3 available
#I  not all input tables for O8-(3).2^2 mod 5 available
#I  not all input tables for O8-(3).2^2 mod 7 available
#I  not all input tables for O8-(3).2^2 mod 13 available
#I  not all input tables for O8-(3).2^2 mod 41 available
#I  excluded cand. 2 (out of 2) for L3(4).D12 by 3-mod. table
#I  excluded cand. 2 (out of 2) for 2^2.L3(4).D12 by 7-mod. table
```

```
#I  not all input tables for O8+(3).D8 mod 3 available
#I  not all input tables for L4(4).2^2 mod 3 available
#I  not all input tables for L4(4).2^2 mod 5 available
#I  not all input tables for L4(4).2^2 mod 7 available
#I  not all input tables for L4(4).2^2 mod 17 available
#I  not all input tables for U4(5).2^2 mod 2 available
#I  not all input tables for U4(5).2^2 mod 3 available
#I  not all input tables for U4(5).2^2 mod 5 available
#I  not all input tables for U4(5).2^2 mod 7 available
#I  not all input tables for U4(5).2^2 mod 13 available
```

The groups $3.A_6.2^2$, $3.L_3(4).2^2$, and $3_2.U_4(3).(2^2)_{133}$ have also the structure $M.G.A$, with $M.G$ equal to $3.A_6.2_3$, $3.L_3(4).2_1$, and $3_2.U_4(3).2_3$, respectively, and $G.A$ equal to $A_6.2^2$, $L_3(4).2^2$, and $U_4(3).(2^2)_{133}$, respectively (see Section 2.4.3).

Similarly, the group $L_3(4).D_{12}$ has also the structure $G.S_3$, with $G = L_3(4).2_1$, $G.2 = L_3(4).2^2$, and $G.3 = L_3(4).6$, respectively (see Section 2.5.2).

2.6.3 The Character Table of $S_4(9).2^2$ (September 2011)

The available functions yield two possibilities for the ordinary character table of $S_4(9).2^2$.

```
──────────── Example ────────────
gap> tblG:= CharacterTable( "S4(9)" );;
gap> tblsG2:= List( [ "S4(9).2_1", "S4(9).2_2", "S4(9).2_3" ],
>                    CharacterTable );;
gap> lib:= CharacterTable( "S4(9).2^2" );;
gap> poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, "newS4(9).2^2", lib );;
#I  newS4(9).2^2: 2 equivalence classes
gap> poss:= RepresentativesCharacterTables( poss );;
```

The two candidates differ w. r. t. the action of $S_4(9).2^2$ on the classes of element order 80 in $S_4(9).2_2$. In the two possible tables, each element of order 80 is conjugate to its third power or to its 13-th power, respectively.

```
──────────── Example ────────────
gap> order80:= PositionsProperty( OrdersClassRepresentatives( tblsG2[2] ),
>                  x -> x = 80 );
[ 98, 99, 100, 101, 102, 103, 104, 105 ]
gap> List( poss, r -> r.G2fusGV4[2]{ order80 } );
[ [ 77, 77, 78, 79, 80, 78, 79, 80 ],
  [ 77, 78, 79, 79, 77, 80, 80, 78 ] ]
gap> PowerMap( tblsG2[2], 3 ){ order80 };
[ 99, 98, 103, 104, 105, 100, 101, 102 ]
gap> PowerMap( tblsG2[2], 13 ){ order80 };
[ 102, 105, 101, 100, 98, 104, 103, 99 ]
```

We claim that the first candidate is the correct one. For that, first note that $S_4(9).2_2$ is the extension of the simple group by a diagonal automorphism. (An easy way to see this is that for any subgroup of $S_4(9)$ isomorphic with $S_2(81) \cong L_2(81)$, the extension by a diagonal automorphism contains elements of order 80 –this group is isomorphic with PGL$(2,81)$– and only $S_4(9).2_2$ contains elements of order 80.)

```
──────────── Example ────────────
gap> List( tblsG2, x -> 80 in OrdersClassRepresentatives( x ) );
[ false, true, false ]
```

Now the field automorphism of $S_4(9).2_2$ maps each element x of order 80 in $S_4(9).2_2$ to a conjugate of x^3.

```
──────────── Example ────────────
gap> tbl:= poss[1].table;;
gap> IsRecord( TransformingPermutationsCharacterTables( tbl, lib ) );
true
```

2.6.4 The Character Tables of Groups of the Type $2.L_3(4).2^2$ (June 2010)

The outer automorphism group of the group $L_3(4)$ is a dihedral group of order 12; its Sylow 2-subgroups are Klein four groups, so there is a unique almost simple group H of the type $L_3(4).2^2$, up to isomorphism. In this section, we construct the character tables of the double covers of this group with the approach from Section 2.3.3.

The group H has three subgroups of index two, of the types $L_3(4).2_1$, $L_3(4).2_2$, and $L_3(4).2_3$, respectively. So any double cover of H contains one subgroup of each of the types $2.L_3(4).2_1$, $2.L_3(4).2_2$, and $2.L_3(4).2_3$, and there are two isoclinic variants of each of these group to consider, see Section 2.2.6. So we start with eight different inputs for the construction of the character tables of double covers.

```
──────────── Example ────────────
gap> names:= List( [ 1 .. 3 ],
>                  i -> Concatenation( "2.L3(4).2_", String( i ) ) );;
gap> tbls:= List( names, CharacterTable );
[ CharacterTable( "2.L3(4).2_1" ), CharacterTable( "2.L3(4).2_2" ),
  CharacterTable( "2.L3(4).2_3" ) ]
gap> isos:= List( names, nam -> CharacterTable( Concatenation( nam, "*" ) ) );
[ CharacterTable( "Isoclinic(2.L3(4).2_1)" ),
  CharacterTable( "Isoclinic(2.L3(4).2_2)" ),
  CharacterTable( "Isoclinic(2.L3(4).2_3)" ) ]
gap> inputs:= [
> [ tbls[1], tbls[2], tbls[3], "2.L3(4).(2^2)_{123}" ],
> [ tbls[1], isos[2], tbls[3], "2.L3(4).(2^2)_{12*3}" ],
> [ tbls[1], tbls[2], isos[3], "2.L3(4).(2^2)_{123*}" ],
> [ tbls[1], isos[2], isos[3], "2.L3(4).(2^2)_{12*3*}" ],
> [ isos[1], tbls[2], tbls[3], "2.L3(4).(2^2)_{1*23}" ],
> [ isos[1], isos[2], tbls[3], "2.L3(4).(2^2)_{1*2*3}" ],
> [ isos[1], tbls[2], isos[3], "2.L3(4).(2^2)_{1*23*}" ],
> [ isos[1], isos[2], isos[3], "2.L3(4).(2^2)_{1*2*3*}" ] ];;
gap> tblG:= CharacterTable( "2.L3(4)" );;
gap> result:= [];;
gap> for input in inputs do
>       tblsG2:= input{ [ 1 .. 3 ] };
>       lib:= CharacterTable( input[4] );
>       poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, input[4], lib );
>       ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
>       Append( result, RepresentativesCharacterTables( poss ) );
>    od;
```

```
#I  excluded cand. 2 (out of 8) for 2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 2.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 2.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 2.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 2.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 2.L3(4).(2^2)_{12*3*} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 2.L3(4).(2^2)_{12*3*} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 2.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 2.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 2.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 2.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 2.L3(4).(2^2)_{1*2*3} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 2.L3(4).(2^2)_{1*2*3} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 2.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 2.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 2.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 2.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 2.L3(4).(2^2)_{1*23*} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 2.L3(4).(2^2)_{1*23*} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 2.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 2.L3(4).(2^2)_{1*23*} by
5-mod. table
gap> result:= List( result, x -> x.table );
[ CharacterTable( "new2.L3(4).(2^2)_{123}" ),
  CharacterTable( "new2.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new2.L3(4).(2^2)_{123*}" ),
```

```
CharacterTable( "new2.L3(4).(2^2)_{12*3*}" ),
CharacterTable( "new2.L3(4).(2^2)_{1*23}" ),
CharacterTable( "new2.L3(4).(2^2)_{1*2*3}" ),
CharacterTable( "new2.L3(4).(2^2)_{1*23*}" ),
CharacterTable( "new2.L3(4).(2^2)_{1*2*3*}" ) ]
```

We get exactly one character table for each input. For each of these tables, there are three possibilities to form an isoclinic table, corresponding to the three subgroups of index two. It turns out that the eight solutions form two orbits under forming some isoclinic table. Tables in different orbits are essentially different, already their numbers of conjugacy classes differ.

```
─────────────────────────── Example ───────────────────────────
gap> List( result, NrConjugacyClasses );
[ 39, 33, 33, 39, 33, 39, 39, 33 ]
gap> t:= result[1];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 4, 7, 6 ]
gap> t:= result[2];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 3, 8, 5 ]
```

Up to now, it is not clear that the character tables we have constructed are really the character tables of some groups. The existence of groups for the first orbit of character tables can be established as follows.

The group $U_6(2).2$ contains a maximal subgroup M of the type $L_3(4).2^2$, see [CCN+85, p. 115]. Its derived subgroup M' of the type $L_3(4)$ lies inside $U_6(2)$, and we claim that the preimage of M' under the natural epimorphism from $2.U_6(2)$ to $U_6(2)$ is a double cover of $L_3(4)$. Unfortunately, $L_3(4)$ admits class fusions into $2.U_6(2)$, so this criterion cannot be used.

```
─────────────────────────── Example ───────────────────────────
gap> l34:= CharacterTable( "L3(4)" );;
gap> u:= CharacterTable( "U6(2)" );;
gap> 2u:= CharacterTable( "2.U6(2)" );;
gap> cand:= PossibleClassFusions( l34, 2u );
[ [ 1, 5, 12, 16, 22, 22, 23, 23, 41, 41 ],
  [ 1, 5, 12, 22, 16, 22, 23, 23, 41, 41 ],
  [ 1, 5, 12, 22, 22, 16, 23, 23, 41, 41 ] ]
gap> OrdersClassRepresentatives( l34 );
[ 1, 2, 3, 4, 4, 4, 5, 5, 7, 7 ]
```

Consider the three classes of elements of order four in $L_3(4)$. Under the possible fusions into $2.U_6(2)$, they are mapped to the classes 16 and 22, which are preimages of the classes 10 and 14 (4C and 4G) of $U_6(2)$. Note that the maximal subgroups of type $L_3(4).2$ in $U_6(2)$ extend to $L_3(4).6$ type

subgroups in $U_6(2).3$, and the three classes 4C, 4D, 4E form one orbit under the action of an outer automorphism of order three of $U_6(2)$.

```
 ─────────────────────────────── Example ───────────────────────────────
  gap> GetFusionMap( 2u, u ){ [ 16, 22 ] };
  [ 10, 14 ]
  gap> ClassNames( u, "ATLAS" ){ [ 10, 14 ] };
  [ "4C", "4G" ]
  gap> GetFusionMap( u, CharacterTable( "U6(2).3" ) );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 11, 12, 13, 14, 15, 16, 17,
    18, 19, 20, 21, 22, 23, 24, 24, 24, 25, 26, 27, 28, 29, 30, 31, 32,
    33, 34, 35, 36, 36, 36, 37, 38, 39, 40 ]
```

This means that any $L_3(4)$ type subgroup of $U_6(2)$ that extends to an $L_3(4).6$ type subgroup in $U_6(2).3$ either contains elements from all three classes 4C, 4D, 4E of $U_6(2)$, or contains no element from these classes. Thus we know that any double cover of $U_6(2).2$ contains a double cover of $L_3(4).2^2$. Only the first of our result tables admits a class fusion into the table of $2.U_6(2).2$.

```
 ─────────────────────────────── Example ───────────────────────────────
  gap> 2u2:= CharacterTable( "2.U6(2).2" );;
  gap> fus:= List( result, x -> PossibleClassFusions( x, 2u2 ) );;
  gap> List( fus, Length );
  [ 4, 0, 0, 0, 0, 0, 0, 0 ]
```

As a consequence, the fourth result table is established as that of a maximal subgroup of the group isoclinic to $2.U_6(2).2$.

```
 ─────────────────────────────── Example ───────────────────────────────
  gap> 2u2iso:= CharacterTableIsoclinic( 2u2 );;
  gap> fus:= List( result, x -> PossibleClassFusions( x, 2u2iso ) );;
  gap> List( fus, Length );
  [ 0, 0, 0, 4, 0, 0, 0, 0 ]
```

Similarly, the group $HS.2$ contains a maximal subgroup M of the type $L_3(4).2^2$, see [CCN$^+$85, p. 80]. Its derived subgroup M' of the type $L_3(4)$ lies inside HS, and the preimage of M' under the natural epimorphism from $2.HS$ to HS is a double cover of $L_3(4)$, because $L_3(4)$ does not admit a class fusion into $2.HS.2$.

```
 ─────────────────────────────── Example ───────────────────────────────
  gap> h2:= CharacterTable( "HS.2" );;
  gap> 2h2:= CharacterTable( "2.HS.2" );;
  gap> PossibleClassFusions( l34, 2h2 );
  [ ]
```

Only the fifth of our result tables admits a class fusion into $2.HS.2$, which means that $2.L_3(4).(2^2)_{1*23}$ is a subgroup of $2.HS.2$, and the eighth result table $-2.L_3(4).(2^2)_{1*2*3*}-$ admits a class fusion into the isoclinic variant of $2.HS.2$ This shows the existence of groups for the tables from the second orbit.

```
 ─────────────────────────────── Example ───────────────────────────────
  gap> fus:= List( result, x -> PossibleClassFusions( x, 2h2 ) );;
  gap> List( fus, Length );
  [ 0, 0, 0, 0, 4, 0, 0, 0 ]
```

```
gap> 2h2iso:= CharacterTableIsoclinic( 2h2 );;
gap> fus:= List( result, x -> PossibleClassFusions( x, 2h2iso ) );;
gap> List( fus, Length );
[ 0, 0, 0, 0, 0, 0, 0, 4 ]
```

2.6.5 The Character Tables of Groups of the Type $6.L_3(4).2^2$ (October 2011)

We have two approaches for constructing the character tables of these groups.

First, we may regard them as normal products of the three normal subgroups of index two, each of them having the structure $6.L_3(4).2$, and use the approach from Section 2.3.3, as we did in Section 2.6.4 for the groups of the structure $2.L_3(4).2^2$.

Second, we may use the approach from Section 2.3.1. Note that the factor group $L_3(4).2^2$ contains each of the three groups $L_3(4).2_i$ as a subgroup, for $1 \le i \le 3$, and the groups of the type $6.L_3(4).2_1$ have a centre of order six, whereas the centres of the $6.L_3(4).2_2$ and $6.L_3(4).2_3$ type groups have order two. For that, the character tables of the subgroups $6.L_3(4).2_1$ and $6.L_3(4).2_1^*$ are needed, as well as the character tables of the eight possible factor groups $2.L_3(4).2^2$; the latter tables are known from Section 2.6.4.

We show both approaches. (The second approach is better suited for storing the character tables in the Character Table Library, since the irreducible characters need not be stored, and since the Brauer tables of the groups can be derived from the Brauer tables of the compound tables.)

```
──────────────────────── Example ────────────────────────
 gap> tbls:= List( [ "1", "2", "3" ],
 >      i -> CharacterTable( Concatenation( "6.L3(4).2_", i ) ) );
 [ CharacterTable( "6.L3(4).2_1" ), CharacterTable( "6.L3(4).2_2" ),
   CharacterTable( "6.L3(4).2_3" ) ]
 gap> isos:= List( [ "1", "2", "3" ],
 >      i -> CharacterTable( Concatenation( "6.L3(4).2_", i, "*" ) ) );
 [ CharacterTable( "Isoclinic(6.L3(4).2_1)" ),
   CharacterTable( "Isoclinic(6.L3(4).2_2)" ),
   CharacterTable( "Isoclinic(6.L3(4).2_3)" ) ]
 gap> inputs:= [
 > [ tbls[1], tbls[2], tbls[3], "6.L3(4).(2^2)_{123}" ],
 > [ tbls[1], isos[2], tbls[3], "6.L3(4).(2^2)_{12*3}" ],
 > [ tbls[1], tbls[2], isos[3], "6.L3(4).(2^2)_{123*}" ],
 > [ tbls[1], isos[2], isos[3], "6.L3(4).(2^2)_{12*3*}" ],
 > [ isos[1], tbls[2], tbls[3], "6.L3(4).(2^2)_{1*23}" ],
 > [ isos[1], isos[2], tbls[3], "6.L3(4).(2^2)_{1*2*3}" ],
 > [ isos[1], tbls[2], isos[3], "6.L3(4).(2^2)_{1*23*}" ],
 > [ isos[1], isos[2], isos[3], "6.L3(4).(2^2)_{1*2*3*}" ] ];;
 gap> tblG:= CharacterTable( "6.L3(4)" );;
 gap> result:= [];;
 gap> for input in inputs do
 >      tblsG2:= input{ [ 1 .. 3 ] };
 >      lib:= CharacterTable( input[4] );
 >      poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, input[4], lib );
 >      ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
 >      Append( result, RepresentativesCharacterTables( poss ) );
 >    od;
 #I  excluded cand. 2 (out of 8) for 6.L3(4).(2^2)_{123} by
 5-mod. table
```

```
#I  excluded cand. 3 (out of 8) for 6.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 6.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 6.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 6.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 6.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 6.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 6.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 6.L3(4).(2^2)_{12*3*} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 6.L3(4).(2^2)_{12*3*} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 6.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 6.L3(4).(2^2)_{12*3*} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 6.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 6.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 6.L3(4).(2^2)_{1*2*3} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 6.L3(4).(2^2)_{1*2*3} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 6.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 6.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 6.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 6.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 6.L3(4).(2^2)_{1*23*} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 6.L3(4).(2^2)_{1*23*} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 6.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 6.L3(4).(2^2)_{1*23*} by
5-mod. table
gap> result:= List( result, x -> x.table );
[ CharacterTable( "new6.L3(4).(2^2)_{123}" ),
  CharacterTable( "new6.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new6.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new6.L3(4).(2^2)_{12*3*}" ),
  CharacterTable( "new6.L3(4).(2^2)_{1*23}" ),
```

```
      CharacterTable( "new6.L3(4).(2^2)_{1*2*3}" ),
      CharacterTable( "new6.L3(4).(2^2)_{1*23*}" ),
      CharacterTable( "new6.L3(4).(2^2)_{1*2*3*}" ) ]
```

As in Section 2.6.4, we get exactly one character table for each input, and the eight solutions lie in two orbits under isoclinism.

```
─────────────────────── Example ───────────────────────
  gap> List( result, NrConjugacyClasses );
  [ 59, 53, 53, 59, 53, 59, 59, 53 ]
  gap> t:= result[1];;
  gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
  >           x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
  gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
  gap> List( iso, x -> PositionProperty( result, y ->
  >           TransformingPermutationsCharacterTables( x, y ) <> fail ) );
  [ 7, 6, 4 ]
  gap> t:= result[2];;
  gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
  >           x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
  gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
  gap> List( iso, x -> PositionProperty( result, y ->
  >           TransformingPermutationsCharacterTables( x, y ) <> fail ) );
  [ 8, 5, 3 ]
```

Up to now, it is not clear that the character tables we have constructed are really the character tables of some groups. The existence of groups for the first orbit of character tables can be established as follows.

We have shown in Section 2.6.4 that the maximal subgroups M of the type $L_3(4).2^2$ in $U_6(2).2$ lift to double covers $2.L_3(4).2^2$ in $2.U_6(2).2$. The preimages of these groups under the natural epimorphism from $6.U_6(2).2$ have the structure $6.L_3(4).2^2$, where the derived subgroup is the six-fold cover of $L_3(4)$; this follows from the fact that $6.U_6(2)$ does not admit a class fusion from the double cover $2.L_3(4)$.

```
─────────────────────── Example ───────────────────────
  gap> 2l34:= CharacterTable( "2.L3(4)" );;
  gap> 6u:= CharacterTable( "6.U6(2)" );;
  gap> cand:= PossibleClassFusions( 2l34, 6u );
  [  ]
```

This establishes the first and the fourth result as character tables of subgroups of $6.U_6(2)$ and its isoclinic variant, respectively.

```
─────────────────────── Example ───────────────────────
  gap> 6u2:= CharacterTable( "6.U6(2).2" );;
  gap> fus:= List( result, x -> PossibleClassFusions( x, 6u2 ) );;
  gap> List( fus, Length );
  [ 8, 0, 0, 0, 0, 0, 0, 0 ]
  gap> 6u2iso:= CharacterTableIsoclinic( 6u2 );;
  gap> fus:= List( result, x -> PossibleClassFusions( x, 6u2iso ) );;
  gap> List( fus, Length );
  [ 0, 0, 0, 8, 0, 0, 0, 0 ]
```

Similarly, the group $G_2(4).2$ contains a maximal subgroup M of the type $3.L_3(4).2^2$, see [CCN$^+$85, p. 97]. Its derived subgroup M' of the type $3.L_3(4)$ lies inside $G_2(4)$, and the preimage of M' under the natural epimorphism from $2.G_2(4)$ to $G_2(4)$ is a double cover of $3.L_3(4)$, because $3.L_3(4)$ does not admit a class fusion into $2.G_2(4).2$.

```
———————————————————————— Example ————————————————————————
gap> 3l34:= CharacterTable( "3.L3(4)" );;
gap> g2:= CharacterTable( "G2(4).2" );;
gap> 2g2:= CharacterTable( "2.G2(4).2" );;
gap> PossibleClassFusions( 3l34, 2g2 );
[  ]
```

Only the third and eighth of our result tables admit a class fusion into $2.G_2(4).2$ and its isoclinic variant, respectively. This shows the existence of groups for the tables from the second orbit.

```
———————————————————————— Example ————————————————————————
gap> fus:= List( result, x -> PossibleClassFusions( x, 2g2 ) );;
gap> List( fus, Length );
[ 0, 0, 16, 0, 0, 0, 0, 0 ]
gap> 2g2iso:= CharacterTableIsoclinic( 2g2 );;
gap> fus:= List( result, x -> PossibleClassFusions( x, 2g2iso ) );;
gap> List( fus, Length );
[ 0, 0, 0, 0, 0, 0, 0, 16 ]
```

Now we try the second approach and compare the results.

```
———————————————————————— Example ————————————————————————
gap> names:= [ "L3(4).(2^2)_{123}", "L3(4).(2^2)_{12*3}",
>              "L3(4).(2^2)_{123*}", "L3(4).(2^2)_{12*3*}" ];;
gap> inputs1:= List( names, nam -> [ "6.L3(4).2_1", "2.L3(4).2_1",
>        Concatenation( "2.", nam ), Concatenation( "6.", nam ) ] );;
gap> names:= List( names, nam -> ReplacedString( nam, "1", "1*" ) );;
gap> inputs2:= List( names, nam -> [ "6.L3(4).2_1*", "2.L3(4).2_1*",
>        Concatenation( "2.", nam ), Concatenation( "6.", nam ) ] );;
gap> inputs:= Concatenation( inputs1, inputs2 );
[ [ "6.L3(4).2_1", "2.L3(4).2_1", "2.L3(4).(2^2)_{123}",
      "6.L3(4).(2^2)_{123}" ],
  [ "6.L3(4).2_1", "2.L3(4).2_1", "2.L3(4).(2^2)_{12*3}",
      "6.L3(4).(2^2)_{12*3}" ],
  [ "6.L3(4).2_1", "2.L3(4).2_1", "2.L3(4).(2^2)_{123*}",
      "6.L3(4).(2^2)_{123*}" ],
  [ "6.L3(4).2_1", "2.L3(4).2_1", "2.L3(4).(2^2)_{12*3*}",
      "6.L3(4).(2^2)_{12*3*}" ],
  [ "6.L3(4).2_1*", "2.L3(4).2_1*", "2.L3(4).(2^2)_{1*23}",
      "6.L3(4).(2^2)_{1*23}" ],
  [ "6.L3(4).2_1*", "2.L3(4).2_1*", "2.L3(4).(2^2)_{1*2*3}",
      "6.L3(4).(2^2)_{1*2*3}" ],
  [ "6.L3(4).2_1*", "2.L3(4).2_1*", "2.L3(4).(2^2)_{1*23*}",
      "6.L3(4).(2^2)_{1*23*}" ],
  [ "6.L3(4).2_1*", "2.L3(4).2_1*", "2.L3(4).(2^2)_{1*2*3*}",
      "6.L3(4).(2^2)_{1*2*3*}" ] ]
gap> result2:= [];;
gap> for  input in inputs do
>        tblMG := CharacterTable( input[1] );
```

```
>        tblG   := CharacterTable( input[2] );
>        tblGA  := CharacterTable( input[3] );
>        name   := Concatenation( "new", input[4] );
>        lib    := CharacterTable( input[4] );
>        poss:= ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );
>        Append( result2, poss );
>    od;
gap> result2:= List( result2, x -> x.table );
[ CharacterTable( "new6.L3(4).(2^2)_{123}" ),
  CharacterTable( "new6.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new6.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new6.L3(4).(2^2)_{12*3*}" ),
  CharacterTable( "new6.L3(4).(2^2)_{1*23}" ),
  CharacterTable( "new6.L3(4).(2^2)_{1*2*3}" ),
  CharacterTable( "new6.L3(4).(2^2)_{1*23*}" ),
  CharacterTable( "new6.L3(4).(2^2)_{1*2*3*}" ) ]
gap> trans:= List( [ 1 .. 8 ], i ->
>        TransformingPermutationsCharacterTables( result[i],
>            result2[i] ) );;
gap> ForAll( trans, IsRecord );
true
```

2.6.6 The Character Tables of Groups of the Type $2.U_4(3).2^2$ (February 2012)

The outer automorphism group of the group $U_4(3)$ is a dihedral group of order 8. There are two almost simple groups of the type $U_4(3).2^2$, up to isomorphism, denoted as $U_4(3).(2^2)_{122}$ and $U_4(3).(2^2)_{133}$, respectively. Note that $U_4(3).2_1$ is the extension by the central involution of the outer automorphism group of $U_4(3)$, the other two subgroups of index two in $U_4(3).(2^2)_{122}$ are $U_4(3).2_2$ and $U_4(3).2'_2$, respectively, and the other two subgroups of index two in $U_4(3).(2^2)_{133}$ are $U_4(3).2_3$ and $U_4(3).2'_3$, respectively.

Since $\mathrm{Aut}(U_4(3))$ possesses a double cover (see [CCN+85, p. 52]), double covers of $U_4(3).(2^2)_{122}$ and $U_4(3).(2^2)_{133}$ exist.

First we deal with the double covers of $U_4(3).(2^2)_{122}$. Any such group contains one subgroup of the type $2.U_4(3).2_1$ and two subgroups of the type $2.U_4(3).2_2$, and there are two isoclinic variants of each of these group to consider, see Section 2.2.6. Thus we start with six different inputs for the construction of the character tables of double covers.

```
——————————————————— Example ———————————————————
 gap> tbls:= List( [ "1", "2", "2'" ], i ->
 >       CharacterTable( Concatenation( "2.U4(3).2_", i ) ) );;
 gap> isos:= List( [ "1", "2", "2'" ], i ->
 >       CharacterTable( Concatenation( "Isoclinic(2.U4(3).2_", i, ")" ) ) );;
 gap> inputs:= [
 > [ tbls[1], tbls[2], tbls[3], "2.U4(3).(2^2)_{122}" ],
 > [ isos[1], tbls[2], tbls[3], "2.U4(3).(2^2)_{1*22}" ],
 > [ tbls[1], isos[2], tbls[3], "2.U4(3).(2^2)_{12*2}" ],
 > [ isos[1], isos[2], tbls[3], "2.U4(3).(2^2)_{1*2*2}" ],
 > [ tbls[1], isos[2], isos[3], "2.U4(3).(2^2)_{12*2*}" ],
 > [ isos[1], isos[2], isos[3], "2.U4(3).(2^2)_{1*2*2*}" ] ];;
 gap> tblG:= CharacterTable( "2.U4(3)" );;
 gap> result:= [];;
```

```
gap> for input in inputs do
>       tblsG2:= input{ [ 1 .. 3 ] };
>       lib:= CharacterTable( input[4] );
>       poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, input[4], lib );
>       ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
>       Append( result, RepresentativesCharacterTables( poss ) );
>    od;
gap> result:= List( result, x -> x.table );
[ CharacterTable( "new2.U4(3).(2^2)_{122}" ),
  CharacterTable( "new2.U4(3).(2^2)_{1*22}" ),
  CharacterTable( "new2.U4(3).(2^2)_{12*2}" ),
  CharacterTable( "new2.U4(3).(2^2)_{1*2*2}" ),
  CharacterTable( "new2.U4(3).(2^2)_{12*2*}" ),
  CharacterTable( "new2.U4(3).(2^2)_{1*2*2*}" ) ]
```

We get exactly one character table for each input. For each of these tables, there are three possibilities to form an isoclinic table, corresponding to the three subgroups of index two. It turns out that the six solutions form two orbits under forming some isoclinic table. Tables in different orbits are essentially different, already their numbers of conjugacy classes differ.

```
───────────────────────── Example ─────────────────────────
gap> List( result, NrConjugacyClasses );
[ 87, 102, 102, 87, 87, 102 ]
gap> t:= result[1];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>          x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>          TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 4, 4, 5 ]
gap> t:= result[2];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>          x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>          TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 3, 3, 6 ]
```

Up to now, it is not clear that the character tables we have constructed are really the character tables of some groups. The existence of groups for the first orbit of character tables can be established as follows.

The group $O_8^+(3)$ contains maximal subgroups of the type $2.U_4(3).2^2$, see [CCN+85, p. 140]. Only the first of our result tables admits a class fusion into the table of $O_8^+(3)$.

```
───────────────────────── Example ─────────────────────────
gap> u:= CharacterTable( "O8+(3)" );;
gap> fus:= List( result, x -> PossibleClassFusions( x, u ) );;
gap> List( fus, Length );
[ 24, 0, 0, 0, 0, 0 ]
```

A table in the second orbit belongs to a maximal subgroup of $O_7(3).2$, see [CCN+85, p. 109].

```
                         ───── Example ─────
  gap> u:= CharacterTable( "O7(3).2" );;
  gap> fus:= List( result, x -> PossibleClassFusions( x, u ) );;
  gap> List( fus, Length );
  [ 0, 16, 0, 0, 0, 0 ]
```

Note that this subgroup of $O_7(3).2 \cong SO(7,3)$ is the orthogonal group $GO_6^-(3)$.

Now we deal with the double covers of $U_4(3).(2^2)_{133}$. The constructions of the character tables are completely analogous to those in the case of $U_4(3).(2^2)_{122}$.

```
                         ───── Example ─────
  gap> tbls:= List( [ "1", "3", "3'" ],
  >        i -> CharacterTable( Concatenation( "2.U4(3).2_", i ) ) );;
  gap> isos:= List( [ "1", "3", "3'" ], i ->
  >        CharacterTable( Concatenation( "Isoclinic(2.U4(3).2_", i, ")" ) ) ) );;
  gap> inputs:= [
  > [ tbls[1], tbls[2], tbls[3], "2.U4(3).(2^2)_{133}" ],
  > [ isos[1], tbls[2], tbls[3], "2.U4(3).(2^2)_{1*33}" ],
  > [ tbls[1], isos[2], tbls[3], "2.U4(3).(2^2)_{13*3}" ],
  > [ isos[1], isos[2], tbls[3], "2.U4(3).(2^2)_{1*3*3}" ],
  > [ tbls[1], isos[2], isos[3], "2.U4(3).(2^2)_{13*3*}" ],
  > [ isos[1], isos[2], isos[3], "2.U4(3).(2^2)_{1*3*3*}" ] ];;
  gap> tblG:= CharacterTable( "2.U4(3)" );;
  gap> result:= [];;
  gap> for input in inputs do
  >        tblsG2:= input{ [ 1 .. 3 ] };
  >        lib:= CharacterTable( input[4] );
  >        poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, input[4], lib );
  >        ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
  >        Append( result, RepresentativesCharacterTables( poss ) );
  >     od;
  #I  excluded cand. 2 (out of 4) for 2.U4(3).(2^2)_{1*33} by
  3-mod. table
  #I  excluded cand. 3 (out of 4) for 2.U4(3).(2^2)_{1*33} by
  3-mod. table
  #I  excluded cand. 2 (out of 4) for 2.U4(3).(2^2)_{13*3} by
  3-mod. table
  #I  excluded cand. 3 (out of 4) for 2.U4(3).(2^2)_{13*3} by
  3-mod. table
  #I  excluded cand. 2 (out of 4) for 2.U4(3).(2^2)_{1*3*3*} by
  3-mod. table
  #I  excluded cand. 3 (out of 4) for 2.U4(3).(2^2)_{1*3*3*} by
  3-mod. table
  gap> result:= List( result, x -> x.table );
  [ CharacterTable( "new2.U4(3).(2^2)_{133}" ),
    CharacterTable( "new2.U4(3).(2^2)_{1*33}" ),
    CharacterTable( "new2.U4(3).(2^2)_{13*3}" ),
    CharacterTable( "new2.U4(3).(2^2)_{1*3*3}" ),
    CharacterTable( "new2.U4(3).(2^2)_{13*3*}" ),
    CharacterTable( "new2.U4(3).(2^2)_{1*3*3*}" ) ]
  gap> List( result, NrConjugacyClasses );
  [ 69, 72, 72, 69, 69, 72 ]
  gap> t:= result[1];;
```

```
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 4, 4, 5 ]
gap> t:= result[2];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 3, 3, 6 ]
```

2.6.7 The Character Tables of Groups of the Type $4_1.L_3(4).2^2$ (October 2011)

The situation with $4_1.L_3(4).2^2$ is analogous to that with $6.L_3(4).2^2$, see Section 2.6.5.

——————————— Example ———————————
```
gap> tbls:= List( [ "1", "2", "3" ],
>     i -> CharacterTable( Concatenation( "4_1.L3(4).2_", i ) ) );
[ CharacterTable( "4_1.L3(4).2_1" ), CharacterTable( "4_1.L3(4).2_2" )
    , CharacterTable( "4_1.L3(4).2_3" ) ]
gap> isos:= List( [ "1", "2", "3" ],
>     i -> CharacterTable( Concatenation( "4_1.L3(4).2_", i, "*" ) ) );
[ CharacterTable( "Isoclinic(4_1.L3(4).2_1)" ),
  CharacterTable( "Isoclinic(4_1.L3(4).2_2)" ),
  CharacterTable( "4_1.L3(4).2_3*" ) ]
```

Note that the group $4_1.L_3(4).2_3$ has a centre of order four, so one cannot construct the isoclinic variant by calling the one argument version of `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**).

——————————— Example ———————————
```
gap> List( tbls, ClassPositionsOfCentre );
[ [ 1, 3 ], [ 1, 3 ], [ 1, 2, 3, 4 ] ]
gap> IsRecord( TransformingPermutationsCharacterTables( tbls[3],
>       CharacterTableIsoclinic( tbls[3] ) ) );
true
```

Again, we get eight different character tables, in two orbits.

——————————— Example ———————————
```
gap> inputs:= [
> [ tbls[1], tbls[2], tbls[3], "4_1.L3(4).(2^2)_{123}" ],
> [ isos[1], tbls[2], tbls[3], "4_1.L3(4).(2^2)_{1*23}" ],
> [ tbls[1], isos[2], tbls[3], "4_1.L3(4).(2^2)_{12*3}" ],
> [ isos[1], isos[2], tbls[3], "4_1.L3(4).(2^2)_{1*2*3}" ],
> [ tbls[1], tbls[2], isos[3], "4_1.L3(4).(2^2)_{123*}" ],
> [ isos[1], tbls[2], isos[3], "4_1.L3(4).(2^2)_{1*23*}" ],
> [ tbls[1], isos[2], isos[3], "4_1.L3(4).(2^2)_{12*3*}" ],
> [ isos[1], isos[2], isos[3], "4_1.L3(4).(2^2)_{1*2*3*}" ] ];;
gap> tblG:= CharacterTable( "4_1.L3(4)" );;
```

```
gap> result:= [];;
gap> for input in inputs do
>       tblsG2:= input{ [ 1 .. 3 ] };
>       lib:= CharacterTable( input[4] );
>       poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, input[4], lib );
>       ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
>       Append( result, RepresentativesCharacterTables( poss ) );
>     od;
#I  excluded cand. 2 (out of 8) for 4_1.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_1.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_1.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_1.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_1.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_1.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 4_1.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_1.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_1.L3(4).(2^2)_{1*23} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_1.L3(4).(2^2)_{1*23} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_1.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_1.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 4_1.L3(4).(2^2)_{12*3} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_1.L3(4).(2^2)_{12*3} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_1.L3(4).(2^2)_{12*3} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_1.L3(4).(2^2)_{12*3} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_1.L3(4).(2^2)_{12*3} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_1.L3(4).(2^2)_{12*3} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 4_1.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_1.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_1.L3(4).(2^2)_{1*2*3} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_1.L3(4).(2^2)_{1*2*3} by
7-mod. table
```

```
#I  excluded cand. 6 (out of 8) for 4_1.L3(4).(2^2)_{1*2*3} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_1.L3(4).(2^2)_{1*2*3} by
5-mod. table
gap> result:= List( result, x -> x.table );
[ CharacterTable( "new4_1.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{1*23}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{1*2*3}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{1*23*}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{12*3*}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{1*2*3*}" ) ]
gap> List( result, NrConjugacyClasses );
[ 48, 48, 48, 48, 42, 42, 42, 42 ]
gap> t:= result[1];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 3, 2, 4 ]
gap> t:= result[5];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 7, 6, 8 ]
```

Note that only two out of the eight tables of the type $2.L_3(4).2^2$ occur as factors of the eight tables.

```
─────────────────── Example ───────────────────
gap> facts:= [ CharacterTable( "2.L3(4).(2^2)_{123}" ),
>              CharacterTable( "2.L3(4).(2^2)_{123*}" ) ];;
gap> factresults:= List( result, t -> t / ClassPositionsOfCentre( t ) );;
gap> List( factresults, t -> PositionProperty( facts, f ->
>          IsRecord( TransformingPermutationsCharacterTables( t, f ) ) ) );
[ 1, 1, 1, 1, 2, 2, 2, 2 ]
```

This is not surprising; note that for $1 \leq i \leq 2$, the two isoclinic variants of $4_1.L_3(4).2_i$ have isomorphic factor groups of the type $2.L_3(4).2_i$. (For $i = 3$, this is not the case.)

```
─────────────────── Example ───────────────────
gap> test:= [ CharacterTable( "4_1.L3(4).2_1" ),
>             CharacterTable( "4_1.L3(4).2_1*" ) ];;
gap> List( test, ClassPositionsOfCentre );
[ [ 1, 3 ], [ 1, 3 ] ]
gap> fact:= List( test, t -> t / ClassPositionsOfCentre( t ) );;
gap> IsRecord( TransformingPermutationsCharacterTables( fact[1], fact[2] ) );
true
gap> test:= [ CharacterTable( "4_1.L3(4).2_2" ),
>             CharacterTable( "4_1.L3(4).2_2*" ) ];;
```

```
gap> List( test, ClassPositionsOfCentre );
[ [ 1, 3 ], [ 1, 3 ] ]
gap> fact:= List( test, t -> t / ClassPositionsOfCentre( t ) );;
gap> IsRecord( TransformingPermutationsCharacterTables( fact[1], fact[2] ) );
true
```

Now we try the second approach and compare the results. By the abovementioned asymmetry, it is clear that the tables are not uniquely determined by the input data.

```
                              Example
gap> names:= [ "L3(4).(2^2)_{123}", "L3(4).(2^2)_{1*23}",
>               "L3(4).(2^2)_{12*3}", "L3(4).(2^2)_{1*2*3}" ];;
gap> inputs1:= List( names, nam -> [ "4_1.L3(4).2_3", "2.L3(4).2_3",
>       Concatenation( "2.", nam ), Concatenation( "4_1.", nam ) ] );;
gap> names:= List( names, nam -> ReplacedString( nam, "3}", "3*}" ) );;
gap> inputs2:= List( names, nam -> [ "4_1.L3(4).2_3*", "2.L3(4).2_3*",
>       Concatenation( "2.", nam ), Concatenation( "4_1.", nam ) ] );;
gap> inputs:= Concatenation( inputs1, inputs2 );
[ [ "4_1.L3(4).2_3", "2.L3(4).2_3", "2.L3(4).(2^2)_{123}",
      "4_1.L3(4).(2^2)_{123}" ],
  [ "4_1.L3(4).2_3", "2.L3(4).2_3", "2.L3(4).(2^2)_{1*23}",
      "4_1.L3(4).(2^2)_{1*23}" ],
  [ "4_1.L3(4).2_3", "2.L3(4).2_3", "2.L3(4).(2^2)_{12*3}",
      "4_1.L3(4).(2^2)_{12*3}" ],
  [ "4_1.L3(4).2_3", "2.L3(4).2_3", "2.L3(4).(2^2)_{1*2*3}",
      "4_1.L3(4).(2^2)_{1*2*3}" ],
  [ "4_1.L3(4).2_3*", "2.L3(4).2_3*", "2.L3(4).(2^2)_{123*}",
      "4_1.L3(4).(2^2)_{123*}" ],
  [ "4_1.L3(4).2_3*", "2.L3(4).2_3*", "2.L3(4).(2^2)_{1*23*}",
      "4_1.L3(4).(2^2)_{1*23*}" ],
  [ "4_1.L3(4).2_3*", "2.L3(4).2_3*", "2.L3(4).(2^2)_{12*3*}",
      "4_1.L3(4).(2^2)_{12*3*}" ],
  [ "4_1.L3(4).2_3*", "2.L3(4).2_3*", "2.L3(4).(2^2)_{1*2*3*}",
      "4_1.L3(4).(2^2)_{1*2*3*}" ] ]
gap> result2:= [];;
gap> for  input in inputs do
>       tblMG := CharacterTable( input[1] );
>       tblG  := CharacterTable( input[2] );
>       tblGA := CharacterTable( input[3] );
>       name  := Concatenation( "new", input[4] );
>       lib   := CharacterTable( input[4] );
>       poss:= ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );
>       Append( result2, poss );
>    od;
#E  4 possibilities for new4_1.L3(4).(2^2)_{123}
#E  no solution for new4_1.L3(4).(2^2)_{1*23}
#E  no solution for new4_1.L3(4).(2^2)_{12*3}
#E  no solution for new4_1.L3(4).(2^2)_{1*2*3}
#E  4 possibilities for new4_1.L3(4).(2^2)_{123*}
#E  no solution for new4_1.L3(4).(2^2)_{1*23*}
#E  no solution for new4_1.L3(4).(2^2)_{12*3*}
#E  no solution for new4_1.L3(4).(2^2)_{1*2*3*}
gap> Length( result2 );
```

```
8
gap> result2:= List( result2, x -> x.table );
[ CharacterTable( "new4_1.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new4_1.L3(4).(2^2)_{123*}" ) ]
gap> List( result, t1 -> PositionsProperty( result2, t2 -> IsRecord(
>        TransformingPermutationsCharacterTables( t1, t2 ) ) ) );
[ [ 1 ], [ 4 ], [ 3 ], [ 2 ], [ 7 ], [ 6 ], [ 5 ], [ 8 ] ]
```

At the moment, I do not know interesting groups that contain one of the $4_1.L_3(4).2^2$ type groups and whose character tables are available.

2.6.8 The Character Tables of Groups of the Type $4_2.L_3(4).2^2$ (October 2011)

The situation with $4_2.L_3(4).2^2$ is analogous to that with $6.L_3(4).2^2$, see Section 2.6.5.

```
———————————————————————— Example ————————————————————————
gap> tbls:= List( [ "1", "2", "3" ],
>        i -> CharacterTable( Concatenation( "4_2.L3(4).2_", i ) ) );
[ CharacterTable( "4_2.L3(4).2_1" ), CharacterTable( "4_2.L3(4).2_2" )
    , CharacterTable( "4_2.L3(4).2_3" ) ]
gap> isos:= List( [ "1", "2", "3" ],
>        i -> CharacterTable( Concatenation( "4_2.L3(4).2_", i, "*" ) ) );
[ CharacterTable( "Isoclinic(4_2.L3(4).2_1)" ),
  CharacterTable( "4_2.L3(4).2_2*" ),
  CharacterTable( "Isoclinic(4_2.L3(4).2_3)" ) ]
```

Note that the group $4_1.L_3(4).2_2$ has a centre of order four, so one cannot construct the isoclinic variant not by calling the one argument version of `CharacterTableIsoclinic` (**Reference: CharacterTableIsoclinic**).

```
———————————————————————— Example ————————————————————————
gap> List( tbls, ClassPositionsOfCentre );
[ [ 1, 3 ], [ 1, 2, 3, 4 ], [ 1, 3 ] ]
gap> IsRecord( TransformingPermutationsCharacterTables( tbls[2],
>        CharacterTableIsoclinic( tbls[2] ) ) );
true
```

Again, we get eight different character tables, in two orbits.

```
———————————————————————— Example ————————————————————————
gap> inputs:= [
> [ tbls[1], tbls[2], tbls[3], "4_2.L3(4).(2^2)_{123}" ],
> [ isos[1], tbls[2], tbls[3], "4_2.L3(4).(2^2)_{1*23}" ],
> [ tbls[1], isos[2], tbls[3], "4_2.L3(4).(2^2)_{12*3}" ],
> [ tbls[1], tbls[2], isos[3], "4_2.L3(4).(2^2)_{123*}" ],
> [ isos[1], isos[2], tbls[3], "4_2.L3(4).(2^2)_{1*2*3}" ],
> [ isos[1], tbls[2], isos[3], "4_2.L3(4).(2^2)_{1*23*}" ],
```

```
>       [ tbls[1], isos[2], isos[3], "4_2.L3(4).(2^2)_{12*3*}" ],
>       [ isos[1], isos[2], isos[3], "4_2.L3(4).(2^2)_{1*2*3*}" ] ];;
gap> tblG:= CharacterTable( "4_2.L3(4)" );;
gap> result:= [];;
gap> for input in inputs do
>       tblsG2:= input{ [ 1 .. 3 ] };
>       lib:= CharacterTable( input[4] );
>       poss:= ConstructOrdinaryGV4Table( tblG, tblsG2, input[4], lib );
>       ConstructModularGV4Tables( tblG, tblsG2, poss, lib );
>       Append( result, RepresentativesCharacterTables( poss ) );
>    od;
#I  excluded cand. 2 (out of 8) for 4_2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_2.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_2.L3(4).(2^2)_{123} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_2.L3(4).(2^2)_{123} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 4_2.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_2.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_2.L3(4).(2^2)_{1*23} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_2.L3(4).(2^2)_{1*23} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_2.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_2.L3(4).(2^2)_{1*23} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 4_2.L3(4).(2^2)_{123*} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_2.L3(4).(2^2)_{123*} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_2.L3(4).(2^2)_{123*} by
7-mod. table
#I  excluded cand. 5 (out of 8) for 4_2.L3(4).(2^2)_{123*} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_2.L3(4).(2^2)_{123*} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_2.L3(4).(2^2)_{123*} by
5-mod. table
#I  excluded cand. 2 (out of 8) for 4_2.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 3 (out of 8) for 4_2.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 4 (out of 8) for 4_2.L3(4).(2^2)_{1*23*} by
```

```
    7-mod. table
#I  excluded cand. 5 (out of 8) for 4_2.L3(4).(2^2)_{1*23*} by
7-mod. table
#I  excluded cand. 6 (out of 8) for 4_2.L3(4).(2^2)_{1*23*} by
5-mod. table
#I  excluded cand. 7 (out of 8) for 4_2.L3(4).(2^2)_{1*23*} by
5-mod. table
gap> result:= List( result, x -> x.table );
[ CharacterTable( "new4_2.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{1*23}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{123*}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{1*2*3}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{1*23*}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{12*3*}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{1*2*3*}" ) ]
gap> List( result, NrConjugacyClasses );
[ 50, 50, 44, 50, 44, 50, 44, 44 ]
gap> t:= result[1];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 4, 2, 6 ]
gap> t:= result[3];;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( t ),
>            x -> Sum( SizesConjugacyClasses( t ){ x } ) = Size( t ) / 2 );;
gap> iso:= List( nsg, x -> CharacterTableIsoclinic( t, x ) );;
gap> List( iso, x -> PositionProperty( result, y ->
>            TransformingPermutationsCharacterTables( x, y ) <> fail ) );
[ 7, 5, 8 ]
```

Note that only two out of the eight tables of the type $2.L_3(4).2^2$ occur as factors of the eight tables.

```
 _____ Example _____
gap> facts:= [ CharacterTable( "2.L3(4).(2^2)_{123}" ),
>              CharacterTable( "2.L3(4).(2^2)_{12*3}" ) ];;
gap> factresults:= List( result, t -> t / ClassPositionsOfCentre( t ) );;
gap> List( factresults, t -> PositionProperty( facts, f ->
>            IsRecord( TransformingPermutationsCharacterTables( t, f ) ) ) );
[ 1, 1, 2, 1, 2, 1, 2, 2 ]
```

This is not surprising; note that for $i \in \{1,3\}$, the two isoclinic variants of $4_1.L_3(4).2_i$ have isomorphic factor groups of the type $2.L_3(4).2_i$. (For $i = 2$, this is not the case.)

```
 _____ Example _____
gap> test:= [ CharacterTable( "4_2.L3(4).2_1" ),
>             CharacterTable( "4_2.L3(4).2_1*" ) ];;
gap> List( test, ClassPositionsOfCentre );
[ [ 1, 3 ], [ 1, 3 ] ]
gap> fact:= List( test, t -> t / ClassPositionsOfCentre( t ) );;
gap> IsRecord( TransformingPermutationsCharacterTables( fact[1], fact[2] ) );
```

```
    true
    gap> test:= [ CharacterTable( "4_2.L3(4).2_3" ),
    >              CharacterTable( "4_2.L3(4).2_3*" ) ];;
    gap> List( test, ClassPositionsOfCentre );
    [ [ 1, 3 ], [ 1, 3 ] ]
    gap> fact:= List( test, t -> t / ClassPositionsOfCentre( t ) );;
    gap> IsRecord( TransformingPermutationsCharacterTables( fact[1], fact[2] ) );
    true
```

Now we try the second approach and compare the results. By the abovementioned asymmetry, it is clear that the tables are not uniquely determined by the input data.

```
                                  —— Example ——
  gap> names:= [ "L3(4).(2^2)_{123}", "L3(4).(2^2)_{1*23}",
  >              "L3(4).(2^2)_{123*}", "L3(4).(2^2)_{1*23*}" ];;
  gap> inputs1:= List( names, nam -> [ "4_2.L3(4).2_2", "2.L3(4).2_2",
  >       Concatenation( "2.", nam ), Concatenation( "4_2.", nam ) ] );;
  gap> names:= List( names, nam -> ReplacedString( nam, "23", "2*3" ) );;
  gap> inputs2:= List( names, nam -> [ "4_2.L3(4).2_2*", "2.L3(4).2_2*",
  >       Concatenation( "2.", nam ), Concatenation( "4_2.", nam ) ] );;
  gap> inputs:= Concatenation( inputs1, inputs2 );
  [ [ "4_2.L3(4).2_2", "2.L3(4).2_2", "2.L3(4).(2^2)_{123}",
        "4_2.L3(4).(2^2)_{123}" ],
    [ "4_2.L3(4).2_2", "2.L3(4).2_2", "2.L3(4).(2^2)_{1*23}",
        "4_2.L3(4).(2^2)_{1*23}" ],
    [ "4_2.L3(4).2_2", "2.L3(4).2_2", "2.L3(4).(2^2)_{123*}",
        "4_2.L3(4).(2^2)_{123*}" ],
    [ "4_2.L3(4).2_2", "2.L3(4).2_2", "2.L3(4).(2^2)_{1*23*}",
        "4_2.L3(4).(2^2)_{1*23*}" ],
    [ "4_2.L3(4).2_2*", "2.L3(4).2_2*", "2.L3(4).(2^2)_{12*3}",
        "4_2.L3(4).(2^2)_{12*3}" ],
    [ "4_2.L3(4).2_2*", "2.L3(4).2_2*", "2.L3(4).(2^2)_{1*2*3}",
        "4_2.L3(4).(2^2)_{1*2*3}" ],
    [ "4_2.L3(4).2_2*", "2.L3(4).2_2*", "2.L3(4).(2^2)_{12*3*}",
        "4_2.L3(4).(2^2)_{12*3*}" ],
    [ "4_2.L3(4).2_2*", "2.L3(4).2_2*", "2.L3(4).(2^2)_{1*2*3*}",
        "4_2.L3(4).(2^2)_{1*2*3*}" ] ]
  gap> result2:= [];;
  gap> for  input in inputs do
  >       tblMG := CharacterTable( input[1] );
  >       tblG  := CharacterTable( input[2] );
  >       tblGA := CharacterTable( input[3] );
  >       name  := Concatenation( "new", input[4] );
  >       lib   := CharacterTable( input[4] );
  >       poss:= ConstructOrdinaryMGATable( tblMG, tblG, tblGA, name, lib );
  >       Append( result2, poss );
  >    od;
  #E  4 possibilities for new4_2.L3(4).(2^2)_{123}
  #E  no solution for new4_2.L3(4).(2^2)_{1*23}
  #E  no solution for new4_2.L3(4).(2^2)_{123*}
  #E  no solution for new4_2.L3(4).(2^2)_{1*23*}
  #E  4 possibilities for new4_2.L3(4).(2^2)_{12*3}
  #E  no solution for new4_2.L3(4).(2^2)_{1*2*3}
```

```
#E  no solution for new4_2.L3(4).(2^2)_{12*3*}
#E  no solution for new4_2.L3(4).(2^2)_{1*2*3*}
gap> Length( result2 );
8
gap> result2:= List( result2, x -> x.table );
[ CharacterTable( "new4_2.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{123}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{12*3}" ),
  CharacterTable( "new4_2.L3(4).(2^2)_{12*3}" ) ]
gap> List( result, t1 -> PositionsProperty( result2, t2 -> IsRecord(
>       TransformingPermutationsCharacterTables( t1, t2 ) ) ) );
[ [ 1 ], [ 4 ], [ 7 ], [ 3 ], [ 6 ], [ 2 ], [ 5 ], [ 8 ] ]
```

The group $ON.2$ contains a maximal subgroup M of the type $4_2.L_3(4).2^2$, see [CCN$^+$85, p. 132]. Only the third result table admits a class fusion into $ON.2$. This shows the existence of groups for the tables from the second orbit.

```
 ──────────── Example ────────────
 gap> on2:= CharacterTable( "ON.2" );;
 gap> fus:= List( result, x -> PossibleClassFusions( x, on2 ) );;
 gap> List( fus, Length );
 [ 0, 0, 16, 0, 0, 0, 0, 0 ]
```

2.6.9 The Character Table of Aut($L_2(81)$)

The group $\mathrm{Aut}(L_2(81)) \cong L_2(81).(2 \times 4)$ has the structure $G.2^2$ where $G = L_2(81).2_1$. Here we get two triples of possible actions on the tables of the groups $G.2_i$, and one possible character table for each triple.

```
 ──────────── Example ────────────
 gap> input:= [ "L2(81).2_1", "L2(81).4_1", "L2(81).4_2", "L2(81).2^2",
 >                                              "L2(81).(2x4)" ];;
 gap> tblG   := CharacterTable( input[1] );;
 gap> tblsG2 := List( input{ [ 2 .. 4 ] }, CharacterTable );;
 gap> name   := Concatenation( "new", input[5] );;
 gap> lib    := CharacterTable( input[5] );;
 gap> poss   := ConstructOrdinaryGV4Table( tblG, tblsG2, name, lib );;
 #I  newL2(81).(2x4): 2 equivalence classes
 gap> reps:= RepresentativesCharacterTables( poss );;
 gap> Length( reps );
 2
```

Due to the different underlying actions, the power maps of the two candidate tables differ.

```
 ──────────── Example ────────────
 gap> ord:= OrdersClassRepresentatives( reps[1].table );;
 gap> ord = OrdersClassRepresentatives( reps[2].table );
 true
 gap> pos:= Position( ord, 80 );
```

```
33
gap> PowerMap( reps[1].table, 3 )[ pos ];
34
gap> PowerMap( reps[2].table, 3 )[ pos ];
33
```

Aut($L_2(81)$) can be generated by PGL$(2,81) = L_2(81).2_2$ and the Frobenius automorphism of order four that is defined on GL$(2,81)$ as the map that cubes the matrix entries. The elements of order 80 in Aut($L_2(81)$) are conjugates of diagonal matrices modulo scalar matrices, which are mapped to their third powers by the Frobenius homomorphism. So the third power map of Aut($L_2(81)$) fixes the classes of elements of order 80. In other words, the second of the two tables is the right one.

```
───────────────── Example ─────────────────
gap> trans:= TransformingPermutationsCharacterTables( reps[2].table, lib );;
gap> IsRecord( trans );
true
gap> List( reps[2].G2fusGV4, x -> OnTuples( x, trans.columns ) )
> = List( tblsG2, x -> GetFusionMap( x, lib ) );
true
gap> ConstructModularGV4Tables( tblG, tblsG2, [ reps[2] ], lib );;
#I  not all input tables for L2(81).(2x4) mod 41 available
```

2.6.10 The Character Table of $O_8^+(3).2_{111}^2$

The construction of the character table of the group $O_8^+(3).2_{111}^2$ is not as straightforward as the constructions shown in Section 2.6.2. Here we get 26 triples of actions on the tables of the three subgroups $G.2_i$ of index two, but only one of them leads to candidates for the desired character table. Specifically, we get 64 such candidates, in two equivalence classes w.r.t. permutation equivalence.

```
───────────────── Example ─────────────────
gap> input:= [ "O8+(3)", "O8+(3).2_1",  "O8+(3).2_1'", "O8+(3).2_1''",
>                                       "O8+(3).(2^2)_{111}" ];;
gap> tblG   := CharacterTable( input[1] );;
gap> tblsG2 := List( input{ [ 2 .. 4 ] }, CharacterTable );;
gap> name   := Concatenation( "new", input[5] );;
gap> lib    := CharacterTable( input[5] );;
gap> poss   := ConstructOrdinaryGV4Table( tblG, tblsG2, name, lib );;
#I  newO8+(3).(2^2)_{111}: 2 equivalence classes
gap> Length( poss );
64
gap> reps:= RepresentativesCharacterTables( poss );;
gap> Length( reps );
2
```

The two candidate tables differ only in four irreducible characters involving irrationalities on the classes of element order 28. All three subgroups $G.2_i$ contain elements of order 28 that do not lie in the simple group G; these classes are roots of the same (unique) class of element order 7. The centralizer C of an order 7 element in $G.2^2$ has order $112 = 2^4 \cdot 7$, the intersection of C with G has the structure $2^2 \times 7$ since G contains three classes of cyclic subgroups of the order 14, and each of the intersections of C with one of the subgroups $G.2_i$ has the structure $2 \times 4 \times 7$, so the structure of C is $4^2 \times 7 \cong 4 \times 28$.

```
─────────────────── Example ───────────────────
gap> t:= reps[1].table;;
gap> ord7:= Filtered( [ 1 .. NrConjugacyClasses( t ) ],
>              i -> OrdersClassRepresentatives( t )[i] = 7 );
[ 37 ]
gap> SizesCentralizers( t ){ ord7 };
[ 112 ]
gap> ord28:= Filtered( [ 1 .. NrConjugacyClasses( t ) ],
>               i -> OrdersClassRepresentatives( t )[i] = 28 );
[ 112, 113, 114, 115, 161, 162, 163, 164, 210, 211, 212, 213 ]
gap> List( reps[1].G2fusGV4, x -> Intersection( ord28, x ) );
[ [ 112, 113, 114, 115 ], [ 161, 162, 163, 164 ],
  [ 210, 211, 212, 213 ] ]
gap> sub:= CharacterTable( "Cyclic", 28 ) * CharacterTable( "Cyclic", 4 );;
gap> List( reps, x -> Length( PossibleClassFusions( sub, x.table ) ) );
[ 0, 96 ]
```

It turns out that only one of the two candidate tables admits a class fusion from the character table of C, thus we have determined the ordinary character table of $O_8^+(3).2_{111}^2$. It coincides with the table from the library.

```
─────────────────── Example ───────────────────
gap> trans:= TransformingPermutationsCharacterTables( reps[2].table, lib );;
gap> IsRecord( trans );
true
gap> List( reps[2].G2fusGV4, x -> OnTuples( x, trans.columns ) )
>  = List( tblsG2, x -> GetFusionMap( x, lib ) );
true
```

(If we do not believe the statement about the structure of C then we can check all 14 groups of order 112 that contain a central subgroup of order 7. A unique such group admits a class fusion into at least one of the two candidate tables.)

The wrong candidate for the ordinary table cannot be excluded via conditions that are forced by the construction of the p-modular tables of $O_8^+(3).2_{111}^2$. Thus we restrict the ordinary tables used for this construction to those candidates that are equivalent to the correct table.

```
─────────────────── Example ───────────────────
gap> poss:= Filtered( poss,
>      r -> TransformingPermutationsCharacterTables( r.table, lib )
>           <> fail );;
gap> ConstructModularGV4Tables( tblG, tblsG2, poss, lib );;
#I  not all input tables for O8+(3).(2^2)_{111} mod 3 available
```

So also the p-modular tables of $O_8^+(3).2_{111}^2$ can be computed this way, provided that the p-modular tables of the index 2 subgroups are available.

2.7 Examples for the Type $2^2.G$

We compute the character table of a group of the type $2^2.G$ from the character tables of the three factor groups of the type $2.G$, using the function PossibleCharacterTablesOfTypeV4G (**CTblLib: PossibleCharacterTablesOfTypeV4G**), see Section 2.3.4.

2.7.1 The Character Table of $2^2.Sz(8)$

The three central involutions in $2^2.Sz(8)$ are permuted cyclically by an outer automorphism α of order three. In order to apply `PossibleCharacterTablesOfTypeV4G` (**CTblLib: PossibleCharacterTablesOfTypeV4G**), we need the character table of the group $2.Sz(8)$ and the action on the classes of $Sz(8)$ that is induced by α.

The ordinary character table of $G = Sz(8)$ admits exactly five table automorphisms of order dividing 3. Each of these possibilities leads to exactly one possible character table of $2^2.G$, and the five tables are permutation equivalent. From this point of view, we need not know which of the table automorphisms are induced by outer *group* automorphisms of G.

```
                           Example
  gap> t:= CharacterTable( "Sz(8)" );;
  gap> 2t:= CharacterTable( "2.Sz(8)" );;
  gap> aut:= AutomorphismsOfTable( t );;
  gap> elms:= Set( Filtered( aut, x -> Order( x ) in [ 1, 3 ] ),
  >               SmallestGeneratorPerm );
  [ (), (9,10,11), (6,7,8), (6,7,8)(9,10,11), (6,7,8)(9,11,10) ]
  gap> poss:= List( elms,
  >       pi -> PossibleCharacterTablesOfTypeV4G( t, 2t, pi, "2^2.Sz(8)" ) );
  [ [ CharacterTable( "2^2.Sz(8)" ) ], [ CharacterTable( "2^2.Sz(8)" ) ]
      , [ CharacterTable( "2^2.Sz(8)" ) ],
    [ CharacterTable( "2^2.Sz(8)" ) ],
    [ CharacterTable( "2^2.Sz(8)" ) ] ]
  gap> reps:= RepresentativesCharacterTables( Concatenation( poss ) );
  [ CharacterTable( "2^2.Sz(8)" ) ]
```

The tables coincide with the one that is stored in the GAP library.

```
                           Example
  gap> IsRecord( TransformingPermutationsCharacterTables( reps[1],
  >       CharacterTable( "2^2.Sz(8)" ) ) );
  true
```

The computation of the p-modular character table of $2^2.G$ from the p-modular character table of $2.G$ and the three factor fusions from $2^2.G$ to $2.G$ is straightforward, as is stated in Section 2.3.4. The three fusions are stored on the tables returned by `PossibleCharacterTablesOfTypeV4G` (**CTblLib: PossibleCharacterTablesOfTypeV4G**).

```
                           Example
  gap> GetFusionMap( poss[1][1], 2t, "1" );
  [ 1, 1, 2, 2, 3, 4, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12,
    12, 13, 13, 14, 14, 15, 15, 16, 16, 17, 17, 18, 18, 19, 19 ]
  gap> GetFusionMap( poss[1][1], 2t, "2" );
  [ 1, 2, 1, 2, 3, 4, 5, 6, 7, 6, 7, 8, 9, 8, 9, 10, 11, 10, 11, 12,
    13, 12, 13, 14, 15, 14, 15, 16, 17, 16, 17, 18, 19, 18, 19 ]
  gap> GetFusionMap( poss[1][1], 2t, "3" );
  [ 1, 2, 2, 1, 3, 4, 5, 6, 7, 7, 6, 8, 9, 9, 8, 10, 11, 11, 10, 12,
    13, 13, 12, 14, 15, 15, 14, 16, 17, 17, 16, 18, 19, 19, 18 ]
```

The GAP library function `BrauerTableOfTypeV4G` (**CTblLib: BrauerTableOfTypeV4G**) can be used to derive Brauer tables of $2^2.G$. We have to compute the p-modular tables for prime divisors p of $|G|$, that is, for $p \in \{2, 5, 7, 13\}$.

```
                              ─ Example ─
  gap> PrimeDivisors( Size( t ) );
  [ 2, 5, 7, 13 ]
```

Clearly $p = 2$ is uninteresting from this point of view because the 2-modular table of $2^2.G$ can be identified with the 2-modular table of G.

For each of the five ordinary tables (corresponding to the five possible table automorphisms of G) constructed above, we get one candidate of a 5-modular table. However, these tables are *not* all equivalent. There are two equivalence classes, and one of the two possibilities is inconsistent in the sense that not all tensor products of irreducibles decompose into irreducibles.

```
                              ─ Example ─
  gap> cand:= List( poss, l -> BrauerTableOfTypeV4G( l[1], 2t mod 5,
  >       ConstructionInfoCharacterTable( l[1] )[3] ) );
  [ BrauerTable( "2^2.Sz(8)", 5 ), BrauerTable( "2^2.Sz(8)", 5 ),
    BrauerTable( "2^2.Sz(8)", 5 ), BrauerTable( "2^2.Sz(8)", 5 ),
    BrauerTable( "2^2.Sz(8)", 5 ) ]
  gap> Length( RepresentativesCharacterTables( cand ) );
  2
  gap> List( cand, CTblLib.Test.TensorDecomposition );
  [ false, true, false, true, true ]
  gap> Length( RepresentativesCharacterTables( cand{ [ 2, 4, 5 ] } ) );
  1
  gap> IsRecord( TransformingPermutationsCharacterTables( cand[2],
  >       CharacterTable( "2^2.Sz(8)" ) mod 5 ) );
  true
```

This implies that only those table automorphisms of G can be induced by an outer group automorphism that move the classes of element order 13.

The 7-modular table of $2^2.G$ is uniquely determined, independent of the choice of the table automorphism of G.

```
                              ─ Example ─
  gap> cand:= List( poss, l -> BrauerTableOfTypeV4G( l[1], 2t mod 7,
  >       ConstructionInfoCharacterTable( l[1] )[3] ) );
  [ BrauerTable( "2^2.Sz(8)", 7 ), BrauerTable( "2^2.Sz(8)", 7 ),
    BrauerTable( "2^2.Sz(8)", 7 ), BrauerTable( "2^2.Sz(8)", 7 ),
    BrauerTable( "2^2.Sz(8)", 7 ) ]
  gap> Length( RepresentativesCharacterTables( cand ) );
  1
  gap> IsRecord( TransformingPermutationsCharacterTables( cand[1],
  >       CharacterTable( "2^2.Sz(8)" ) mod 7 ) );
  true
```

We get two candidates for the 13-modular table of $2^2.G$, also if we consider only the three admissible table automorphisms.

```
                              ─ Example ─
  gap> elms:= elms{ [ 2, 4, 5 ] };
  [ (9,10,11), (6,7,8)(9,10,11), (6,7,8)(9,11,10) ]
  gap> poss:= poss{ [ 2, 4, 5 ] };;
  gap> cand:= List( poss, l -> BrauerTableOfTypeV4G( l[1], 2t mod 13,
```

```
>          ConstructionInfoCharacterTable( l[1] )[3] ) );
[ BrauerTable( "2^2.Sz(8)", 13 ), BrauerTable( "2^2.Sz(8)", 13 ),
  BrauerTable( "2^2.Sz(8)", 13 ) ]
gap> Length( RepresentativesCharacterTables( cand ) );
2
gap> List( cand, CTblLib.Test.TensorDecomposition );
[ true, true, true ]
```

The action of the outer automorphism of order three of *G* can be read off from the 2-modular table of *G*. Note that the ordinary and the 5-modular character table of *G* possess two independent table automorphisms of order three, whereas the group of table automorphisms of the 2-modular table has order three. (The reason is that the irrational values on the classes of the element orders 7 and 13 appear in the same irreducible 2-modular Brauer characters.)

```
―――――――――――――― Example ――――――――――――――
gap> mod2:= CharacterTable( "Sz(8)" ) mod 2;
BrauerTable( "Sz(8)", 2 )
gap> AutomorphismsOfTable( mod2 );
Group([ (3,4,5)(6,7,8) ])
gap> OrdersClassRepresentatives( mod2 );
[ 1, 5, 7, 7, 7, 13, 13, 13 ]
```

This means that the first candidate is ruled out; this determines the 13-modular character table of $2^2.G$.

```
―――――――――――――― Example ――――――――――――――
gap> Length( RepresentativesCharacterTables( cand{ [ 2, 3 ] } ) );
1
gap> IsRecord( TransformingPermutationsCharacterTables( cand[2],
>         CharacterTable( "2^2.Sz(8)" ) mod 13 ) );
true
```

2.7.2 **Atlas** Tables of the Type $2^2.G$ (September 2005)

Besides $2^2.Sz(8)$ (cf. Section 2.7.1), $2^2.O_8^+(3)$ (cf. Section 2.7.3), and certain central extensions of $L_3(4)$ (cf. Section 2.7.4), the following examples of central extensions of nearly simple **Atlas** groups *G* by a Klein four group occur.

```
―――――――――――――― Example ――――――――――――――
gap> listV4G:= [
>     [ "2^2.L3(4)",         "2.L3(4)",       "L3(4)"        ],
>     [ "2^2.L3(4).2_1",     "2.L3(4).2_1",   "L3(4).2_1"    ],
>     [ "(2^2x3).L3(4)",     "6.L3(4)",       "3.L3(4)"      ],
>     [ "(2^2x3).L3(4).2_1", "6.L3(4).2_1",   "3.L3(4).2_1"  ],
>     [ "2^2.O8+(2)",        "2.O8+(2)",      "O8+(2)"       ],
>     [ "2^2.U6(2)",         "2.U6(2)",       "U6(2)"        ],
>     [ "(2^2x3).U6(2)",     "6.U6(2)",       "3.U6(2)"      ],
>     [ "2^2.2E6(2)",        "2.2E6(2)",      "2E6(2)"       ],
>     [ "(2^2x3).2E6(2)",    "6.2E6(2)",      "3.2E6(2)"     ],
> ];;
```

(For the tables of $(2^2 \times 3).G$, with G one of $L_3(4)$, $U_6(2)$, or ${}^2E_6(2)$, we could alternatively use the tables of $2^2.G$ and $3.G$, and the construction described in Chapter 3.)

The function for computing the candidates for the ordinary character tables is similar to the one from Section 2.6.2.

```
                              ──── Example ────
 gap> ConstructOrdinaryV4GTable:= function( tblG, tbl2G, name, lib )
 >       local ord3, nam, poss, reps, trans;
 >
 >       # Compute the possible actions for the ordinary tables.
 >       ord3:= Set( Filtered( AutomorphismsOfTable( tblG ),
 >                             x -> Order( x ) = 3 ),
 >                 SmallestGeneratorPerm );
 >       if 1 < Length( ord3 ) then
 >         Print( "#I  ", name,
 >                ": the action of the automorphism is not unique" );
 >       fi;
 >       # Compute the possible ordinary tables for the given actions.
 >       nam:= Concatenation( "new", name );
 >       poss:= Concatenation( List( ord3, pi ->
 >             PossibleCharacterTablesOfTypeV4G( tblG, tbl2G, pi, nam ) ) );
 >       # Test the possibilities for permutation equivalence.
 >       reps:= RepresentativesCharacterTables( poss );
 >       if 1 < Length( reps ) then
 >         Print( "#I  ", name, ": ", Length( reps ),
 >                " equivalence classes\n" );
 >       elif Length( reps ) = 0 then
 >         Print( "#E  ", name, ": no solution\n" );
 >       else
 >         # Compare the computed table with the library table.
 >         if not IsCharacterTable( lib ) then
 >           Print( "#I  no library table for ", name, "\n" );
 >           PrintToLib( name, poss[1].table );
 >         else
 >           trans:= TransformingPermutationsCharacterTables( reps[1], lib );
 >           if not IsRecord( trans ) then
 >             Print( "#E  computed table and library table for ", name,
 >                    " differ\n" );
 >           fi;
 >         fi;
 >       fi;
 >       return poss;
 >     end;;
```

Concerning the Brauer tables, the same ambiguity problem may occur as in Section 2.6.2: Some candidates for the ordinary table may be excluded due to information provided by some p-modular table, see Section 2.7.1 for an easy example. Our strategy is analogous to the one used in Section 2.6.2.

```
                              ──── Example ────
 gap> ConstructModularV4GTables:= function( tblG, tbl2G, ordposs,
 >                                          ordlibtblV4G )
 >       local name, modposs, primes, checkordinary, i, p, tmodp, 2tmodp, aut,
 >             poss, modlib, trans, reps;
```

```
>
>         if not IsCharacterTable( ordlibtblV4G ) then
>           Print( "#I  no ordinary library table ...\n" );
>           return [];
>         fi;
>         name:= Identifier( ordlibtblV4G );
>         modposs:= [];
>         primes:= ShallowCopy( PrimeDivisors( Size( tblG ) ) );
>         ordposs:= ShallowCopy( ordposs );
>         checkordinary:= false;
>         for i in [ 1 .. Length( ordposs ) ] do
>           modposs[i]:= [];
>           for p in primes do
>             tmodp := tblG  mod p;
>             2tmodp:= tbl2G mod p;
>             if IsCharacterTable( tmodp ) and IsCharacterTable( 2tmodp ) then
>               aut:= ConstructionInfoCharacterTable( ordposs[i] )[3];
>               poss:= BrauerTableOfTypeV4G( ordposs[i], 2tmodp, aut );
>               if CTblLib.Test.TensorDecomposition( poss, false ) = false then
>                 Print( "#I  excluded cand. ", i, " (out of ",
>                         Length( ordposs ), ") for ", name, " by ", p,
>                         "-mod. table\n" );
>                 Unbind( ordposs[i] );
>                 Unbind( modposs[i] );
>                 checkordinary:= true;
>                 break;
>               fi;
>               Add( modposs[i], poss );
>             else
>               Print( "#I  not all input tables for ", name, " mod ", p,
>                       " available\n" );
>               primes:= Difference( primes, [ p ] );
>             fi;
>           od;
>           if IsBound( modposs[i] ) then
>             # Compare the computed Brauer tables with the library tables.
>             for poss in modposs[i] do
>               p:= UnderlyingCharacteristic( poss );
>               modlib:= ordlibtblV4G mod p;
>               if IsCharacterTable( modlib ) then
>                 trans:= TransformingPermutationsCharacterTables(
>                           poss, modlib );
>                 if not IsRecord( trans ) then
>                   Print( "#E  computed table and library table for ",
>                           name, " mod ", p, " differ\n" );
>                 fi;
>               else
>                 Print( "#I  no library table for ",
>                         name, " mod ", p, "\n" );
>                 PrintToLib( name, poss );
>               fi;
>             od;
```

```
>          fi;
>        od;
>        if checkordinary then
>          # Test whether the ordinary table is admissible.
>          ordposs:= Compacted( ordposs );
>          modposs:= Compacted( modposs );
>          reps:= RepresentativesCharacterTables( ordposs );
>          if 1 < Length( reps ) then
>            Print( "#I  ", name, ": ", Length( reps ),
>                   " equivalence classes (ord. table)\n" );
>          elif Length( reps ) = 0 then
>            Print( "#E  ", name, ": no solution (ord. table)\n" );
>          else
>            # Compare the computed table with the library table.
>            trans:= TransformingPermutationsCharacterTables( reps[1],
>                      ordlibtblV4G );
>            if not IsRecord( trans ) then
>              Print( "#E  computed table and library table for ", name,
>                     " differ\n" );
>            fi;
>          fi;
>        fi;
>        # Test the uniqueness of the Brauer tables.
>        for poss in TransposedMat( modposs ) do
>          reps:= RepresentativesCharacterTables( poss );
>          if Length( reps ) <> 1 then
>            Print( "#I  ", name, ": ", Length( reps ), " candidates for the ",
>                   UnderlyingCharacteristic( reps[1] ), "-modular table\n" );
>          fi;
>        od;
>        return rec( ordinary:= ordposs, modular:= modposs );
>      end;;
```

In our examples, the action of the outer automorphism of order three on the classes of *G* turns out to be uniquely determined by the table automorphisms of the character table of *G*.

```
———— Example ————
gap> for input in listV4G do
>      tblG  := CharacterTable( input[3] );
>      tbl2G := CharacterTable( input[2] );
>      lib   := CharacterTable( input[1] );
>      poss  := ConstructOrdinaryV4GTable( tblG, tbl2G, input[1], lib );
>      ConstructModularV4GTables( tblG, tbl2G, poss, lib );
>    od;
#I  excluded cand. 1 (out of 16) for 2^2.L3(4).2_1 by 7-mod. table
#I  excluded cand. 2 (out of 16) for 2^2.L3(4).2_1 by 7-mod. table
#I  excluded cand. 7 (out of 16) for 2^2.L3(4).2_1 by 7-mod. table
#I  excluded cand. 10 (out of 16) for 2^2.L3(4).2_1 by 7-mod. table
#I  excluded cand. 15 (out of 16) for 2^2.L3(4).2_1 by 7-mod. table
#I  excluded cand. 16 (out of 16) for 2^2.L3(4).2_1 by 7-mod. table
#I  excluded cand. 1 (out of 16) for (2^2x3).L3(4).2_1 by 7-mod. table
#I  excluded cand. 2 (out of 16) for (2^2x3).L3(4).2_1 by 7-mod. table
#I  excluded cand. 7 (out of 16) for (2^2x3).L3(4).2_1 by 7-mod. table
```

```
#I  excluded cand. 10 (out of 16) for (2^2x3).L3(4).2_1 by
7-mod. table
#I  excluded cand. 15 (out of 16) for (2^2x3).L3(4).2_1 by
7-mod. table
#I  excluded cand. 16 (out of 16) for (2^2x3).L3(4).2_1 by
7-mod. table
#I  not all input tables for 2^2.2E6(2) mod 2 available
#I  not all input tables for 2^2.2E6(2) mod 3 available
#I  not all input tables for 2^2.2E6(2) mod 5 available
#I  not all input tables for 2^2.2E6(2) mod 7 available
#I  not all input tables for (2^2x3).2E6(2) mod 2 available
#I  not all input tables for (2^2x3).2E6(2) mod 3 available
#I  not all input tables for (2^2x3).2E6(2) mod 5 available
#I  not all input tables for (2^2x3).2E6(2) mod 7 available
#I  not all input tables for (2^2x3).2E6(2) mod 11 available
#I  not all input tables for (2^2x3).2E6(2) mod 13 available
#I  not all input tables for (2^2x3).2E6(2) mod 17 available
#I  not all input tables for (2^2x3).2E6(2) mod 19 available
```

2.7.3 The Character Table of $2^2.O_8^+(3)$ (March 2009)

When one tries to construct the character table of the central extensions of $G = O_8^+(3)$ by a Klein four group, in the same way as in Section 2.7.2, one notices that the order three automorphism that relates the three central extensions of G by an involution is *not* uniquely determined.

```
─────────────── Example ───────────────
gap> entry:= [ "2^2.O8+(3)", "2.O8+(3)", "O8+(3)" ];;
gap> tblG:= CharacterTable( entry[3] );;
gap> aut:= AutomorphismsOfTable( tblG );;
gap> ord3:= Set( Filtered( aut, x -> Order( x ) = 3 ),
>               SmallestGeneratorPerm );;
gap> Length( ord3 );
4
```

However, the table candidates one gets from the four possible automorphisms turn out to be all equivalent, hence the character table of $2^2.O_8^+(3)$ can be constructed as follows.

```
─────────────── Example ───────────────
gap> poss:= [];;
gap> tbl2G:= CharacterTable( entry[2] );
CharacterTable( "2.O8+(3)" )
gap> for pi in ord3 do
>   Append( poss,
>           PossibleCharacterTablesOfTypeV4G( tblG, tbl2G, pi, entry[1] ) );
> od;
gap> Length( poss );
32
gap> poss:= RepresentativesCharacterTables( poss );;
gap> Length( poss );
1
```

The computed table coincides with the library table.

```
  ─────────────────────────── Example ───────────────────────────
  gap> lib:= CharacterTable( entry[1] );;
  gap> if TransformingPermutationsCharacterTables( poss[1], lib ) = fail then
  >        Print( "#E  differences for ", entry[1], "\n" );
  >     fi;
```

2.7.4 The Character Table of the Schur Cover of $L_3(4)$ (September 2005)

The Schur cover of $G = L_3(4)$ has the structure $(4^2 \times 3).L_3(4)$. Following [CCN$^+$85, p. 23], we regard the multiplier of G as

$$M = \langle a, b, c, d \mid [a, b] = [a, c] = [a, d] = [b, c] = [b, d] = [c, d] = a^4 = b^4 = c^4 = d^3 = abc \rangle,$$

and we will consider the automorphism α of $M.G$ that acts as $(a, b, c)(d)$ on M.

The subgroup lattice of the subgroup $\langle a, b, c \rangle = \langle a, b \rangle \cong 4^2$ of M looks as follows. (The subgroup in the centre of the picture is the Klein four group $\langle a^2, b^2, c^2 \rangle = \langle a^2, b^2 \rangle$.)
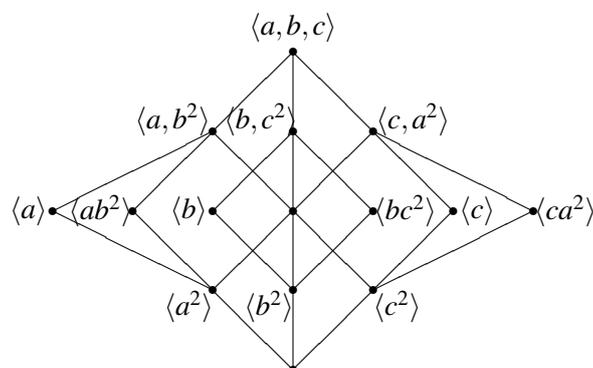
(The symmetry w.r.t. α would be reflected better in a three dimensional model, with $\langle a, b \rangle$, $\langle a^2, b^2 \rangle$, and the trivial subgroup on a vertical symmetry axis, and with the remaining subgroups on three circles such that α induces a rotation.)

The following is a 3D variant of the picture, which shows the symmetry of order three of the group 4×4.
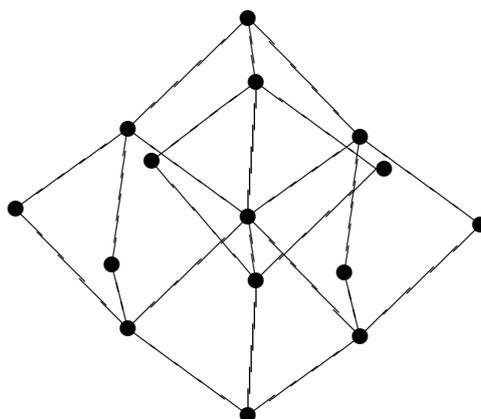
We have $(M/\langle a \rangle).G \cong (M/\langle b \rangle).G \cong (M/\langle c \rangle).G \cong 12_2.G$ and $(M/\langle ab^2 \rangle).G \cong (M/\langle bc^2 \rangle).G \cong (M/\langle ca^2 \rangle).G \cong 12_1.G$. This is because the action of $G.2_2$ fixes a, and swaps b and c; so b is inverted

modulo $\langle a \rangle$ but fixed modulo $\langle ab^2 \rangle$, and the normal subgroup of order four in $4_2.G.2_2$ is central but that in $4_1.G.2_2$ is not central.

The constructions of the character tables of $4^2.G$ and $(4^2 \times 3).G$ are essentially the same. We start with the table of $4^2.G$. It can be regarded as a central extension $H = V.2^2.G$ of $2^2.G$ by a Klein four group V. The three subgroups of order two in V are cyclicly permuted by the automorphism of $M/\langle d \rangle$ induced by α, so the three factors by these subgroups are isomorphic groups F, say, with the structure $(2 \times 4).G$.

The group F itself is a central extension of $2.G$ by a Klein four group, but in this case the three factor groups by the order two subgroups of the Klein four group are nonisomorphic groups, of the types $4_1.G$, $4_2.G$, and $2^2.G$, respectively. The GAP function `PossibleCharacterTablesOfTypeV4G` (**CTblLib: PossibleCharacterTablesOfTypeV4G**) can be used to construct the character table of F from the three factors. Note that in this case, no information about table automorphisms is required.

```
————————————————————— Example —————————————————————
gap> tblG:= CharacterTable( "2.L3(4)" );;
gap> tbls2G:= List( [ "4_1.L3(4)", "4_2.L3(4)", "2^2.L3(4)"],
>                    CharacterTable );;
gap> poss:= PossibleCharacterTablesOfTypeV4G( tblG, tbls2G, "(2x4).L3(4)" );;
gap> Length( poss );
2
gap> reps:= RepresentativesCharacterTables( poss );
[ CharacterTable( "(2x4).L3(4)" ) ]
gap> lib:= CharacterTable( "(2x4).L3(4)" );;
gap> IsRecord( TransformingPermutationsCharacterTables( reps[1], lib ) );
true
```

In the second step, we construct the table of $4^2.G$ from that of $(2 \times 4).G$ and the table automorphism of $2^2.G$ that is induced by α; it turns out that the group of table automorphisms of $2^2.G$ contains a unique subgroup of order three.

```
————————————————————— Example —————————————————————
gap> tblG:= tbls2G[3];
CharacterTable( "2^2.L3(4)" )
gap> tbl2G:= lib;
CharacterTable( "(2x4).L3(4)" )
gap> aut:= AutomorphismsOfTable( tblG );;
gap> ord3:= Set( Filtered( aut, x -> Order( x ) = 3 ),
>                 SmallestGeneratorPerm );
[ (2,3,4)(6,7,8)(10,11,12)(13,15,17)(14,16,18)(20,21,22)(24,25,26)(28,
    29,30)(32,33,34) ]
gap> pi:= ord3[1];;
gap> poss:= PossibleCharacterTablesOfTypeV4G( tblG, tbl2G, pi, "4^2.L3(4)" );;
gap> Length( poss );
4
gap> reps:= RepresentativesCharacterTables( poss );
[ CharacterTable( "4^2.L3(4)" ) ]
gap> lib:= CharacterTable( "4^2.L3(4)" );;
gap> IsRecord( TransformingPermutationsCharacterTables( reps[1], lib ) );
true
```

With the same approach, we compute the table of $(2 \times 12).G = 2^2.6.G$ from the tables of the three nonisomorphic factor groups $12_1.G$, $12_2.G$, and $(2^2 \times 3).G$, and we compute the table of $(4^2 \times 3).G = 2^2.(2^2 \times 3).G$ from the three tables of the factor groups $(2 \times 12).G$ and the action induced by α.

```
──────────────────────── Example ────────────────────────
gap> tblG:= CharacterTable( "6.L3(4)" );;
gap> tbls2G:= List( [ "12_1.L3(4)", "12_2.L3(4)", "(2^2x3).L3(4)"],
>                    CharacterTable );;
gap> poss:= PossibleCharacterTablesOfTypeV4G( tblG, tbls2G, "(2x12).L3(4)" );;
gap> Length( poss );
2
gap> reps:= RepresentativesCharacterTables( poss );
[ CharacterTable( "(2x12).L3(4)" ) ]
gap> lib:= CharacterTable( "(2x12).L3(4)" );;
gap> IsRecord( TransformingPermutationsCharacterTables( reps[1], lib ) );
true
gap> tblG:= CharacterTable( "(2^2x3).L3(4)" );
CharacterTable( "(2^2x3).L3(4)" )
gap> tbl2G:= CharacterTable( "(2x12).L3(4)" );
CharacterTable( "(2x12).L3(4)" )
gap> aut:= AutomorphismsOfTable( tblG );;
gap> ord3:= Set( Filtered( aut, x -> Order( x ) = 3 ),
>                 SmallestGeneratorPerm );
[ (2,7,8)(3,4,10)(6,11,12)(14,19,20)(15,16,22)(18,23,24)(26,27,28)(29,
    35,41)(30,37,43)(31,39,45)(32,36,42)(33,38,44)(34,40,46)(48,53,
    54)(49,50,56)(52,57,58)(60,65,66)(61,62,68)(64,69,70)(72,77,
    78)(73,74,80)(76,81,82)(84,89,90)(85,86,92)(88,93,94) ]
gap> pi:= ord3[1];;
gap> poss:= PossibleCharacterTablesOfTypeV4G( tblG, tbl2G, pi,
>                                             "(4^2x3).L3(4)" );;
gap> Length( poss );
4
gap> reps:= RepresentativesCharacterTables( poss );
[ CharacterTable( "(4^2x3).L3(4)" ) ]
gap> lib:= CharacterTable( "(4^2x3).L3(4)" );;
gap> IsRecord( TransformingPermutationsCharacterTables( reps[1], lib ) );
true
```

2.8 Examples of Extensions by p-singular Automorphisms

2.8.1 Some p-Modular Tables of Groups of the Type $M.G.A$

We show an alternative construction of p-modular tables of certain groups that have been met in Section 2.4.3. Each entry in the GAP list listMGA contains the Identifier (**Reference: Identifier for tables of marks**) values of character tables of groups of the types $M.G$, G, $G.A$, and $M.G.A$. For each entry with $|A| = p$, a prime integer, we fetch the p-modular table of G and the ordinary table of $G.A$, compute the action of $G.A$ on the p-regular classes of G, and then compute the p-modular table of $G.A$. Analogously, we compute the p-modular table of $M.G.A$ from the p-modular table of $M.G$ and the ordinary table of $M.G.A$.

```
──────────────────────── Example ────────────────────────
gap> for input in listMGA do
>        ordtblMG  := CharacterTable( input[1] );
>        ordtblG   := CharacterTable( input[2] );
>        ordtblGA  := CharacterTable( input[3] );
```

```
>        ordtblMGA := CharacterTable( input[4] );
>        p:= Size( ordtblGA ) / Size( ordtblG );
>        if IsPrimeInt( p ) then
>          modtblG:= ordtblG mod p;
>          if modtblG <> fail then
>            modtblGA := CharacterTableRegular( ordtblGA, p );
>            SetIrr( modtblGA, IBrOfExtensionBySingularAutomorphism( modtblG,
>                             ordtblGA ) );
>            modlibtblGA:= ordtblGA mod p;
>            if modlibtblGA = fail then
>              Print( "#E  ", p, "-modular table of '", Identifier( ordtblGA ),
>                    "' is missing\n" );
>            elif TransformingPermutationsCharacterTables( modtblGA,
>                    modlibtblGA ) = fail then
>              Print( "#E  computed table and library table for ", input[3],
>                    " mod ", p, " differ\n" );
>            fi;
>          fi;
>          modtblMG:= ordtblMG mod p;
>          if modtblMG <> fail then
>            modtblMGA := CharacterTableRegular( ordtblMGA, p );
>            SetIrr( modtblMGA, IBrOfExtensionBySingularAutomorphism( modtblMG,
>                             ordtblMGA ) );
>            modlibtblMGA:= ordtblMGA mod p;
>            if modlibtblMGA = fail then
>              Print( "#E  ", p, "-modular table of '", Identifier( ordtblMGA ),
>                    "' is missing\n" );
>            elif TransformingPermutationsCharacterTables( modtblMGA,
>                    modlibtblMGA ) = fail then
>              Print( "#E  computed table and library table for ", input[4],
>                    " mod ", p, " differ\n" );
>            fi;
>          fi;
>        fi;
>      od;
```

2.8.2 Some p-**Modular Tables of Groups of the Type** $G.S_3$

We show an alternative construction of 2- and 3-modular tables of certain groups that have been met in Section 2.5.2. Each entry in the GAP list `listGS3` contains the `Identifier` (**Reference: Identifier for tables of marks**) values of character tables of groups of the types G, $G.2$, $G.3$, and $G.S_3$. For each entry, we fetch the 2-modular table of G and the ordinary table of $G.2$, compute the action of $G.2$ on the 2-regular classes of G, and then compute the 2-modular table of $G.2$. Analogously, we compute the 3-modular table of $G.3$ from the 3-modular table of G and the ordinary table of $G.3$, and we compute the 2-modular table of $G.S_3$ from the 2-modular table of $G.3$ and the ordinary table of $G.S_3$.

```
─────────────── Example ───────────────
gap> for input in listGS3 do
>      modtblG:= CharacterTable( input[1] ) mod 2;
>      if modtblG <> fail then
>        ordtblG2 := CharacterTable( input[2] );
```

```
>         modtblG2 := CharacterTableRegular( ordtblG2, 2 );
>         SetIrr( modtblG2, IBrOfExtensionBySingularAutomorphism( modtblG,
>                         ordtblG2 ) );
>       modlibtblG2:= ordtblG2 mod 2;
>       if modlibtblG2 = fail then
>         Print( "#E  2-modular table of '", Identifier( ordtblG2 ),
>                 "' is missing\n" );
>       elif TransformingPermutationsCharacterTables( modtblG2,
>               modlibtblG2 ) = fail then
>         Print( "#E  computed table and library table for ", input[2],
>                 " mod 2 differ\n" );
>       fi;
>     fi;
>     modtblG:= CharacterTable( input[1] ) mod 3;
>     if modtblG <> fail then
>       ordtblG3 := CharacterTable( input[3] );
>       modtblG3 := CharacterTableRegular( ordtblG3, 3 );
>       SetIrr( modtblG3, IBrOfExtensionBySingularAutomorphism( modtblG,
>                         ordtblG3 ) );
>       modlibtblG3:= ordtblG3 mod 3;
>       if modlibtblG3 = fail then
>         Print( "#E  3-modular table of '", Identifier( ordtblG3 ),
>                 "' is missing\n" );
>       elif TransformingPermutationsCharacterTables( modtblG3,
>               modlibtblG3 ) = fail then
>         Print( "#E  computed table and library table for ", input[3],
>                 " mod 3 differ\n" );
>       fi;
>     fi;
>     modtblG3:= CharacterTable( input[3] ) mod 2;
>     if modtblG3 <> fail then
>       ordtblGS3 := CharacterTable( input[4] );
>       modtblGS3 := CharacterTableRegular( ordtblGS3, 2 );
>       SetIrr( modtblGS3, IBrOfExtensionBySingularAutomorphism( modtblG3,
>                           ordtblGS3 ) );
>       modlibtblGS3:= ordtblGS3 mod 2;
>       if modlibtblGS3 = fail then
>         Print( "#E  2-modular table of '", Identifier( ordtblGS3 ),
>                 "' is missing\n" );
>       elif TransformingPermutationsCharacterTables( modtblGS3,
>               modlibtblGS3 ) = fail then
>         Print( "#E  computed table and library table for ", input[4],
>                 " mod 2 differ\n" );
>       fi;
>     fi;
>   od;
```

2.8.3   2-**Modular Tables of Groups of the Type** $G.2^2$

We show an alternative construction of 2-modular tables of certain groups that have been met in Section 2.6.2. Each entry in the GAP list listGV4 contains the Identifier (**Reference: Identifier**

for tables of marks) values of character tables of groups of the types G, $G.2_1$, $G.2_2$, $G.2_3$, and $G.2^2$. For each entry, we fetch the 2-modular table of G and the ordinary tables of the groups $G.2_i$, and compute the 2-modular tables of $G.2_i$; Then we compute from this modular table and the ordinary table of $G.2^2$ the 2-modular table of $G.2^2$.

```
──────────────────── Example ────────────────────
gap> for input in listGV4 do
>       modtblG:= CharacterTable( input[1] ) mod 2;
>       if modtblG <> fail then
>         ordtblsG2:= List( input{ [ 2 .. 4 ] }, CharacterTable );
>         ordtblGV4:= CharacterTable( input[5] );
>         for tblG2 in ordtblsG2 do
>           modtblG2:= CharacterTableRegular( tblG2, 2 );
>           SetIrr( modtblG2, IBrOfExtensionBySingularAutomorphism( modtblG,
>                             tblG2 ) );
>           modlibtblG2:= tblG2 mod 2;
>           if modlibtblG2 = fail then
>             Print( "#E  2-modular table of '", Identifier( tblG2 ),
>                   "' is missing\n" );
>           elif TransformingPermutationsCharacterTables( modtblG2,
>                   modlibtblG2 ) = fail then
>             Print( "#E  computed table and library table for ",
>                   Identifier( tblG2 ), " mod 2 differ\n" );
>           fi;
>           modtblGV4:= CharacterTableRegular( ordtblGV4, 2 );
>           SetIrr( modtblGV4, IBrOfExtensionBySingularAutomorphism( modtblG2,
>                             ordtblGV4 ) );
>           modlibtblGV4:= ordtblGV4 mod 2;
>           if modlibtblGV4 = fail then
>             Print( "#E  2-modular table of '", Identifier( ordtblGV4 ),
>                   "' is missing\n" );
>           elif TransformingPermutationsCharacterTables( modtblGV4,
>                   ordtblGV4 mod 2 ) = fail then
>             Print( "#E  computed table and library table for ", input[5],
>                   " mod 2 differ\n" );
>           fi;
>         od;
>       fi;
>     od;
```

2.8.4 The 3-Modular Table of $U_3(8).3^2$

The only example of an Atlas group of the structure $G.3^3$ is $U_3(8).3^2$. Its 3-modular character table can be constructed from the known 3-modular character table of any of its index 3 subgroups, plus the action of $U_3(8).3^2$ on the classes of this subgroup.

```
──────────────────── Example ────────────────────
gap> ordtblG3:= CharacterTable( "U3(8).3^2" );;
gap> modlibtblG3:= ordtblG3 mod 3;
BrauerTable( "U3(8).3^2", 3 )
gap> for nam in [ "U3(8).3_1", "U3(8).3_2", "U3(8).3_3" ] do
>       modtblG:= CharacterTable( nam ) mod 3;
>       if modtblG = fail then
```

```
>          Error( "no 3-modular table of ", nam );
>        fi;
>        modtblG3:= CharacterTableRegular( ordtblG3, 3 );
>        SetIrr( modtblG3, IBrOfExtensionBySingularAutomorphism( modtblG,
>                              ordtblG3 ) );
>        if TransformingPermutationsCharacterTables( modtblG3,
>             modlibtblG3 ) = fail then
>          Print( "#E  computed table and library table for ",
>                 Identifier( ordtblG3 ), " mod 3 differ\n" );
>        fi;
>    od;
```

As expected, we get the same 3-modular table for any choice of the index 3 subgroup. Note that all 3-modular Brauer characters of $U_3(8).3^2$ lift to characteristic zero.

```
 ──────────────────────── Example ────────────────────────
 gap> rest:= RestrictedClassFunctions( Irr( ordtblG3 ), modlibtblG3 );;
 gap> IsSubset( rest, Irr( modlibtblG3 ) );
 true
```

2.9   Examples of Subdirect Products of Index Two

Typical examples of this construction are those maximal subgroups of alternating groups A_n that extend in the corresponding symmetric groups S_n to direct products of the structures $S_m \times S_{n-m}$, for $2 < m < n/2$. Also certain subgroups of these maximal subgroups that have this structure can be interesting, see Section 2.4.2.

2.9.1   Certain Dihedral Groups as Subdirect Products of Index Two

Also dihedral groups of order $2n$ with n divisible by at least two different primes have the required structure: Let $n = n_1 n_2$ with coprime n_1, n_2, and let the normal subgroups H_1, H_2 be cyclic subgroups of order n_1 and n_2, respectively, inside the cyclic subgroup of index two. Then the factors G/N_1, G/N_2 are themselves dihedral groups.

So an example (with $n_1 = 3$ and $n_2 = 5$) is the construction of the dihedral group D_{30} as a subdirect product of index two in the direct product $D_6 \times D_{10}$.

```
 ──────────────────────── Example ────────────────────────
 gap> tblh1:= CharacterTable( "C3" );;
 gap> tblg1:= CharacterTable( "S3" );;
 gap> StoreFusion( tblh1, PossibleClassFusions( tblh1, tblg1 )[1], tblg1 );
 gap> tblh2:= CharacterTable( "C5" );;
 gap> tblg2:= CharacterTable( "D10" );;
 gap> StoreFusion( tblh2, PossibleClassFusions( tblh2, tblg2 )[1], tblg2 );
 gap> subdir:= CharacterTableOfIndexTwoSubdirectProduct( tblh1, tblg1,
 >               tblh2, tblg2, "D30" );;
 gap> IsRecord( TransformingPermutationsCharacterTables( subdir.table,
 >               CharacterTable( "Dihedral", 30 ) ) );
 true
```

2.9.2 The Character Table of $(D_{10} \times HN).2 < M$ (June 2008)

The sporadic simple Monster group contains maximal subgroups with the structure $(D_{10} \times HN).2$ (see [CCN$^+$85, p. 234]), the factor group modulo D_{10} is the automorphism group $HN.2$ of HN, and the factor group modulo HN is the Frobenius group $5:4$ of order 20.

```
─────────── Example ───────────
gap> tblh1:= CharacterTable( "D10" );;
gap> tblg1:= CharacterTable( "5:4" );;
gap> tblh2:= CharacterTable( "HN" );;
gap> tblg2:= CharacterTable( "HN.2" );;
gap> subdir:= CharacterTableOfIndexTwoSubdirectProduct( tblh1, tblg1,
>                 tblh2, tblg2, "(D10xHN).2" );;
gap> IsRecord( TransformingPermutationsCharacterTables( subdir.table,
>                 CharacterTable( "(D10xHN).2" ) ) );
true
gap> m:= CharacterTable( "M" );;
gap> fus:= PossibleClassFusions( subdir.table, m );;
gap> Length( fus );
16
gap> Length( RepresentativesFusions( subdir.table, fus, m ) );
1
```

An alternative construction is the one described in Section 2.3.1, as $(D_{10} \times HN).2 = M.G.A$ with $G = 2 \times HN$, $M.G = D_{10} \times HN$, and $G.A$ the subdirect product of $HN.2$ and a cyclic group of order four (which can be constructed as the isoclinic variant of $2 \times HN.2$, see Section 2.2.4).

Here is this construction:

```
─────────── Example ───────────
gap> c2:= CharacterTable( "C2" );;
gap> hn:= CharacterTable( "HN" );;
gap> g:= c2 * hn;;
gap> d10:= CharacterTable( "D10" );;
gap> mg:= d10 * hn;;
gap> nsg:= ClassPositionsOfNormalSubgroups( mg );
[ [ 1 ], [ 1, 55 .. 109 ], [ 1, 55 .. 163 ], [ 1 .. 54 ],
  [ 1 .. 162 ], [ 1 .. 216 ] ]
gap> SizesConjugacyClasses( mg ){ nsg[2] };
[ 1, 2, 2 ]
gap> g:= mg / nsg[2];
CharacterTable( "D10xHN/[ 1, 55, 109 ]" )
gap> help:= c2 * CharacterTable( "HN.2" );
CharacterTable( "C2xHN.2" )
gap> ga:= CharacterTableIsoclinic( help );
CharacterTable( "Isoclinic(C2xHN.2)" )
gap> gfusga:= PossibleClassFusions( g, ga );
[ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 12, 13, 14, 15, 16, 17,
      18, 19, 20, 21, 22, 23, 23, 24, 25, 25, 26, 27, 28, 29, 30, 31,
      32, 32, 33, 33, 34, 35, 36, 37, 37, 38, 39, 40, 40, 41, 42, 42,
      43, 43, 44, 44, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 89,
      90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 101, 102,
      103, 103, 104, 105, 106, 107, 108, 109, 110, 110, 111, 111,
      112, 113, 114, 115, 115, 116, 117, 118, 118, 119, 120, 120,
      121, 121, 122, 122 ],
```

```
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 12, 13, 14, 15, 16, 17,
     18, 19, 20, 21, 22, 23, 23, 24, 25, 25, 26, 27, 28, 29, 30, 31,
     32, 32, 33, 33, 35, 34, 36, 37, 37, 38, 39, 40, 40, 41, 42, 42,
     43, 43, 44, 44, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 89,
     90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 101, 102,
     103, 103, 104, 105, 106, 107, 108, 109, 110, 110, 111, 111,
     113, 112, 114, 115, 115, 116, 117, 118, 118, 119, 120, 120,
     121, 121, 122, 122 ] ]
gap> StoreFusion( g, gfusga[1], ga );
gap> acts:= PossibleActionsForTypeMGA( mg, g, ga );;
gap> Length( acts );
1
gap> poss:= PossibleCharacterTablesOfTypeMGA( mg, g, ga, acts[1],
>               "(D10xHN).2" );;
gap> Length( poss );
1
gap> IsRecord( TransformingPermutationsCharacterTables( poss[1].table,
>               CharacterTable( "(D10xHN).2" ) ) );
true
```

2.9.3   A Counterexample (August 2015)

A group G is called *real* if each of its elements is conjugate in G to its inverse. Equivalently, a group is real if and only if all its character values are real. One might ask whether the Sylow 2-subgroup of a real group is itself real. Counterexamples can be found by a search through GAP's library of small groups. Using the facts we have collected about index two subdirect products in Section 2.3.6, we can demonstrate such a counterexample without using GAP.

Let $H_1 = A_4$, $G_1 = S_4$, $H_2 = C_4$, and G_2 a nonabelian group of order 8, and consider the unique index two subgroup G of $G_1 \times G_2$ that is different from $H_1 \times G_2$ and $G_1 \times H_2$.

Each irreducible character of G either extends to $G_1 \times G_2$ or it is induced from an irreducible character of $H_1 \times H_2$. In the former case, the character is integer valued. Irrational values in the latter case arise as follows.

Let χ be an irreducible character of $H_1 \times H_2$; then it is the product of irreducible characters χ_1 and χ_2 of H_1 and H_2, respectively. If χ has irrational values then χ_1 takes primitive third roots of unity ω, ω^2 on elements of order three in H_1, or χ_2 takes primitive fourth roots of unity $\pm i$ on elements of order four in H_2, or both. In the first two cases, inducing χ to G yields an integer valued character, because each pair of Galois conjugate classes fuses in G on which χ takes irrational values. In the last case, χ takes primitive 12-th roots of unity $\pm i\omega$ and $\pm i\omega^2$ on elements of order 12; since G fuses the classes with the character values $i\omega$ and $-i\omega^2$, we get the character value $i\omega - i\omega^2 = -\sqrt{3}$ in the induced character χ^G. This means that this character is real valued. Hence G is real.

Now we consider a Sylow 2-subgroup of G. It has also the structure of a subdirect product, as follows. Let $H_1 = V_4$, $G_1 = D_8$, and H_2 and G_2 as above, and consider the unique index two subgroup G of $G_1 \times G_2$ that is different from $H_1 \times G_2$ and $G_1 \times H_2$.

As above, irrational values in an irreducible character of G arise only if this character is induced from a character χ, say, that is the product of irreducible characters χ_1 and χ_2 of H_1 and H_2, respectively. In this case, χ_2 takes primitive fourth roots of unity $\pm i$ on elements of order four in H_2. Moreover, χ_1 takes different values ± 1 on the two classes of H_1 that are fused in G if the induced character has irrational values, and these values are $\pm 2i$. Hence the group G is *not* real.

(In fact the above two groups of order 96 are the smallest real groups with non-real Sylow 2-subgroup, and there are no other such groups of this order.)

Chapter 3

Constructing Character Tables of Central Extensions in GAP

Date: February 19th, 2004

This chapter has three aims. First it shows how the GAP system [GAP24] can be utilized to construct character tables of certain central extensions from known character tables; the GAP functions used for that are part of the GAP Character Table Library [Bre25]. Second it documents several constructions of character tables which are contained in the GAP Character Table Library. Third it serves as a testfile for the GAP functions.

A typo (wrong sign of ε^5) in the picture in Section 3.1.4 has been corrected in 2013.

3.1   Coprime Central Extensions

In this section, we will deal with the following situation. Let H be a group, Z be a cyclic central subgroup in H, and $Z = Z_1 Z_2$ for subgroups Z_1 and Z_2 of coprime orders m and n, say. For the sake of simplicity, suppose that both m and n are primes; the general case is then obtained by iterating the construction process.

Our aim is to compute the character table of H from the character tables of H/Z_1 and H/Z_2. We assume that the factor fusions from these tables to that of the common factor group H/Z are known. Again for the sake of simplicity, we will take the character table of H/Z as an input. (See Section 3.2.4 for an example where two different orderings of classes and characters of H/Z arise from the tables of H/Z_1 and H/Z_2.)

For example, the character table of $H = 12.M_{22}$ can be computed from those of $6.M_{22}$ and $4.M_{22}$, and the character table of $6.M_{22}$ can be computed from those of $3.M_{22}$ and $2.M_{22}$ (see Section 3.2.1).

3.1.1   The Character Table Head

The conjugacy classes and power maps of H are uniquely determined by the input data specified above.
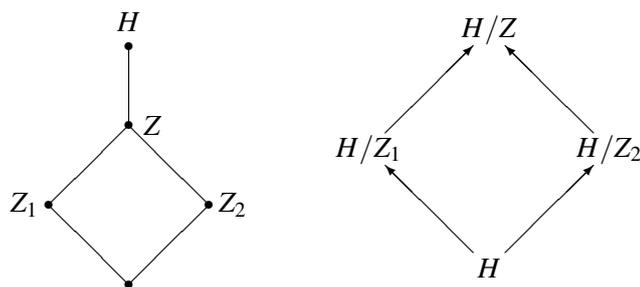
Suppose that a class C of elements of H/Z has n_C preimage classes in H/Z_1 and m_C preimage classes in H/Z_2; then n_C is either 1 or n, and m_C is either 1 or m. The preimage classes of C in H/Z_1 and H/Z_2 are parametrized by $\{j; 0 \le j < n_C\}$ and $\{i; 0 \le i < m_C\}$, respectively, and the preimage classes in H are parametrized by the pairs $\{(i,j); 0 \le i < m_C, 0 \le j < n_C\}$.

The centralizer orders of these classes in H are $m_C n_C$ times the centralizer order of C in H/Z.

The factor fusion onto H/Z_1 is then given by mapping the class with the parameter (i,j) to the class with the parameter j; analogously, the factor fusion onto H/Z_2 maps this class to the class with the parameter i. To see this, let $Z = \langle z \rangle$, and set $z_1 = z^n$ and $z_2 = z^m$. Now take an element $g \in H$ for which gZ lies in C. Then the elements $g z_1^i z_2^j$, $1 \le i \le m_C$, $1 \le j \le n_C$ form a set of representatives of the preimage classes of C in H. In H/Z_1 and H/Z_2, these elements map to $g z_2^j Z_1$, $1 \le j \le n_C$ and $g z_1^i Z_2$, $1 \le i \le m_C$, respectively, which are sets of representatives of the classes in question in these groups.

For each prime p, the factor fusions determine the p-th power map of H from the p-th power maps of H/Z_1 and H/Z_2. To see this, take a class C_0 in H that is a preimage of the class C of H/Z, and let K be the class of p-th powers of the elements in C. Then the image of C_0 under the p-th power map is one of the preimages of K. We know the images of C_0 under the factor fusions to H/Z_1 and H/Z_2, and thus also their images K_1 and K_2 under the p-th power maps of these groups. So the class of p-th powers of the elements in C_0 is the unique class that is mapped to K_1 and K_2 under the factor fusions.

The construction of the character table head of H from the input data specified above is implemented by the GAP function `CharacterTableOfCommonCentralExtension` (**CTblLib: CharacterTableOfCommonCentralExtension**).

3.1.2 The Irreducible Characters

First of all, it should be said that it is not obvious how the irreducible characters of H can be computed from the irreducible characters of H/Z_1 and H/Z_2. Clearly the irreducible characters of the two factor groups can be inflated to H via the factor fusions, so we have to find those irreducibles that have neither Z_1 nor Z_2 in their kernels.

For that, we use the following heuristic. Let ε_z be a complex primitive $|z|$-th root of unity. For integers i, set $\mathrm{Irr}_{z,i}(H) = \{\chi \in \mathrm{Irr}(H); \chi(z) = \varepsilon_z^i \chi(1)\}$. Then $\mathrm{Irr}(H) = \bigcup_{i=0}^{|z|-1} \mathrm{Irr}_{z,i}(H)$, as a disjoint union. If i is a multiple of m or n, respectively, then $\mathrm{Irr}_{z,i}(H)$ consists of the inflations of certain irreducible characters of H/Z_1 or H/Z_2, respectively. The remaining irreducible characters of H lie in $\mathrm{Irr}_{z,i}(H)$ with i coprime to $|z|$. These characters are algebraic conjugates of $\mathrm{Irr}_{z,1}(H)$, so it suffices to compute this subset; the conjugates are then derived as the last step.

Since $\mathrm{Irr}_{z,i}(H) \otimes \mathrm{Irr}_{z,j}(H) \subset \mathbb{Z}[\mathrm{Irr}_{z,i+j}(H)]$ holds, we start with the tensor products of the known irreducible characters in $\mathrm{Irr}_{z,i}(H)$ and $\mathrm{Irr}_{z,j}(H)$ with the property $i + j \equiv 1 \bmod mn$.

For example, if we have $m = 2$ and $n = 3$ then $\mathrm{Irr}_{z,3}(H)$ consists of the inflations of those characters in $\mathrm{Irr}(H/Z_2)$ that are not characters of H/Z, and $\mathrm{Irr}_{z,4}(H)$ consists of the inflations of certain characters in $\mathrm{Irr}(H/Z_1)$ that are not characters of H/Z. The tensor products of these sets of characters lie in the span of $\mathrm{Irr}_{z,1}(H)$.

In general these tensor products are reducible, but some of them may be in fact irreducible, so we first take these irreducibles, and reduce the other tensor products with them. (If H is a direct product of Z and H/Z then all missing irreducibles are obtained this way.)

Then we tensor algebraic conjugates of the known characters in the span of $\mathrm{Irr}_{z,1}(H)$ with characters in suitable sets $\mathrm{Irr}_{z,i}(H)$, in order to get more characters in $\mathrm{Irr}_{z,1}(H)$; for example, $\mathrm{Irr}_{z,1}(H) \otimes \mathrm{Irr}_{z,0}(H)$ is a subset of $\mathbb{Z}[\mathrm{Irr}_{z,1}(H)]$.

In the case $m = 2$ and $n = 3$, also $\mathrm{Irr}_{z,5}(H) \otimes Irr_{z,2}(H)$ yields linear combinations of $\mathrm{Irr}_{z,1}(H)$. Note that $\mathrm{Irr}_{z,5}(H)$ consists of the complex conjugates of $\mathrm{Irr}_{z,1}(H)$.

In the next step, we apply the LLL algorithm (implemented via the GAP function LLL (**Reference: LLL**)) to the set of reducible characters in $\mathbb{Z}[\mathrm{Irr}_{z,1}(H)]$ which we got from the tensor products, and hope to find irreducibles. In the examples shown below, this step yields all desired irreducible characters.

The GAP function `CharacterTableOfCommonCentralExtension` (**CTblLib: CharacterTableOfCommonCentralExtension**) implements the strategy sketched above.

3.1.3 Ordering of Conjugacy Classes

One "natural" choice for the ordering of the columns in the character table of H is given by respecting the ordering of columns in the character table of H/Z, and taking the preimage of the class C corresponding to the parameter $(k \bmod m_C, k \bmod n_C)$ as the k-th class for C.

If the preimages of C in H/Z_1 and H/Z_2 have class representatives gZ_1, gz_2Z_1, $gz_2^2Z_1$, ... and gZ_2, gz_1Z_2, $gz_1^2Z_2$, ..., respectively (in this ordering), then the above rule yields representatives of preimages in H in the ordering g, $g(z_1z_2)$, $g(z_1z_2)^2$, ...

In the case $m = 2$, $n = 3$, the following pattern arises for classes of H/Z that have m and n preimages in H/Z_1 and H/Z_2, respectively. The vertices are labelled by the roots of unity with which the values of the characters in the set $\mathrm{Irr}_{z,1}(H)$ on the first preimage must be multiplied in order to obtain the values on the given class; we have $\omega = \exp(2\pi i/3)$.
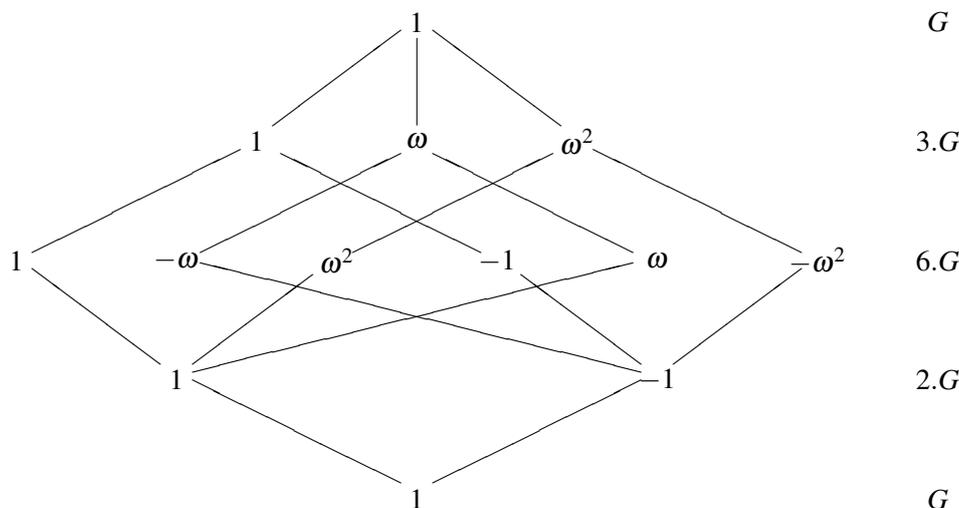
3.1.4 Compatibility with Smaller Factor Groups

It may happen that a cyclic central subgroup Z_0 of H contains Z properly. Then we choose a class ordering relative to that in the factor group H/Z_0, mainly because the Atlas tables of this type are sorted this way.

The typical case is the character table of a central extension of the type $12.G$ that shall be constructed from the character tables of the groups of the types $4.G$ and $6.G$; here we prefer to order the preimages of a class in the smaller factor group of the type G according to the above rule. This results in the following pattern, where $\varepsilon = \exp(2\pi i/12)$ holds (cf. Section "ATLAS Tables" in the manual of the GAP Character Table Library).
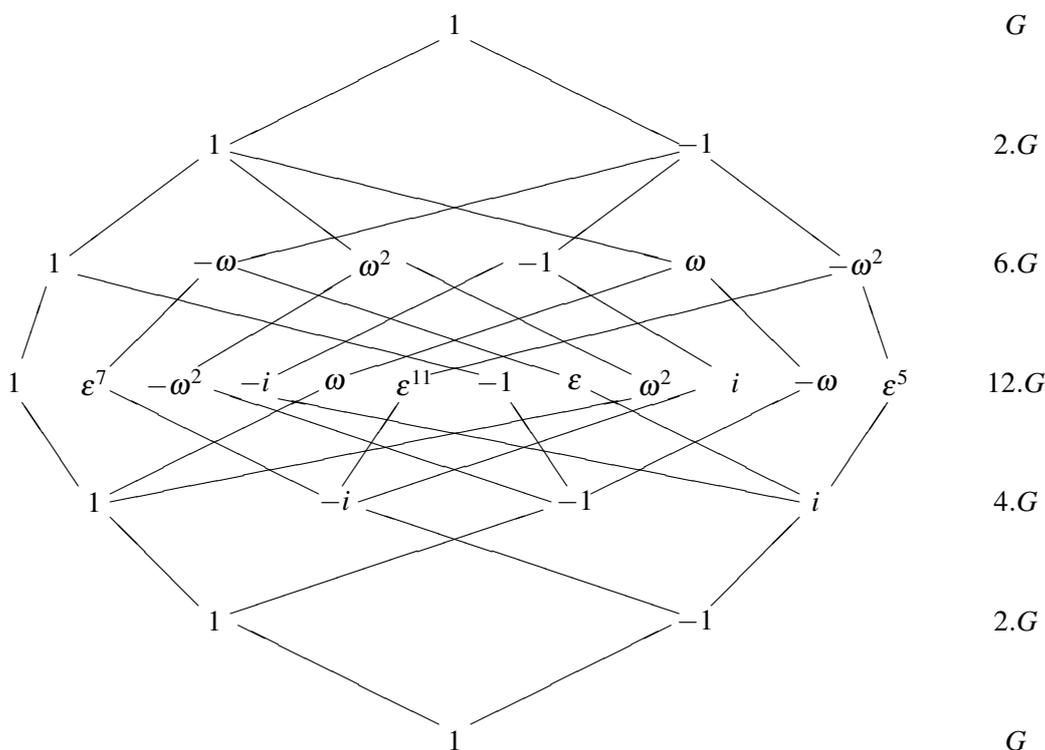
A more important aspect concerns the computation of the irreducible characters. Let $Z_0 = \langle z_0 \rangle$. Instead of computing $\mathrm{Irr}_{z,1}(H)$, we compute the set $\mathrm{Irr}_{z_0,1}(H)$.

In the computation of the character table of a central extension of the type $12.G$ as mentioned above, with $|z_0| = 12$, we start with the characters

$$\mathrm{Irr}_{z_0,3}(H) \otimes \mathrm{Irr}_{z_0,10}(H) \cup \mathrm{Irr}_{z_0,4}(H) \otimes \mathrm{Irr}_{z_0,9}(H) \subseteq \mathbb{Z}[\mathrm{Irr}_{z_0,1}(H)],$$

and later form tensor products involving algebraic conjugates of the characters in the span of $\mathrm{Irr}_{z_0,1}(H)$, using that

$$\mathrm{Irr}_{z_0,1}(H) \otimes \mathrm{Irr}_{z_0,0}(H) \cup \mathrm{Irr}_{z_0,2}(H) \otimes \mathrm{Irr}_{z_0,11}(H) \cup \mathrm{Irr}_{z_0,5}(H) \otimes \mathrm{Irr}_{z_0,8}(H) \cup \mathrm{Irr}_{z_0,6}(H) \otimes \mathrm{Irr}_{z_0,7}(H)$$

is a subset of $\mathbb{Z}[\mathrm{Irr}_{z_0,1}(H)]$.

Without that modification, the computation of irreducibles is significantly more involved.

The GAP function `CharacterTableOfCommonCentralExtension` (**CTblLib: CharacterTableOfCommonCentralExtension**) chooses the class ordering relative to larger cyclic factor groups, as in the above picture, and also uses the above refinement for the computation of irreducible characters.

3.2 Examples

The following examples use the GAP Character Table Library, so we first load this package.

```
───────────────── Example ─────────────────
gap> LoadPackage( "ctbllib", false );
true
```

3.2.1 Central Extensions of Simple Atlas Groups

For the following groups, the Atlas contains the character tables of central extensions $M.G$ of simple groups G with $|M|$ divisible by two different primes; in all these cases, we have $M \in \{6, 12\}$.

(The entry concerning $6.^2E_6(2)$ has been added to the list after the character table of $3.^2E_6(2)$ became available. This table has been computed by Frank Lübeck.)

```
───────────────── Example ─────────────────
gap> list:= [
>     #         G          m.G          n.G          mn.G
>
>     [      "A6",      "2.A6",      "3.A6",       "6.A6" ],
>     [      "A7",      "2.A7",      "3.A7",       "6.A7" ],
>     [    "L3(4)",   "2.L3(4)",   "3.L3(4)",     "6.L3(4)" ],
>     [ "2.L3(4)", "4_1.L3(4)",   "6.L3(4)",  "12_1.L3(4)" ],
>     [ "2.L3(4)", "4_2.L3(4)",   "6.L3(4)",  "12_2.L3(4)" ],
>     [     "M22",     "2.M22",     "3.M22",       "6.M22" ],
>     [   "2.M22",     "4.M22",     "6.M22",      "12.M22" ],
>     [   "U4(3)",   "2.U4(3)", "3_1.U4(3)",   "6_1.U4(3)" ],
>     [   "U4(3)",   "2.U4(3)", "3_2.U4(3)",   "6_2.U4(3)" ],
>     [ "2.U4(3)",   "4.U4(3)", "6_1.U4(3)",  "12_1.U4(3)" ],
>     [ "2.U4(3)",   "4.U4(3)", "6_2.U4(3)",  "12_2.U4(3)" ],
>     [   "O7(3)",   "2.O7(3)",   "3.O7(3)",      "6.O7(3)" ],
>     [   "U6(2)",   "2.U6(2)",   "3.U6(2)",      "6.U6(2)" ],
>     [     "Suz",     "2.Suz",     "3.Suz",       "6.Suz" ],
>     [    "Fi22",    "2.Fi22",    "3.Fi22",       "6.Fi22" ],
>     [  "2E6(2)",  "2.2E6(2)",  "3.2E6(2)",    "6.2E6(2)" ],
>   ];;
```

As was discussed in the sections 3.1.3 and 3.1.4, the class ordering of the result tables is the same as that in the GAP library tables, so it is enough to check whether the set of characters in the computed table coincides with the set of characters in the library table.

In order to list information about the progress, we set the relevant info level to 1.

```
───────────────── Example ─────────────────
gap> SetInfoLevel( InfoCharacterTable, 1 );
gap> for entry in list do
>    id    := entry[4];
>    tblG  := CharacterTable( entry[1] );
```

```
>    tblmG := CharacterTable( entry[2] );
>    tblnG := CharacterTable( entry[3] );
>    lib   := CharacterTable( id );
>    res:= CharacterTableOfCommonCentralExtension( tblG, tblmG, tblnG, id );
>    if not res.IsComplete then
>      Print( "#E  not complete: ", id, "\n" );
>    fi;
>    if not IsSubset( Irr( lib ), res.irreducibles ) then
>      Print( "#E  inconsistent: ", id, "\n" );
>    fi;
> od;
#I  6.A6: need 4 faithful irreducibles
#I  6.A6: 4 found by tensoring
#I  6.A7: need 5 faithful irreducibles
#I  6.A7: 5 found by tensoring
#I  6.L3(4): need 7 faithful irreducibles
#I  6.L3(4): 7 found by LLL
#I  12_1.L3(4): need 5 faithful irreducibles
#I  12_1.L3(4): 2 found by tensoring
#I  12_1.L3(4): 3 found by tensoring
#I  12_2.L3(4): need 6 faithful irreducibles
#I  12_2.L3(4): 6 found by LLL
#I  6.M22: need 10 faithful irreducibles
#I  6.M22: 1 found by tensoring
#I  6.M22: 9 found by LLL
#I  12.M22: need 7 faithful irreducibles
#I  12.M22: 7 found by LLL
#I  6_1.U4(3): need 15 faithful irreducibles
#I  6_1.U4(3): 1 found by tensoring
#I  6_1.U4(3): 14 found by LLL
#I  6_2.U4(3): need 12 faithful irreducibles
#I  6_2.U4(3): 12 found by LLL
#I  12_1.U4(3): need 12 faithful irreducibles
#I  12_1.U4(3): 4 found by tensoring
#I  12_1.U4(3): 8 found by tensoring
#I  12_2.U4(3): need 9 faithful irreducibles
#I  12_2.U4(3): 9 found by LLL
#I  6.O7(3): need 12 faithful irreducibles
#I  6.O7(3): 2 found by tensoring
#I  6.O7(3): 10 found by LLL
#I  6.U6(2): need 28 faithful irreducibles
#I  6.U6(2): 2 found by tensoring
#I  6.U6(2): 26 found by LLL
#I  6.Suz: need 29 faithful irreducibles
#I  6.Suz: 29 found by LLL
#I  6.Fi22: need 34 faithful irreducibles
#I  6.Fi22: 4 found by tensoring
#I  6.Fi22: 30 found by LLL
#I  6.2E6(2): need 60 faithful irreducibles
#I  6.2E6(2): 60 found by LLL
gap> SetInfoLevel( InfoCharacterTable, 0 );
```

We see that in all cases, the irreducible characters of the groups $M.G$ are obtained by reducing tensor products and applying the LLL algorithm.

3.2.2 Central Extensions of Other Atlas Groups

The following cases also fit to the pattern introduced above.

(The following examples were added in October 2006.)

The group $(2^2 \times 3).L_3(4)$ can be viewed as a common central extension of its factor group $2.L_3(4)$ by the two groups $2^2.L_3(4)$ and $6.L_3(4)$.

Analogously, the group $(4^2 \times 3).L_3(4)$ can be viewed as a common central extension of its factor group $(2 \times 4).L_3(4)$ by the two groups $4^2.L_3(4)$ and $(2 \times 12).L_3(4)$.

Finally, the group $(2 \times 12).L_3(4)$ can be viewed as a common central extension of the factor group $2^2.L_3(4)$ by the two groups $(2 \times 4).L_3(4)$ and $(2^2 \times 3).L_3(4)$.

The construction of the character tables of the involved factor groups $2^2.L_3(4)$ and $(2 \times 4).L_3(4)$, as well as an alternative construction of the table of $(2 \times 12).L_3(4)$ can be found in the sections 2.7.2 and 2.7.4.

```
———————————————— Example ————————————————
gap> list2:= [
>    [ "2.L3(4)",     "2^2.L3(4)",   "6.L3(4)",       "(2^2x3).L3(4)" ],
>    [ "2^2.L3(4)",   "(2x4).L3(4)", "(2^2x3).L3(4)", "(2x12).L3(4)"  ],
>    [ "(2x4).L3(4)", "4^2.L3(4)",   "(2x12).L3(4)",  "(4^2x3).L3(4)" ],
>    ];;
```

(The following examples were added in December 2010.)

The group $(3^2 \times 2).U_4(3)$ can be viewed as a common central extension of its factor group $3_1.U_4(3)$ by the two groups $6_1.U_4(3)$ and $3^2.U_4(3)$, or as a common central extension of its factor group $3_2.U_4(3)$ by the two groups $6_2.U_4(3)$ and $3^2.U_4(3)$.

Analogously, the group $(3^2 \times 4).U_4(3)$ can be viewed as a common central extension of its factor group $6_1.U_4(3)$ by the two groups $12_1.U_4(3)$ and $(3^2 \times 2).U_4(3)$, or as a common central extension of its factor group $6_2.U_4(3)$ by the two groups $12_2.U_4(3)$ and $(3^2 \times 2).U_4(3)$.

```
———————————————— Example ————————————————
gap> Append( list2, [
>    [ "3_1.U4(3)",   "6_1.U4(3)",   "3^2.U4(3)",     "(3^2x2).U4(3)" ],
>    [ "3_2.U4(3)",   "6_2.U4(3)",   "3^2.U4(3)",     "(3^2x2).U4(3)" ],
>    [ "6_1.U4(3)",   "12_1.U4(3)",  "(3^2x2).U4(3)", "(3^2x4).U4(3)" ],
>    [ "6_2.U4(3)",   "12_2.U4(3)",  "(3^2x2).U4(3)", "(3^2x4).U4(3)" ],
>    ] );
gap> SetInfoLevel( InfoCharacterTable, 1 );
gap> for entry in list2 do
>    id    := entry[4];
>    tblG  := CharacterTable( entry[1] );
>    tblmG := CharacterTable( entry[2] );
>    tblnG := CharacterTable( entry[3] );
>    lib   := CharacterTable( id );
>    res:= CharacterTableOfCommonCentralExtension(
>            tblG, tblmG, tblnG, id );
>    if not res.IsComplete then
>      Print( "#E  not complete: ", id, "\n" );
>    fi;
```

```
>    if TransformingPermutationsCharacterTables( res.tblmnG, lib )
>         = fail then
>      Print( "#E  inconsistent: ", id, "\n" );
>    fi;
> od;
#I  (2^2x3).L3(4): need 14 faithful irreducibles
#I  (2^2x3).L3(4): 14 found by tensoring
#I  (2x12).L3(4): need 11 faithful irreducibles
#I  (2x12).L3(4): 7 found by tensoring
#I  (2x12).L3(4): 4 found by LLL
#I  (4^2x3).L3(4): need 22 faithful irreducibles
#I  (4^2x3).L3(4): 14 found by tensoring
#I  (4^2x3).L3(4): 8 found by LLL
#I  (3^2x2).U4(3): need 39 faithful irreducibles
#I  (3^2x2).U4(3): 27 found by tensoring
#I  (3^2x2).U4(3): 12 found by LLL
#I  (3^2x2).U4(3): need 42 faithful irreducibles
#I  (3^2x2).U4(3): 2 found by tensoring
#I  (3^2x2).U4(3): 40 found by LLL
#I  (3^2x4).U4(3): need 30 faithful irreducibles
#I  (3^2x4).U4(3): 6 found by tensoring
#I  (3^2x4).U4(3): 8 found by tensoring
#I  (3^2x4).U4(3): 16 found by LLL
#I  (3^2x4).U4(3): need 33 faithful irreducibles
#I  (3^2x4).U4(3): 9 found by tensoring
#I  (3^2x4).U4(3): 18 found by tensoring
#I  (3^2x4).U4(3): 6 found by further tensoring
gap> SetInfoLevel( InfoCharacterTable, 0 );
```

3.2.3   Compatible Central Extensions of Maximal Subgroups

The GAP Character Table Library contains the character tables of all maximal subgroups of the groups
$4.M_{22}$, $3.M_{22}$, $2.Suz$, and $3.Suz$. So we can use the approach from Section 3.1 for computing the
character tables of the maximal subgroups of $6.M_{22}$, $12.M_{22}$, and $6.Suz$.

   These tables are contained in the GAP Character Table Library. Several of the groups are direct
products, and the library tables of direct products are usually stored in the form of Kronecker products
of the tables of the factors, so the class ordering of the result tables does not necessarily coincide with
the class ordering in the library tables.

```
────────────────────────────── Example ──────────────────────────────
gap> sublist:= list{ [ 6, 7, 14 ] };
[ [ "M22", "2.M22", "3.M22", "6.M22" ],
  [ "2.M22", "4.M22", "6.M22", "12.M22" ],
  [ "Suz", "2.Suz", "3.Suz", "6.Suz" ] ]
gap> for entry in sublist do
>    tblG  := CharacterTable( entry[1] );
>    tblmG := CharacterTable( entry[2] );
>    tblnG := CharacterTable( entry[3] );
>    lib   := CharacterTable( entry[4] );
>
>    maxesG   := List( Maxes( tblG ), CharacterTable );
>    maxesmG  := List( Maxes( tblmG ), CharacterTable );
```

```
>    maxesnG  := List( Maxes( tblnG ), CharacterTable );
>    maxeslib := List( Maxes( lib ), CharacterTable );
>
>    for i in [ 1 .. Length( maxesG ) ] do
>      id:= Identifier( maxeslib[i] );
>      res:= CharacterTableOfCommonCentralExtension( maxesG[i],
>              maxesmG[i], maxesnG[i], id );
>      if not res.IsComplete then
>        Print( "#E  not complete: ", id, "\n" );
>      fi;
>      if not IsSubset( Irr( maxeslib[i] ), res.irreducibles ) then
>        trans:= TransformingPermutationsCharacterTables( maxeslib[i],
>                                              res.tblmnG );
>        if not IsRecord( trans ) then
>          Print( "#E  not transformable: ", id, "\n" );
>        fi;
>      fi;
>    od;
> od;
```

Since we get no output, all tables in question can be computed with the GAP functions, and coincide (up to permutations of rows and columns) with the library tables.

3.2.4   The 2B Centralizer in $3.Fi'_{24}$ (January 2004)

As is stated in [CCN+85, p. 207], the 2B centralizer N_0 in the sporadic simple Fischer group Fi'_{24} has the structure $2^{1+12}_+.3U_4(3).2_2$. The character table of N_0 is contained in the GAP Character Table Library since the year 2000.

Our aim is to compute the character table of the preimage N of N_0 in the central extension $3.Fi'_{24}$ of Fi'_{24}; let Z_1 denote the centre of $3.Fi'_{24}$.

Using the "dihedral group method" in the faithful permutation representation of degree $920\,808$ for $3.Fi'_{24}$, we first compute a generating set of N. This group has three orbits of the lengths $774\,144$, $145\,152$, and 1512; the actions on the first two orbits are faithful, and the action on the orbit of length 1512 (which consists of the fixed points of the central involution of N) has kernel exactly the central subgroup Z_2, say, of order 2 in N.

Since the permutation representation on 1512 points is so small, it is straightforward to compute the character table of N/Z_2 using the implementation of Dixon's algorithm in GAP; now this table is part of the GAP Character Table Library.

Note that N is a central extension of $N_0/Z(N_0)$ by the cyclic group $Z = Z_1 Z_2$ of order 6, and that we know the character tables of the groups N/Z_1 and N/Z_2. So we can apply the method described in Section 3.1 for computing the character table of N.

First we fetch the input data.

```
──────────────── Example ────────────────
gap> tblmG := CharacterTable( "F3+N2B" );;
gap> tblG  := tblmG / ClassPositionsOfCentre( tblmG );;
gap> tblnG := CharacterTable( "2^12.3^2.U4(3).2_2'" );;
```

The character tables of the library table of N_0 and the character table of N/Z_2 obtained from the permutation group are not compatible in the sense that the tables of the factor groups modulo the centres are not sorted compatibly, so we have to compute and store the fusion from `tblnG` to `tblG`.

```
                        ___ Example ___
  gap> f2:= tblnG / ClassPositionsOfCentre( tblnG );;
  gap> trans:= TransformingPermutationsCharacterTables( f2, tblG );;
  gap> tblnGfustblG:= OnTuples( GetFusionMap( tblnG, f2 ),
  >                                  trans.columns );;
  gap> StoreFusion( tblnG, tblnGfustblG, tblG );
  gap> IsSubset( Irr( tblnG ),
  >             List( Irr( tblG ), x -> x{ tblnGfustblG } ) );
  true
```

Now we apply `CharacterTableOfCommonCentralExtension` (**CTblLib: CharacterTableOf-CommonCentralExtension**).

```
                        ___ Example ___
  gap> SetInfoLevel( InfoCharacterTable, 1 );
  gap> id:= "3.2^(1+12).3U4(3).2";;
  gap> res:= CharacterTableOfCommonCentralExtension(
  >             tblG, tblmG, tblnG, id );;
  #I  3.2^(1+12).3U4(3).2: need 36 faithful irreducibles
  #I  3.2^(1+12).3U4(3).2: 16 found by tensoring
  #I  3.2^(1+12).3U4(3).2: 20 found by LLL
  gap> SetInfoLevel( InfoCharacterTable, 0 );
```

So we have found all missing irreducibles of *N*. Let us check whether the result table coincides with the table in the GAP Character Table Library.

```
                        ___ Example ___
  gap> lib:= CharacterTable( "3.F3+N2B" );;
  gap> IsRecord( TransformingPermutationsCharacterTables(
  >             res.tblmnG, lib ) );
  true
```

We were interested in the character table because *N* is a maximal subgroup of $3.Fi'_{24}$. So the class fusion into the table of this group is an interesting information. We assume that the class fusion of N_0 into Fi'_{24} is known, and compute only those possible class fusions that are compatible with this map.

```
                        ___ Example ___
  gap> 3f3p:= CharacterTable( "3.F3+" );;
  gap> f3p:= CharacterTable( "F3+" );;
  gap> approxfus:= CompositionMaps(
  >             InverseMap( GetFusionMap( 3f3p, f3p ) ),
  >             CompositionMaps( GetFusionMap( tblmG, f3p ),
  >                 GetFusionMap( lib, tblmG ) ) );;
  gap> poss:= PossibleClassFusions( lib, 3f3p,
  >             rec( fusionmap:= approxfus ) );;
  gap> Length( poss );
  1
```

It turns out that only one map has this property. (Without the condition on the compatibility, we would have got 128 possibilities, which form one orbit under table automorphisms.)

Chapter 4

GAP Computations Concerning Hamiltonian Cycles in the Generating Graphs of Finite Groups

Date: April 24th, 2012

This is a collection of examples showing how the GAP system [GAP24] can be used to compute information about the generating graphs of finite groups. It includes all examples that were needed for the computational results in [BGL$^+$10].

The purpose of this writeup is twofold. On the one hand, the computations are documented this way. On the other hand, the GAP code shown for the examples can be used as test input for automatic checking of the data and the functions used.

A first version of this document, which was based on GAP 4.4.12, is available in the arXiv at `http://arxiv.org/abs/0911.5589v1` since November 2009. The differences between this file and the current document are as follows.

- The format of the GAP output was adjusted to the changed behaviour of GAP 4.5.

- The records returned by `IsomorphismTypeInfoFiniteSimpleGroup` (**Reference: IsomorphismTypeInfoFiniteSimpleGroup**) contain a component `"shortname"` since GAP 4.11.

- The lower bounds computed for the sporadic simple Monster group have been improved in three steps.

  First, the existence of exactly one class of maximal subgroups of the type $PSL(2, 41)$ (see [NW13]) and the nonexistence of maximal subgroups with socle $PSL(2, 27)$ (see [Wil10]) have been incorporated.

  Second, the classification of classes of maximal subgroups of the Monster has been completed in [DLP25]. As a consequence, the nonexistence of maximal subgroups with socle $Sz(8)$ and $PSU(3, 8)$ and the existence of exactly one class of maximal subgroups with the isomorphism types $PSL(2, 13).2$ and $PSU(3, 4).4$ have been proved, and the proof of the nonexistence of the previously claimed subgroups of the type $L_2(59)$ yields that the Monster has maximal subgroups of the type $59 : 29$. Moreover, meanwhile also all class fusions of the maximal subgroups are known; previously, we got only candidates for some primitive permutation characters.

Third, all character tables of maximal subgroups of the Monster group and their class fusions are available since version 1.3.10 of CTblLib, hence no special treatment for the Monster group is needed anymore in order to compute its primitive permutation characters. In particular, the data file `data/prim_perm_M.json` of CTblLib is no longer needed. In earlier versions, that file had been used to collect information about primitive permutation characters for which the character table was not yet available.

See Section 4.4.2 for comments on earlier versions, and for the bounds that had been computed from the partial information that was available at that time.

4.1 Overview

The purpose of this note is to document the GAP computations that were carried out in order to obtain the computational results in [BGL+10].

In order to keep this note self-contained, we first describe the theory needed, in Section 4.2. The translation of the relevant formulae into GAP functions can be found in Section 4.3. Then Section 4.4 describes the computations that only require (ordinary) character tables in the GAP Character Table Library [Bre25]. Computations using also the groups are shown in Section 4.5.

The examples use the GAP Character Table Library and the GAP Library of Tables of Marks, so we first load these packages in the required versions.

```
_____ Example _____
  gap> if not CompareVersionNumbers( GAPInfo.Version, "4.5" ) then
  >        Error( "need GAP in version at least 4.5" );
  >    fi;
  gap> LoadPackage( "ctbllib", "1.2", false );
  true
  gap> LoadPackage( "tomlib", "1.1.1", false );
  true
```

4.2 Theoretical Background

Let G be a finite noncyclic group and denote by $G^\times$ the set of nonidentity elements in G. We define the *generating graph* $\Gamma(G)$ as the undirected graph on the vertex set $G^\times$ by joining two elements $x, y \in G^\times$ by an edge if and only if $\langle x, y \rangle = G$ holds. For $x \in G^\times$, the *vertex degree* $d(\Gamma, x)$ is $|\{y \in G^\times; \langle x, y \rangle = G\}|$. The *closure* $\mathrm{cl}(\Gamma)$ of the graph Γ with m vertices is defined as the graph with the same vertex set as Γ, where the vertices x, y are joined by an edge if they are joined by an edge in Γ or if $d(\Gamma, x) + d(\Gamma, y) \geq m$. We denote iterated closures by $\mathrm{cl}^{(i)}(\Gamma) = \mathrm{cl}(\mathrm{cl}^{(i-1)}(\Gamma))$, where $\mathrm{cl}^{(0)}(\Gamma) = \Gamma$.

In the following, we will show that the generating graphs of the following groups contain a Hamiltonian cycle:

- Nonabelian simple groups of orders at most 10^7,

- groups G containing a unique minimal normal subgroup N such that N has order at most 10^6, N is nonsolvable, and G/N is cyclic,

- sporadic simple groups and their automorphism groups.

Clearly the condition that G/N is cyclic for all nontrivial normal subgroups N of G is necessary for $\Gamma(G)$ being connected, and [BGL$^+$10, Conjecture 1.6] states that this condition is also sufficient. By [BGL$^+$10, Proposition 1.1], this conjecture is true for all solvable groups, and the second entry in the above list implies that this conjecture holds for all nonsolvable groups of order up to 10^6.

The question whether a graph Γ contains a Hamiltonian cycle (i. e., a closed path in Γ that visits each vertex exactly once) can be answered using the following sufficient criteria (see [BGL$^+$10]). Let $d_1 \leq d_2 \leq \cdots \leq d_m$ be the vertex degrees in Γ.

Pósa's criterion:
> If $d_k \geq k+1$ holds for $1 \leq k < m/2$ then Γ contains a Hamiltonian cycle.

Chvátal's criterion:
> If $d_k \geq k+1$ or $d_{m-k} \geq m-k$ holds for $1 \leq k < m/2$ then Γ contains a Hamiltonian cycle.

Closure criterion:
> A graph contains a Hamiltonian cycle if and only if its closure contains a Hamiltonian cycle.

4.2.1 Character-Theoretic Lower Bounds for Vertex Degrees

Using character-theoretic methods similar to those used to obtain the results in [BGK08] (the computations for that paper are shown in [Breb]), we can compute lower bounds for the vertex degrees in generating graphs, as follows.

Let R be a set of representatives of conjugacy classes of nonidentity elements in G, fix $s \in G^{\times}$, let $\mathbb{M}(G,s)$ denote the set of those maximal subgroups of G that contain s, let $\mathbb{M}(G,s)/\sim$ denote a set of representatives in $\mathbb{M}(G,s)$ w. r. t. conjugacy in G. For a subgroup M of G, the *permutation character* 1_M^G is defined by

$$1_M^G(g) := (|G| \cdot |g^G \cap M|)/(|M| \cdot |g^G|),$$

where $g^G = \{g^x; x \in G\}$, with $g^x = x^{-1}gx$, denotes the conjugacy class of g in G. So we have $1_M^G(1) = |G|/|M|$ and thus $|g^G \cap M| = |g^G| \cdot 1_M^G(g)/1_M^G(1)$.

Doubly counting the set $\{(s^x, M^y); x, y \in G, s^x \in M^y\}$ yields $|M^G| \cdot |s^G \cap M| = |s^G| \cdot |\{M^x; x \in G, s \in M^x\}|$ and thus $|\{M^x; x \in G, s \in M^x\}| = |M^G| \cdot 1_M^G(s)/1_M^G(1) \leq 1_M^G(s)$. (If M is a *maximal* subgroup of G then either M is normal in G or self-normalizing, and in the latter case the inequality is in fact an equality.)

Let Π denote the multiset of *primitive* permutation characters of G, i. e., of the permutation characters 1_M^G where M ranges over representatives of the conjugacy classes of maximal subgroups of G.

Define

$$\delta(s, g^G) := |g^G| \cdot \max\left\{0, 1 - \sum_{\pi \in \Pi} \pi(g) \cdot \pi(s)/\pi(1)\right\}$$

and $d(s, g^G) := |\{x \in g^G; \langle s, x \rangle = G\}|$, the contribution of the class g^G to the vertex degree of s. Then we have $d(\Gamma(G), s) = \sum_{x \in R} d(s, x^G)$ and

$$
\begin{aligned}
d(s, g^G) &= |g^G| - |\bigcup_{M \in \mathscr{M}(G,s)} \{x \in g^G; \langle x, s \rangle \subseteq M\}| \\
&\geq \max\left\{0, |g^G| - \sum_{M \in \mathscr{M}(G,s)} |g^G \cap M|\right\}
\end{aligned}
$$

$$= |g^G| \cdot \max \left\{ 0, 1 - \sum_{M \in \mathcal{M}(G,s)} 1_M^G(g)/1_M^G(1) \right\}$$

$$\geq |g^G| \cdot \max \left\{ 0, 1 - \sum_{M \in \mathcal{M}(G,s)/\sim} 1_M^G(g) \cdot 1_M^G(s)/1_M^G(1) \right\}$$

$$= \delta(s, g^G)$$

So $\delta(s) := \sum_{x \in R} \delta(s, x^G)$ is a lower bound for the vertex degree of s; this bound can be computed if Π is known.

For computing the vertex degrees of the iterated closures of $\Gamma(G)$, we define $d^{(0)}(s, g^G) := d(s, g^G)$ and

$$d^{(i+1)}(s, g^G) := \left\{ \begin{array}{ll} |g^G| & ; \quad d^{(i)}(\Gamma(G), s) + d^{(i)}(\Gamma(G), g) \geq |G| - 1 \\ d^{(i)}(s, g^G) & ; \quad \text{otherwise} \end{array} \right. .$$

and $\delta^{(i)}(s) := \sum_{x \in R} \delta^{(i)}(s, x^G)$, a lower bound for $d(\mathrm{cl}^{(i)}(\Gamma(G)), s)$ that can be computed if Π is known.

4.2.2 Checking the Criteria

Let us assume that we know lower bounds $\beta(s)$ for the vertex degrees $d(\mathrm{cl}^{(i)}(\Gamma(G)), s)$, for some fixed i, and let us choose representatives $s_1, s_2, \ldots, s_l$ of the nonidentity conjugacy classes of G such that $\beta(s_1) \leq \beta(s_2) \leq \cdots \leq \beta(s_l)$ holds. Let $c_k = |s_k^G|$ be the class lengths of these representatives.

Then the first c_1 vertex degrees, ordered by increasing size, are larger than or equal to $\beta(s_1)$, the next c_2 vertex degrees are larger than or equal to $\beta(s_2)$, and so on.

Then the set of indices in the k-th nonidentity class of G for which Pósa's criterion is not guaranteed by the given bounds is

$$\{x; c_1 + c_2 + \cdots + c_{k-1} < x \leq c_1 + c_2 + \cdots c_k, x < (|G| - 1)/2, \beta(s_k) < x+1\}.$$

This is an interval $\{L_k, L_k + 1, \ldots, U_k\}$ with

$$L_k = \max\{1 + c_1 + c_2 + \cdots + c_{k-1}, \beta(s_k)\}$$

and

$$U_k = \min\{c_1 + c_2 + \cdots + c_k, \lfloor |G|/2 \rfloor - 1\}.$$

(Note that the generating graph has $m = |G| - 1$ vertices, and that $x < m/2$ is equivalent to $x \leq \lfloor |G|/2 \rfloor - 1$.)

The generating graph $\Gamma(G)$ satisfies Pósa's criterion if all these intervals are empty, i. e., if $L_k > U_k$ holds for $1 \leq k \leq l$.

The set of indices for which Chvátal's criterion is not guaranteed is the intersection of

$$\{m - k; 1 \leq m - k < m/2, d_k < k\}$$

with the set of indices for which Pósa's criterion is not guaranteed.

Analogously to the above considerations, the set of indices $m - x$ in the former set for which Chvátal's criterion is not guaranteed by the given bounds and such that x is an index in the k-th nonidentity class of G is

$$\{m - x; c_1 + c_2 + \cdots + c_{k-1} < x \leq c_1 + c_2 + \cdots c_k, 1 \leq m - x < (|G| - 1)/2, \beta(s_k) < x\}.$$

This is again an interval $\{L'_k, L'_k + 1, \ldots, U'_k\}$ with

$$L'_k = \max\{1, m - (c_1 + c_2 + \cdots + c_k)\}$$

and

$$U'_k = \min\{m - (c_1 + c_2 + \cdots + c_{k-1}) - 1, \lfloor |G|/2 \rfloor - 1, m - 1 - \beta(s_k)\}.$$

The generating graph $\Gamma(G)$ satisfies Chvátal's criterion if the union of the intervals $\{L'_k, L'_k + 1, \ldots, U'_k\}$, for $1 \le k \le l$ is disjoint to the union of the intervals $\{L_k, L_k + 1, \ldots, U_k\}$, for $1 \le k \le l$.

4.3 GAP Functions for the Computations

We describe two approaches to compute, for a given group G, vertex degrees for the generating graph of G or lower bounds for them, by calculating exact vertex degrees from G itself (see Section 4.3.1) or by deriving lower bounds for the vertex degrees using just character-theoretic information about G and its subgroups (see Section 4.3.2). Finally, Section 4.3.3 deals with deriving lower bounds of vertex degrees of iterated closures.

4.3.1 Computing Vertex Degrees from the Group

In this section, the task is to compute the vertex degrees $d(s, g^G)$ using explicit computations with the group G.

The function `IsGeneratorsOfTransPermGroup` checks whether the permutations in the list `list` generate the permutation group `G`, *provided that* `G` is transitive on its moved points. (Note that testing the necessary condition that the elements in `list` generate a transitive group is usually much faster than testing generation.) This function has been used already in [Breb].

```
                                   Example
  gap> IsGeneratorsOfTransPermGroup:= function( G, list )
  >       local S;
  >
  >       if not IsTransitive( G ) then
  >         Error( "<G> must be transitive on its moved points" );
  >       fi;
  >       S:= SubgroupNC( G, list );
  >
  >       return IsTransitive( S, MovedPoints( G ) )
  >             and Size( S ) = Size( G );
  > end;;
```

The function `VertexDegreesGeneratingGraph` takes a *transitive* permutation group G (in order to be allowed to use `IsGeneratorsOfTransPermGroup`), the list `classes` of conjugacy classes of G (in order to prescribe an ordering of the classes), and a list `normalsubgroups` of proper normal subgroups of G, and returns the matrix $[d(s, g^G)]_{s,g}$ of vertex degrees, with rows and columns indexed by nonidentity class representatives ordered as in the list `classes`. (The class containing the identity element may be contained in `classes`.)

The following criteria are used in this function.

- The function tests the (non)generation only for representatives of $C_G(g)$–$C_G(s)$–double cosets, where $C_G(g) := \{x \in G; gx = xg\}$ denotes the centralizer of g in G. Note that for $c_1 \in C_G(g)$,

$c_2 \in C_G(s)$, and a representative $r \in G$, we have $\langle s, g^{c_1 r c_2} \rangle = \langle s, g^r \rangle^{c_2}$. If $\langle s, g^r \rangle = G$ then the double coset $D = C_G(g) r C_G(s)$ contributes $|D|/|C_G(g)|$ to the vertex degree $d(s, g^G)$, otherwise the contribution is zero.

- We have $d(s, g^G) \cdot |C_G(g)| = d(g, s^G) \cdot |C_G(s)|$. (To see this, either establish a bijection of the above double cosets, or doubly count the edges between elements of the conjugacy classes of s and g.)

- If $\langle s_1 \rangle = \langle s_2 \rangle$ and $\langle g_1 \rangle = \langle g_2 \rangle$ hold then we have $d(s_1, g_1^G) = d(s_2, g_1^G) = d(s_1, g_2^G) = d(s_2, g_2^G)$, so only one of these values must be computed.

- If both s and g are contained in one of the normal subgroups given then $d(s, g^G)$ is zero.

- If G is not a dihedral group and both s and g are involutions then $d(s, g^G)$ is zero.

```
——————————————— Example ———————————————
gap> BindGlobal( "VertexDegreesGeneratingGraph",
>     function( G, classes, normalsubgroups )
>     local nccl, matrix, cents, powers, normalsubgroupspos, i, j, g_i,
>           nsg, g_j, gen, pair, d, pow;
>
>     if not IsTransitive( G ) then
>       Error( "<G> must be transitive on its moved points" );
>     fi;
>
>     classes:= Filtered( classes,
>                         C -> Order( Representative( C ) ) <> 1 );
>     nccl:= Length( classes );
>     matrix:= [];
>     cents:= [];
>     powers:= [];
>     normalsubgroupspos:= [];
>     for i in [ 1 .. nccl ] do
>       matrix[i]:= [];
>       if IsBound( powers[i] ) then
>         # The i-th row equals the earlier row 'powers[i]'.
>         for j in [ 1 .. i ] do
>           matrix[i][j]:= matrix[ powers[i] ][j];
>           matrix[j][i]:= matrix[j][ powers[i] ];
>         od;
>       else
>         # We have to compute the values.
>         g_i:= Representative( classes[i] );
>         nsg:= Filtered( [ 1 .. Length( normalsubgroups ) ],
>                         i -> g_i in normalsubgroups[i] );
>         normalsubgroupspos[i]:= nsg;
>         cents[i]:= Centralizer( G, g_i );
>         for j in [ 1 .. i ] do
>           g_j:= Representative( classes[j] );
>           if IsBound( powers[j] ) then
>             matrix[i][j]:= matrix[i][ powers[j] ];
>             matrix[j][i]:= matrix[ powers[j] ][i];
>           elif not IsEmpty( Intersection( nsg, normalsubgroupspos[j] ) )
```

```
>                or ( Order( g_i ) = 2 and Order( g_j ) = 2
>                     and not IsDihedralGroup( G ) ) then
>             matrix[i][j]:= 0;
>             matrix[j][i]:= 0;
>           else
>             # Compute $d(g_i, g_j^G)$.
>             gen:= 0;
>             for pair in DoubleCosetRepsAndSizes( G, cents[j],
>                             cents[i] ) do
>               if IsGeneratorsOfTransPermGroup( G,
>                     [ g_i, g_j^pair[1] ] ) then
>                 gen:= gen + pair[2];
>               fi;
>             od;
>             matrix[i][j]:= gen / Size( cents[j] );
>             if i <> j then
>               matrix[j][i]:= gen / Size( cents[i] );
>             fi;
>           fi;
>         od;
>
>         # For later, provide information about algebraic conjugacy.
>         for d in Difference( PrimeResidues( Order( g_i ) ), [ 1 ] ) do
>           pow:= g_i^d;
>           for j in [ i+1 .. nccl ] do
>             if not IsBound( powers[j] ) and pow in classes[j] then
>               powers[j]:= i;
>               break;
>             fi;
>           od;
>         od;
>       fi;
>     od;
>
>     return matrix;
> end );
```

4.3.2 Computing Lower Bounds for Vertex Degrees

In this section, the task is to compute the lower bounds $\delta(s, g^G)$ for the vertex degrees $d(s, g^G)$ using character-theoretic methods.

We provide GAP functions for computing the multiset Π of the primitive permutation characters of a given group G and for computing the lower bounds $\delta(s, g^G)$ from Π.

For many almost simple groups, the GAP libraries of character tables and of tables of marks contain information for quickly computing the primitive permutation characters of the group in question. Therefore, the function `PrimitivePermutationCharacters` takes as its argument not the group G but its character table T, say. (This function is shown already in [Breb].)

If T is contained in the GAP Character Table Library (see [Bre25]) then the complete set of primitive permutation characters can be easily computed if the character tables of all maximal subgroups and their class fusions into T are known (in this case, we check whether the attribute Maxes (**CTblLib:**

Maxes) of T is bound) or if the table of marks of G and the class fusion from T into this table of marks are known (in this case, we check whether the attribute `FusionToTom` (**CTblLib: FusionToTom**) of T is bound). If the attribute `UnderlyingGroup` (**Reference: UnderlyingGroup for tables of marks**) of T is bound then the group stored as the value of this attribute can be used to compute the primitive permutation characters. The latter happens if T was computed from the group G; for tables in the GAP Character Table Library, this is not the case by default.

The GAP function `PrimitivePermutationCharacters` tries to compute the primitive permutation characters of a group using this information; it returns the required list of characters if this can be computed this way, otherwise `fail` is returned. (For convenience, we use the GAP mechanism of *attributes* in order to store the permutation characters in the character table object once they have been computed.)

```
———————————————————————— Example ————————————————————————
gap> DeclareAttribute( "PrimitivePermutationCharacters",
>                      IsCharacterTable );
gap> InstallOtherMethod( PrimitivePermutationCharacters,
>      [ IsCharacterTable ],
>      function( tbl )
>      local maxes, i, fus, poss, tom, G;
>
>      if HasMaxes( tbl ) then
>        maxes:= List( Maxes( tbl ), CharacterTable );
>        for i in [ 1 .. Length( maxes ) ] do
>          fus:= GetFusionMap( maxes[i], tbl );
>          if fus = fail then
>            fus:= PossibleClassFusions( maxes[i], tbl );
>            poss:= Set( fus,
>              map -> InducedClassFunctionsByFusionMap(
>                        maxes[i], tbl,
>                        [ TrivialCharacter( maxes[i] ) ], map )[1] );
>            if Length( poss ) = 1 then
>              maxes[i]:= poss[1];
>            else
>              return fail;
>            fi;
>          else
>            maxes[i]:= TrivialCharacter( maxes[i] )^tbl;
>          fi;
>        od;
>        return maxes;
>      elif HasFusionToTom( tbl ) then
>        tom:= TableOfMarks( tbl );
>        maxes:= MaximalSubgroupsTom( tom );
>        return PermCharsTom( tbl, tom ){ maxes[1] };
>      elif HasUnderlyingGroup( tbl ) then
>        G:= UnderlyingGroup( tbl );
>        return List( MaximalSubgroupClassReps( G ),
>                     M -> TrivialCharacter( M )^tbl );
>      fi;
>
>      return fail;
> end );
```

The next function computes the lower bounds $\delta(s, g^G)$ from the two lists `classlengths` of conjugacy class lengths of the group G and `prim` of all primitive permutation characters of G. (The first entry in `classlengths` is assumed to represent the class containing the identity element of G.) The return value is the matrix that contains in row i and column j the value $\delta(s, g^G)$, where s and g are in the conjugacy classes represented by the $(i+1)$-st and $(j+1)$-st column of `tbl`, respectively. So the row sums of this matrix are the values $\delta(s)$.

```
                     ─────────── Example ───────────
  gap> LowerBoundsVertexDegrees:= function( classlengths, prim )
  >       local sizes, nccl;
  >
  >       nccl:= Length( classlengths );
  >       return List( [ 2 .. nccl ],
  >                i -> List( [ 2 .. nccl ],
  >                     j -> Maximum( 0, classlengths[j] - Sum( prim,
  >                          pi -> classlengths[j] * pi[j] * pi[i]
  >                                   / pi[1] ) ) ) );
  > end;;
```

4.3.3   Evaluating the (Lower Bounds for the) Vertex Degrees

In this section, the task is to compute (lower bounds for) the vertex degrees of iterated closures of a generating graph from (lower bounds for) the vertex degrees of the graph itself, and then to check the criteria of Pósa and Chvátal.

The arguments of all functions defined in this section are the list `classlengths` of conjugacy class lengths for the group G (including the class of the identity element, in the first position) and a matrix bounds of the values $d^{(i)}(s, g^G)$ or $\delta^{(i)}(s, g^G)$, with rows and columns indexed by nonidentity class representatives s and g, respectively. Such a matrix is returned by the functions `VertexDegreesGeneratingGraph` or `LowerBoundsVertexDegrees`, respectively.

The function `LowerBoundsVertexDegreesOfClosure` returns the corresponding matrix of the values $d^{(i+1)}(s, g^G)$ or $\delta^{(i+1)}(s, g^G)$, respectively.

```
                     ─────────── Example ───────────
  gap> LowerBoundsVertexDegreesOfClosure:= function( classlengths, bounds )
  >       local delta, newbounds, size, i, j;
  >
  >       delta:= List( bounds, Sum );
  >       newbounds:= List( bounds, ShallowCopy );
  >       size:= Sum( classlengths );
  >       for i in [ 1 .. Length( bounds ) ] do
  >         for j in [ 1 .. Length( bounds ) ] do
  >           if delta[i] + delta[j] >= size - 1 then
  >             newbounds[i][j]:= classlengths[ j+1 ];
  >           fi;
  >         od;
  >       od;
  >
  >       return newbounds;
  > end;;
```

Once the values $d^{(i)}(s, g^G)$ or $\delta^{(i)}(s, g^G)$ are known, we can check whether Pósa's or Chvátal's criterion is satisfied for the graph $\mathrm{cl}^{(i)}(\Gamma(G))$, using the function `CheckCriteriaOfPosaAndChvatal`

shown below. (Of course a *negative* result is meaningless in the case that only lower bounds for the vertex degrees are used.)

The idea is to compute the row sums of the given matrix, and to compute the intervals $\{L_k, L_k + 1, \ldots, U_k\}$ and $\{L'_k, L'_k + 1, \ldots, U'_k\}$ that were introduced in Section 4.2.2.

The function `CheckCriteriaOfPosaAndChvatal` returns, given the list of class lengths of G and the matrix of (bounds for the) vertex degrees, a record with the components `badForPosa` (the list of those pairs $[L_k, U_k]$ with the property $L_k \leq U_k$), `badForChvatal` (the list of pairs of lower and upper bounds of nonempty intervals where Chvátal's criterion may be violated), and `data` (the sorted list of triples $[\delta(g_k), |g_k^G|, \iota(k)]$, where $\iota(k)$ is the row and column position of g_k in the matrix bounds). The ordering of class lengths must of course be compatible with the ordering of rows and columns of the matrix, and the identity element of G must belong to the first entry in the list of class lengths.

```
——————————————— Example ———————————————
gap> CheckCriteriaOfPosaAndChvatal:= function( classlengths, bounds )
>       local size, degs, addinterval, badForPosa, badForChvatal1, pos,
>             half, i, low1, upp2, upp1, low2, badForChvatal, interval1,
>             interval2;
>
>       size:= Sum( classlengths );
>       degs:= List( [ 2 .. Length( classlengths ) ],
>                    i -> [ Sum( bounds[ i-1 ] ), classlengths[i], i ] );
>       Sort( degs );
>
>       addinterval:= function( intervals, low, upp )
>         if low <= upp then
>           Add( intervals, [ low, upp ] );
>         fi;
>       end;
>
>       badForPosa:= [];
>       badForChvatal1:= [];
>       pos:= 1;
>       half:= Int( size / 2 ) - 1;
>       for i in [ 1 .. Length( degs ) ] do
>         # We have pos = c_1 + c_2 + \cdots + c_{i-1} + 1
>         low1:= Maximum( pos, degs[i][1] );  # L_i
>         upp2:= Minimum( half, size-1-pos, size-1-degs[i][1] ); # U'_i
>         pos:= pos + degs[i][2];
>         upp1:= Minimum( half, pos-1 ); # U_i
>         low2:= Maximum( 1, size-pos ); # L'_i
>         addinterval( badForPosa, low1, upp1 );
>         addinterval( badForChvatal1, low2, upp2 );
>       od;
>
>       # Intersect intervals.
>       badForChvatal:= [];
>       for interval1 in badForPosa do
>         for interval2 in badForChvatal1 do
>           addinterval( badForChvatal,
>                        Maximum( interval1[1], interval2[1] ),
>                        Minimum( interval1[2], interval2[2] ) );
>         od;
```

```
>        od;
>
>        return rec( badForPosa:= badForPosa,
>                    badForChvatal:= Set( badForChvatal ),
>                    data:= degs );
> end;;
```

Finally, the function `HamiltonianCycleInfo` assumes that the matrix bounds contains lower bounds for the vertex degrees in the generating graph Γ, and returns a string that describes the minimal i with the property that the given bounds suffice to show that $cl^{(i)}(\Gamma)$ satisfies Pósa's or Chvátal's criterion, if such a closure exists. If no closure has this property, the string `"no decision"` is returned.

```
————————————————— Example —————————————————
gap> HamiltonianCycleInfo:= function( classlengths, bounds )
>       local i, result, res, oldbounds;
>
>       i:= 0;
>       result:= rec( Posa:= fail, Chvatal:= fail );
>       repeat
>         res:= CheckCriteriaOfPosaAndChvatal( classlengths, bounds );
>         if result.Posa = fail and IsEmpty( res.badForPosa ) then
>           result.Posa:= i;
>         fi;
>         if result.Chvatal = fail and IsEmpty( res.badForChvatal ) then
>           result.Chvatal:= i;
>         fi;
>         i:= i+1;
>         oldbounds:= bounds;
>         bounds:= LowerBoundsVertexDegreesOfClosure( classlengths,
>                   bounds );
>       until oldbounds = bounds;
>
>       if result.Posa <> fail then
>         if result.Posa <> result.Chvatal then
>           return Concatenation(
>               "Chvatal for ", Ordinal( result.Chvatal ), " closure, ",
>               "Posa for ", Ordinal( result.Posa ), " closure" );
>         else
>           return Concatenation( "Posa for ", Ordinal( result.Posa ),
>               " closure" );
>         fi;
>       elif result.Chvatal <> fail then
>         return Concatenation( "Chvatal for ", Ordinal( result.Chvatal ),
>                           " closure" );
>       else
>         return "no decision";
>       fi;
> end;;
```

4.4   Character-Theoretic Computations

In this section, we apply the functions introduced in Section 4.3 to character tables of almost simple groups that are available in the GAP Character Table Library.

Our first examples are the sporadic simple groups, in Section 4.4.1, then their automorphism groups are considered in Section 4.4.3. Small alternating and symmetric groups are treated in Section 4.4.4.

For our convenience, we provide a small function that takes as its argument only the character table in question, and returns a string, either `"no prim. perm. characters"` or the return value of `HamiltonianCycleInfo` for the bounds computed from the primitive permutation characters.

```
──────────────────────────── Example ────────────────────────────
gap> HamiltonianCycleInfoFromCharacterTable:= function( tbl )
>     local prim, classlengths, bounds;
>
>     prim:= PrimitivePermutationCharacters( tbl );
>     if prim = fail then
>       return "no prim. perm. characters";
>     fi;
>     classlengths:= SizesConjugacyClasses( tbl );
>     bounds:= LowerBoundsVertexDegrees( classlengths, prim );
>     return HamiltonianCycleInfo( classlengths, bounds );
> end;;
```

4.4.1   Sporadic Simple Groups

Since version 1.3.10 of CTblLib, the GAP Character Table Library contains the tables of maximal subgroups of all sporadic simple groups.

So the function `PrimitivePermutationCharacters` can be used to compute all their primitive permutation characters.

```
──────────────────────────── Example ────────────────────────────
gap> spornames:= AllCharacterTableNames( IsSporadicSimple, true,
>                   IsDuplicateTable, false );
[ "B", "Co1", "Co2", "Co3", "F3+", "Fi22", "Fi23", "HN", "HS", "He",
  "J1", "J2", "J3", "J4", "Ly", "M", "M11", "M12", "M22", "M23",
  "M24", "McL", "ON", "Ru", "Suz", "Th" ]
gap> for tbl in List( spornames, CharacterTable ) do
>     info:= HamiltonianCycleInfoFromCharacterTable( tbl );
>     if info <> "Posa for 0th closure" then
>       Print( Identifier( tbl ), ": ", info, "\n" );
>     fi;
>   od;
```

It turns out that the information available in the GAP Character Table Library is sufficient to prove that the generating graph contains a Hamiltonian cycle.

4.4.2   The Monster

In the original version of this file, a special treatment of the Monster group had been necessary because not all character tabes of its maximal subgroups were available, and also not all class fusions from

some maximal subgroups were known whose character tables were available. This section shows the computations that were used at that time.

Meanwhile the maximal subgroups of the Monster are classified, see [DLP25], their character tables are available, and also all class fusions from the maximal subgroups are known. Thus we can easily compute the primitive permutation characters of the Monster group. The computations shown in this section are not needed to prove the claims from [BGL$^+$10], they are kept just for historical reasons. Thus this section is an example how the availability of the full list of primitive permutation characters of the Monster simplifies proofs.

The situation for the following arguments is as follows. We do not assume the knowledge of all primitive permutation characters of the Monster group M, and we use only some of the character tables of the maximal subgroups of M and only partial information on their class fusions. Note that for some maximal subgroups of M, the permutation character is not uniquely determined by the character tables involved. However, we can compute upper bounds for the values of the primitive permutation characters 1_S^M from the possible class fusions from S into M if the character table of S is known. For the other subgroups S, the permutation characters 1_S^M have been computed with other methods. Using this information, we will show that the generating graph of M satisfies Pósa's criterion.

The list `primdata` defined below has length 46. The entry at position i is a list of length one or two. If `primdata[i]` has length one then its unique entry is the identifier of the library character table of the i-th maximal subgroup of M. If `primdata[i]` has length two then its entries are a string describing the structure of the i-th maximal subgroup S of M and the permutation character 1_S^M.

(The construction of the explicitly given characters in this list will be documented elsewhere. Some of the constructions can be found in Section 8.16.)

```
gap> dir:= DirectoriesPackageLibrary( "ctbllib", "data" );;
gap> filename:= Filename( dir, "prim_perm_M.json" );;
gap> primdata:= EvalString( StringFile( filename ) )[2];;
gap> Length( primdata );
46
gap> m:= CharacterTable( "M" );;
```

We compute upper bounds for the permutation character values in the cases where the characters are not given explicitly. (We could improve this by using additional information about the class fusions, but this will not be necessary.)

```
gap> s:= "dummy";;      #  Avoid a message about an unbound variable ...
gap> poss:= "dummy";;   #  Avoid a message about an unbound variable ...
gap> for entry in primdata do
>      if not IsBound( entry[2] ) then
>        s:= CharacterTable( entry[1] );
>        poss:= Set( PossibleClassFusions( s, m ),
>                  x -> InducedClassFunctionsByFusionMap( s, m,
>                         [ TrivialCharacter( s ) ], x )[1] );
>        entry[2]:= List( [ 1 .. NrConjugacyClasses( m ) ],
>                    i -> Maximum( List( poss, x -> x[i] ) ) );
>      fi;
>    od;
```

Now we estimate the lower bounds $\delta(s, g^G)$ introduced in Section 4.3.2. Let $\mathbb{M}$ denote a set of

representatives of the classes of maximal subgroups of *M*. Then

$$\delta(s, g^G) = |s^G| - |s^G| \cdot \sum_{S \in \mathbb{M}} 1_S^M(s) \cdot 1_S^M(g) / 1_S^M(1),$$

hence $\delta(s)$ can be computed from the corresponding primitive permutation characters, and a lower bound for $\delta(s)$ can be computed from the upper bounds for the characters 1_S^G which are given by the list `primdata`.

This means that modifying the output of `LowerBoundsVertexDegrees` as follows really yields lower bounds for the vertex degrees.

```
——————————————————— Example ———————————————————
gap> prim:= List( primdata, x -> x[2] );;
gap> classlengths:= SizesConjugacyClasses( m );;
gap> bounds:= LowerBoundsVertexDegrees( classlengths, prim );;
```

Now we sum up the bounds for the individual classes. It turns out that the minimal vertex degree is more than 99.99998 percent of $|M|$. This proves that the generating graph of the Monster satisfies Pósa's criterion.

```
——————————————————— Example ———————————————————
gap> degs:= List( bounds, Sum );;
gap> Int( 100000000 * Minimum( degs ) / Size( m ) );
99999987
```

Without the results from [DLP25], we can argue as follows. (This was the situation in earlier versions of this example file.)

According to [NW13], any maximal subgroup of the Monster is either among the 44 known classes from the above list except $L_2(13).2$ and $U_3(4).4$, or it is an almost simple group whose socle is one of $L_2(13)$, $Sz(8)$, $U_3(4)$, and $U_3(8)$.

We show that the elements of such subgroups are contained in the union of 55 conjugacy classes of the Monster that cover less than one percent of the elements in the Monster. For that, we compute the possible class fusions from the abovementioned simple groups *S* into the Monster, and then the possible class fusions from the automorphic extensions of *S* into the Monster, using the possible class fusions of *S*. (This approach is faster than computing each class fusion from scratch.)

After the following computations, the list `badclasses` will contain the positions of all those classes of *M* that may contain elements in some of the hypothetical maximal subgroups.

For each simple group in question, we enter the identifiers of the character tables of the automorphic extensions that can occur. Note that the automorphism groups of the four groups have the structures $L_2(13).2$, $Sz(8).3$, $U_3(4).4$, and $U_3(8).(3 \times S_3)$, respectively. We need not consider the groups $U_3(8).3^2$ and $U_3(8).(3 \times S_3)$ because already $U_3(8).3_2$ does not admit an embedding into *M*, and we need not consider the group $U_3(8).S_3$ because its set of elements is covered by its subgroups of the types $U_3(8).2$ and $U_3(8).3_2$.

```
——————————————————— Example ———————————————————
gap> PossibleClassFusions( CharacterTable( "U3(8).3_2" ), m );
[  ]
gap> badclasses:= [];;
gap> names:= [
>      [ "L2(13)", "L2(13).2" ],
>      [ "Sz(8)", "Sz(8).3" ],
```

```
>        [ "U3(4)", "U3(4).2", "U3(4).4" ],
>        [ "U3(8)", "U3(8).2", "U3(8).3_1", "U3(8).3_2", "U3(8).3_3",
>                "U3(8).6" ],
>        ];;
gap> for list in names do
>        t:= CharacterTable( list[1] );
>        tfusm:= PossibleClassFusions( t, m );
>        UniteSet( badclasses, Flat( tfusm ) );
>        for nam in list{ [ 2 .. Length( list ) ] } do
>          ext:= CharacterTable( nam );
>          for map1 in PossibleClassFusions( t, ext ) do
>            inv:= InverseMap( map1 );
>            for map2 in tfusm do
>              init:= CompositionMaps( map2, inv );
>              UniteSet( badclasses, Flat( PossibleClassFusions( ext, m,
>                rec( fusionmap:= init ) ) ) );
>            od;
>          od;
>        od;
>      od;
gap> badclasses;
[ 1, 3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 17, 18, 19, 20, 21, 22,
  24, 25, 27, 28, 30, 32, 33, 35, 36, 38, 39, 40, 42, 43, 44, 45, 46,
  48, 49, 50, 51, 52, 53, 54, 55, 56, 60, 61, 62, 63, 70, 72, 73, 78,
  82, 85, 86 ]
gap> Length( badclasses );
55
gap> bad:= Sum( classlengths{ badclasses } ) / Size( m );;
gap> Int( 10000 * bad );
97
```

In the original version of this file, also hypothetical maximal subgroups with socle $L_2(27)$ *had been considered. As a consequence, the list* badclasses *computed above had length* 59 *in the original version; the list contained also the classes at the positions* 90, 94, 95, *and* 96, *that is, the classes* 26B, 28B, 28C, 28D. *The proportion* bad *of elements in the classes of M described by* badclasses *was about* 2.05 *percent of* |M|, *compared to the about* 0.98 *percent in the current version.*

Now we estimate the lower bounds $\delta(s, g^G)$ introduced in Section 4.3.2. Let $\mathscr{B}$ denote the union of the classes described by badclasses, and let $\mathbb{M}$ denote a set of representatives of the 44 known classes of maximal subgroups of M.

If $s \notin \mathscr{B}$ then

$$\delta(s, g^G) = |s^G| - |s^G| \cdot \sum_{S \in \mathbb{M}} 1_S^M(s) \cdot 1_S^M(g) / 1_S^M(1),$$

hence $\delta(s)$ can be computed from the corresponding primitive permutation characters, and a lower bound for $\delta(s)$ can be computed from the upper bounds for the characters 1_S^G which are given by the list primdata.

If $s \in \mathscr{B}$ then the above equation for $\delta(s, g^G)$ holds at least for $g \notin \mathscr{B}$, so $\sum_{g \in R \setminus \mathscr{B}} \delta(s, g^G)$ is a lower bound for $\delta(s)$. So primdata yields a lower bound for $\delta(s)$ also for $s \in \mathscr{B}$, by ignoring the pairs (s, g) where both s and g lie in $\mathscr{B}$.

This means that modifying the output of LowerBoundsVertexDegrees as follows really yields lower bounds for the vertex degrees. (Note that the row and column positions in the matrix returned

by `LowerBoundsVertexDegrees` are shifted by one, compared to `badclasses`.)

```
——————————————— Example ———————————————
gap> prim:= List( primdata, x -> x[2] );;
gap> badpos:= Difference( badclasses, [ 1 ] ) - 1;;
gap> bounds:= LowerBoundsVertexDegrees( classlengths, prim );;
gap> for i in badpos do
>        for j in badpos do
>          bounds[i][j]:= 0;
>        od;
>      od;
```

Now we sum up the bounds for the individual classes. It turns out that the minimal vertex degree is more than 99 percent of $|M|$. This proves that the generating graph of the Monster satisfies Pósa's criterion.

(And the minimal vertex degree of elements outside $\mathscr{B}$ is more than 99.99998 percent of $|M|$.)

In the original version of this file, we got only 97.95 percent of $|M|$ as the lower bound for the minimal vertex degree. The bound for elements outside $\mathscr{B}$ was the same in the original version. The fact that the maximal subgroups of type $L_2(41)$ had been ignored in the original version did not affect the lower bound for the minimal vertex degree.

```
——————————————— Example ———————————————
gap> degs:= List( bounds, Sum );;
gap> Int( 10000 * Minimum( degs ) / Size( m ) );
9902
gap> goodpos:= Difference( [ 1 .. NrConjugacyClasses( m ) - 1 ],
>                          badpos );;
gap> Int( 100000000 * Minimum( degs{ goodpos } ) / Size( m ) );
99999987
```

4.4.3   Nonsimple Automorphism Groups of Sporadic Simple Groups

Next we consider the nonsimple automorphism groups of the sporadic simple groups. Nontrivial outer automorphisms exist exactly in 12 cases, and then the simple group has index 2 in its automorphism group. The character tables of the groups and their maximal subgroups are available in GAP.

```
——————————————— Example ———————————————
gap> spornames:= AllCharacterTableNames( IsSporadicSimple, true,
>                  IsDuplicateTable, false );;
gap> sporautnames:= AllCharacterTableNames( IsSporadicSimple, true,
>                    IsDuplicateTable, false,
>                    OfThose, AutomorphismGroup );;
gap> sporautnames:= Difference( sporautnames, spornames );
[ "F3+.2", "Fi22.2", "HN.2", "HS.2", "He.2", "J2.2", "J3.2", "M12.2",
  "M22.2", "McL.2", "ON.2", "Suz.2" ]
gap> for tbl in List( sporautnames, CharacterTable ) do
>        info:= HamiltonianCycleInfoFromCharacterTable( tbl );
>        Print( Identifier( tbl ), ": ", info, "\n" );
>      od;
F3+.2: Chvatal for 0th closure, Posa for 1st closure
Fi22.2: Chvatal for 0th closure, Posa for 1st closure
HN.2: Chvatal for 0th closure, Posa for 1st closure
HS.2: Chvatal for 1st closure, Posa for 2nd closure
```

```
He.2: Chvatal for 0th closure, Posa for 1st closure
J2.2: Chvatal for 0th closure, Posa for 1st closure
J3.2: Chvatal for 0th closure, Posa for 1st closure
M12.2: Chvatal for 0th closure, Posa for 1st closure
M22.2: Posa for 1st closure
McL.2: Chvatal for 0th closure, Posa for 1st closure
ON.2: Chvatal for 0th closure, Posa for 1st closure
Suz.2: Chvatal for 0th closure, Posa for 1st closure
```

4.4.4 Alternating and Symmetric Groups A_n, S_n, for $5 \leq n \leq 13$

For alternating and symmetric groups A_n and S_n, respectively, with $5 \leq n \leq 13$, the table of marks or the character tables of the group and all its maximal subgroups are available in GAP. So we can compute the character–theoretic bounds for vertex degrees.

```
——————————————— Example ———————————————
gap> for tbl in List( [ 5 .. 13 ], i -> CharacterTable(
>                 Concatenation( "A", String( i ) ) ) )  do
>       info:= HamiltonianCycleInfoFromCharacterTable( tbl );
>       if info <> "Posa for 0th closure" then
>         Print( Identifier( tbl ), ": ", info, "\n" );
>       fi;
>    od;
```

No messages are printed, so the generating graphs of the alternating groups in question satisfy Pósa's criterion.

```
——————————————— Example ———————————————
gap> for tbl in List( [ 5 .. 13 ], i -> CharacterTable(
>                 Concatenation( "S", String( i ) ) ) )  do
>       info:= HamiltonianCycleInfoFromCharacterTable( tbl );
>       Print( Identifier( tbl ), ": ", info, "\n" );
>    od;
A5.2: no decision
A6.2_1: Chvatal for 4th closure, Posa for 5th closure
A7.2: Posa for 1st closure
A8.2: Chvatal for 2nd closure, Posa for 3rd closure
A9.2: Chvatal for 2nd closure, Posa for 3rd closure
A10.2: Chvatal for 2nd closure, Posa for 3rd closure
A11.2: Posa for 1st closure
A12.2: Chvatal for 2nd closure, Posa for 3rd closure
A13.2: Posa for 1st closure
```

We see that sufficiently large closures of the generating graphs of the symmetric groups in question satisfy Pósa's criterion, except that the bounds for the symmetric group S_5 are not sufficient for the proof. In Section 4.5.2, it is shown that the 2nd closure of the generating graph of S_5 satisfies Pósa's criterion.

(We could find slightly better bounds derived only from character tables which suffice to prove that the generating graph for S_5 contains a Hamiltonian cycle, but this seems to be not worth while.)

4.5   Computations With Groups

We prove in Section 4.5.1 that the generating graphs of the nonabelian simple groups of order up to 10^6 satisfy Pósa's criterion, and that the same holds for those nonabelian simple groups of order between 10^6 and 10^7 that are not isomorphic with some $PSL(2, q)$. (In Section 4.6, it is shown that the generating graph of $PSL(2, q)$ satifies Pósa's criterion for any prime power q.) Nonsimple nonsolvable groups of order up to 10^6 are treated in Section 4.5.2.

(We could increase the bounds 10^6 and 10^7 with more computations, using the same methods.)

For our convenience, we provide a small function that takes as its argument only the group in question, and returns a string, the return value of `HamiltonianCycleInfo` for the vertex degrees computed from the group. (In order to speed up the computations, the function computes the proper normal subgroups that contain the derived subgroup of the given group, and enters the list of these groups as the third argument of `VertexDegreesGeneratingGraph`.)

```
———— Example ————
gap> HamiltonianCycleInfoFromGroup:= function( G )
>     local ccl, nsg, der, degrees, classlengths;
>     ccl:= ConjugacyClasses( G );
>     if IsPerfect( G ) then
>       nsg:= [];
>     else
>       der:= DerivedSubgroup( G );
>       nsg:= Concatenation( [ der ],
>                 IntermediateSubgroups( G, der ).subgroups );
>     fi;
>     degrees:= VertexDegreesGeneratingGraph( G, ccl, nsg );
>     classlengths:= List( ccl, Size );
>     return HamiltonianCycleInfo( classlengths, degrees );
> end;;
```

4.5.1   Nonabelian Simple Groups of Order up to 10^7

Representatives of the 56 isomorphism types of nonabelian simple groups of order up to 10^6 can be accessed in GAP with the function `AllSmallNonabelianSimpleGroups`.

```
———— Example ————
gap> grps:= AllSmallNonabelianSimpleGroups( [ 1 .. 10^6 ] );;
gap> Length( grps );
56
gap> List( grps, StructureDescription );
[ "A5", "PSL(3,2)", "A6", "PSL(2,8)", "PSL(2,11)", "PSL(2,13)",
  "PSL(2,17)", "A7", "PSL(2,19)", "PSL(2,16)", "PSL(3,3)",
  "PSU(3,3)", "PSL(2,23)", "PSL(2,25)", "M11", "PSL(2,27)",
  "PSL(2,29)", "PSL(2,31)", "A8", "PSL(3,4)", "PSL(2,37)", "O(5,3)",
  "Sz(8)", "PSL(2,32)", "PSL(2,41)", "PSL(2,43)", "PSL(2,47)",
  "PSL(2,49)", "PSU(3,4)", "PSL(2,53)", "M12", "PSL(2,59)",
  "PSL(2,61)", "PSU(3,5)", "PSL(2,67)", "J1", "PSL(2,71)", "A9",
  "PSL(2,73)", "PSL(2,79)", "PSL(2,64)", "PSL(2,81)", "PSL(2,83)",
  "PSL(2,89)", "PSL(3,5)", "M22", "PSL(2,97)", "PSL(2,101)",
  "PSL(2,103)", "HJ", "PSL(2,107)", "PSL(2,109)", "PSL(2,113)",
  "PSL(2,121)", "PSL(2,125)", "O(5,4)" ]
gap> for g in grps do
```

```
>         info:= HamiltonianCycleInfoFromGroup( g );
>         if info <> "Posa for 0th closure" then
>           Print( StructureDescription( g ), ": ", info, "\n" );
>         fi;
>     od;
```

Nothing is printed during these computations, so the generating graphs of all processed groups satisfy Pósa's criterion.

(On my notebook, the above computations needed about 6300 seconds of CPU time.)

For simple groups of order larger than 10^6, there is not such an easy way (yet) to access representatives for each isomorphism type. Therefore, first we compute the orders of nonabelian simple groups between 10^6 and 10^7.

```
────────────────────────── Example ──────────────────────────
gap> orders:= Filtered( [ 10^6+4, 10^6+8 .. 10^7 ],
>         n -> IsomorphismTypeInfoFiniteSimpleGroup( n ) <> fail );
[ 1024128, 1123980, 1285608, 1342740, 1451520, 1653900, 1721400,
  1814400, 1876896, 1934868, 2097024, 2165292, 2328648, 2413320,
  2588772, 2867580, 2964780, 3265920, 3483840, 3594432, 3822588,
  3940200, 4245696, 4680000, 4696860, 5515776, 5544672, 5663616,
  5848428, 6004380, 6065280, 6324552, 6825840, 6998640, 7174332,
  7906500, 8487168, 9095592, 9732420, 9951120, 9999360 ]
gap> Length( orders );
41
gap> info:= List( orders, IsomorphismTypeInfoFiniteSimpleGroup );;
gap> Number( info, x -> IsBound( x.series ) and x.series = "L"
>                       and x.parameter[1] = 2 );
31
```

We see that there are 31 groups of the type $\mathrm{PSL}(2,q)$ and 10 other nonabelian simple groups with order in the range from 10^6 to 10^7. The former groups can be ignored because the generating graphs of any group $\mathrm{PSL}(2,q)$ satisfies Pósa's criterion, see Section 4.6. For the latter groups, we can apply the character–theoretic method to prove that the generating graph satisfies Pósa's criterion.

```
────────────────────────── Example ──────────────────────────
gap> info:= Filtered( info, x -> not IsBound( x.series ) or
>               x.series <> "L" or x.parameter[1] <> 2 );
[ rec( name := "B(3,2) = O(7,2) ~ C(3,2) = S(6,2)",
      parameter := [ 3, 2 ], series := "B", shortname := "S6(2)" ),
  rec( name := "A(10)", parameter := 10, series := "A",
      shortname := "A10" ),
  rec( name := "A(2,7) = L(3,7) ", parameter := [ 3, 7 ],
      series := "L", shortname := "L3(7)" ),
  rec( name := "2A(3,3) = U(4,3) ~ 2D(3,3) = O-(6,3)",
      parameter := [ 3, 3 ], series := "2A", shortname := "U4(3)" ),
  rec( name := "G(2,3)", parameter := 3, series := "G",
      shortname := "G2(3)" ),
  rec( name := "B(2,5) = O(5,5) ~ C(2,5) = S(4,5)",
      parameter := [ 2, 5 ], series := "B", shortname := "S4(5)" ),
  rec( name := "2A(2,8) = U(3,8)", parameter := [ 2, 8 ],
      series := "2A", shortname := "U3(8)" ),
  rec( name := "2A(2,7) = U(3,7)", parameter := [ 2, 7 ],
```

```
         series := "2A", shortname := "U3(7)" ),
    rec( name := "A(3,3) = L(4,3) ~ D(3,3) = O+(6,3) ",
         parameter := [ 4, 3 ], series := "L", shortname := "L4(3)" ),
    rec( name := "A(4,2) = L(5,2) ", parameter := [ 5, 2 ],
         series := "L", shortname := "L5(2)" ) ]
gap> names:= [ "S6(2)", "A10", "L3(7)", "U4(3)", "G2(3)", "S4(5)",
>              "U3(8)", "U3(7)", "L4(3)", "L5(2)" ];;
gap> for tbl in List( names, CharacterTable ) do
>       info:= HamiltonianCycleInfoFromCharacterTable( tbl );
>       if info <> "Posa for 0th closure" then
>         Print( Identifier( tbl ), ": ", info, "\n" );
>       fi;
>     od;
```

4.5.2 Nonsimple Groups with Nonsolvable Socle of Order at most 10^6

Let G be a nonsolvable group such that G/N is cyclic for all nontrivial normal subgroups N of G. Then the socle $\mathrm{Soc}(G)$ of G is the unique minimal normal subgroup. Moreover, $\mathrm{Soc}(G)$ is nonsolvable and thus a direct product of isomorphic nonabelian simple groups, and G is isomorphic to a subgroup of $\mathrm{Aut}(\mathrm{Soc}(G))$.

In order to deal with all such groups G for which additionally $|\mathrm{Soc}(G)| \leq 10^6$ holds, it is sufficient to run over the simple groups S of order up to 10^6 and to consider those subgroups G of $\mathrm{Aut}(S^n)$, with $|S|^n \leq 10^6$, for which $\mathrm{Inn}(G)$ is the unique minimal normal subgroup and $G/\mathrm{Inn}(G)$ is cyclic.

We show that for each such group, a sufficient closure of the generating graph satisfies Pósa's criterion.

```
─────────────────────── Example ───────────────────────
gap> grps:= AllSmallNonabelianSimpleGroups( [ 1 .. 10^6 ] );;
gap> epi:= "dummy";;   # Avoid a message about an unbound variable ...
gap> for simple in grps do
>      for n in [ 1 .. LogInt( 10^6, Size( simple ) ) ] do
>        # Compute the n-fold direct product S^n.
>        soc:= CallFuncList( DirectProduct,
>                            ListWithIdenticalEntries( n, simple ) );
>        # Compute Aut(S^n) as a permutation group.
>        aut:= Image( IsomorphismPermGroup( AutomorphismGroup( soc ) ) );
>        aut:= Image( SmallerDegreePermutationRepresentation( aut ) );
>        # Compute class representatives of subgroups of
>        # Aut(S^n)/Inn(S^n).
>        socle:= Socle( aut );
>        epi:= NaturalHomomorphismByNormalSubgroup( aut, socle );
>        # Compute the candidates for G.  (By the above computations,
>        # we need not consider simple groups.)
>        reps:= List( ConjugacyClassesSubgroups( Image( epi ) ),
>                     Representative );
>        reps:= Filtered( reps, x -> IsCyclic( x ) and Size( x ) <> 1 );
>        greps:= Filtered( List( reps, x -> PreImages( epi, x ) ),
>                          x -> Length( MinimalNormalSubgroups( x ) ) = 1 );
>        for g in greps do
>          # We have to deal with a *transitive* permutation group.
>          # (Each group in question acts faithfully on an orbit.)
>          if not IsTransitive( g ) then
```

```
>               g:= First( List( Orbits( g, MovedPoints( g ) ),
>                               x -> Action( g, x ) ),
>                       x -> Size( x ) = Size( g ) );
>           fi;
>           # Check this group G.
>           info:= HamiltonianCycleInfoFromGroup( g );
>           Print( Name( simple ), "^", n, ".", Size( g ) / Size( soc ),
>                   ": ", info, "\n" );
>         od;
>       od;
>     od;
A5^1.2: Posa for 2nd closure
A5^2.2: Posa for 0th closure
A5^2.4: Posa for 0th closure
A5^3.3: Posa for 0th closure
A5^3.6: Chvatal for 1st closure, Posa for 2nd closure
PSL(2,7)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,7)^2.2: Posa for 0th closure
PSL(2,7)^2.4: Posa for 0th closure
A6^1.2: Chvatal for 0th closure, Posa for 1st closure
A6^1.2: Chvatal for 4th closure, Posa for 5th closure
A6^1.2: Chvatal for 0th closure, Posa for 1st closure
A6^2.2: Posa for 0th closure
A6^2.4: Posa for 0th closure
A6^2.4: Posa for 0th closure
A6^2.4: Posa for 0th closure
PSL(2,8)^1.3: Posa for 0th closure
PSL(2,8)^2.2: Posa for 0th closure
PSL(2,8)^2.6: Chvatal for 0th closure, Posa for 1st closure
PSL(2,11)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,11)^2.2: Posa for 0th closure
PSL(2,11)^2.4: Posa for 0th closure
PSL(2,13)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,17)^1.2: Chvatal for 0th closure, Posa for 1st closure
A7^1.2: Posa for 1st closure
PSL(2,19)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,16)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,16)^1.4: Chvatal for 0th closure, Posa for 1st closure
PSL(3,3)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSU(3,3)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,23)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,25)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,25)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,25)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,27)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,27)^1.3: Posa for 0th closure
PSL(2,27)^1.6: Chvatal for 0th closure, Posa for 1st closure
PSL(2,29)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,31)^1.2: Chvatal for 0th closure, Posa for 1st closure
A8^1.2: Chvatal for 2nd closure, Posa for 3rd closure
PSL(3,4)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(3,4)^1.2: Chvatal for 1st closure, Posa for 2nd closure
```

```
PSL(3,4)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(3,4)^1.3: Posa for 0th closure
PSL(3,4)^1.6: Chvatal for 0th closure, Posa for 1st closure
PSL(2,37)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSp(4,3)^1.2: Chvatal for 1st closure, Posa for 2nd closure
Sz(8)^1.3: Posa for 0th closure
PSL(2,32)^1.5: Posa for 0th closure
PSL(2,41)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,43)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,47)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,49)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,49)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,49)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSU(3,4)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSU(3,4)^1.4: Chvatal for 0th closure, Posa for 1st closure
PSL(2,53)^1.2: Chvatal for 0th closure, Posa for 1st closure
M12^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,59)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,61)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSU(3,5)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSU(3,5)^1.3: Posa for 0th closure
PSL(2,67)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,71)^1.2: Chvatal for 0th closure, Posa for 1st closure
A9^1.2: Chvatal for 2nd closure, Posa for 3rd closure
PSL(2,73)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,79)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,64)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,64)^1.3: Posa for 0th closure
PSL(2,64)^1.6: Chvatal for 0th closure, Posa for 1st closure
PSL(2,81)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,81)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,81)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,81)^1.4: Chvatal for 0th closure, Posa for 1st closure
PSL(2,81)^1.4: Chvatal for 0th closure, Posa for 1st closure
PSL(2,83)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,89)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(3,5)^1.2: Chvatal for 0th closure, Posa for 1st closure
M22^1.2: Posa for 1st closure
PSL(2,97)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,101)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,103)^1.2: Chvatal for 0th closure, Posa for 1st closure
J_2^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,107)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,109)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,113)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,121)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,121)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,121)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,125)^1.2: Chvatal for 0th closure, Posa for 1st closure
PSL(2,125)^1.3: Posa for 0th closure
PSL(2,125)^1.6: Chvatal for 0th closure, Posa for 1st closure
PSp(4,4)^1.2: Chvatal for 0th closure, Posa for 1st closure
```

```
PSp(4,4)^1.4: Posa for 0th closure
```

4.6 The Groups $\mathrm{PSL}(2,q)$

We show that the generating graph of any group $\mathrm{PSL}(2,q)$, for $q \geq 2$, satisfies Pósa's criterion. Throughout this section, let $q = p^f$ for a prime integer p, and $G = \mathrm{PSL}(2,q)$. Set $k = \gcd(q-1,2)$.

Lemma 1: (see [Hup67, II., § 8]) The subgroups of G are

(1)   cyclic groups of order dividing $(q \pm 1)/k$, and their normalizers, which are dihedral groups of order $2(q \pm 1)/k$,

(2)   subgroups of Sylow p normalizers, which are semidirect products of elementary abelian groups of order q with cyclic groups of order $(q-1)/k$,

(3)   subgroups isomorphic with $\mathrm{PSL}(2,p^m)$ if m divides f, and isomorphic with $\mathrm{PGL}(2,p^m)$ if $2m$ divides f,

(4)   subgroups isomorphic with A_4, S_4, or A_5, for appropriate values of q.

G contains exactly one conjugacy class of cyclic subgroups of each of the orders $(q-1)/k$ and $(q+1)/k$, and each nonidentity element of G is contained in exactly one of these subgroups or in exactly one Sylow p subgroup of G.

We estimate the number of elements that are contained in subgroups of type (3).

Lemma 2: Let $n_{sf}(q)$ denote the number of those nonidentity elements in G that are contained in proper subgroups of type (3). Then $n_{sf}(q) \leq q^2(2p(\sqrt{q}-1)/(p-1)-1)$. If f is a prime then $n_{sf}(q) \leq (2p-1)q^2$ holds, and if $p = q$ then we have of course $n_{sf}(q) = 0$.

Proof: The group $\mathrm{PGL}(2,p^m)$ is equal to $\mathrm{PSL}(2,p^m)$ for $p = 2$, and contains $\mathrm{PSL}(2,p^m)$ as a subgroup of index two if $p \neq 2$. So the largest element order in $\mathrm{PGL}(2,p^m)$ is at most $p^m + 1$. Let C be a cyclic subgroup of order $(q+\varepsilon)/k$ in G, for $\varepsilon \in \{\pm 1\}$. The intersection of C with any subgroup of G isomorphic with $\mathrm{PGL}(2,p^m)$ or $\mathrm{PSL}(2,p^m)$ is contained in the union of the unique subgroups of the orders $\gcd(|C|,p^m+1)$ and $\gcd(|C|,p^m-1)$ in C. So C contains at most $2p^m - 2$ nonidentity elements that can lie inside subgroups isomorphic with $\mathrm{PGL}(2,p^m)$ or $\mathrm{PSL}(2,p^m)$. Hence C contains at most $\sum_m (2p^m - 2)$ nonidentity elements in proper subgroups of type (3), where m runs over the proper divisors of f. This sum is bounded from above by $\sum_{m=1}^{f/2}(2p^m - 2) \leq 2p(\sqrt{q}-1)/(p-1)-2$.

The numbers of cyclic subgroups of the orders $(q+\varepsilon)/k$ in G are $q(q-\varepsilon)/2$, so G contains altogether q^2 such cyclic subgroups. They contain at most $q^2(2p(\sqrt{q}-1)/(p-1)-2)$ elements inside proper subgroups of the type (3).

All elements of order p in G are contained in subgroups of type (3), and there are exactly $q^2 - 1$ such elements. This yields the claimed bound for $n_{sf}(q)$. The better bound for the case that f is a prime follows from $\sum_m (2p^m - 2) = 2p - 2$ if m ranges over the proper divisors of f.   □

Using these bounds, we see that the vertex degree of any element in G that does not lie in subgroups of type (4) is larger than $|G|/2$. (In fact we could use the calculations below to derive a better asymptotic bound, but this is not an issue here.)

Lemma 3: Let $s \in G$ be an element of order larger than 5. Then $|\{g \in G; \langle g,s \rangle = G\}| > |G|/2$.

Proof: First suppose that the order of s divides $(q+1)/k$ or $(q-1)/k$. If $g \in G$ such that $U = \langle s,g \rangle$ is a proper subgroup of G then $U \leq N_G(\langle s \rangle)$ or U lies in a Sylow p normalizer of G or U lies in a subgroup of type (3). Since s is contained in at most two Sylow p normalizers (each Sylow p

normalizer contains q cyclic subgroups of order $(q-1)/k$, and G contains $q+1$ Sylow normalizers and $q(q+1)/2$ cyclic subgroups of order $(q-1)/k$, the number of $g \in G$ with the property that $\langle s,g \rangle \neq G$ is at most $N = 2(q+1)/k + 2q(q-1)/k + n_{sf}(q) = 2(q^2+1)/k + n_{sf}(q)$; for s of order equal to $(q+1)/k$ or $(q-1)/k$, we can set $N = 2(q^2+1)/k$.

Any element s of order p (larger than 5), lies only in a unique Sylow p normalizer and in subgroups of type (3), so the bound N holds also in this case.

For $f = 1$, N is smaller than $|G|/2 = q(q^2-1)/(2k)$ if $q \geq 5$. (The statement of the lemma is trivially true for $q \leq 5$.)

For primes f, N is smaller than $|G|/2$ if $q^2(q-8p) > q+4$ holds, which is true for $p^f > 8p$. Only the following values of p^f with prime f do not satisfy this condition: 2^2 and 3^2 (where no element of order larger than 5 exists), 2^3 (where only elements of order equal to $q \pm 1$ must be considered), 5^2 and 7^2 (where $n_{sf}(q) < (p-1)q(q+1)$ because in these cases the cyclic subgroups of order $(q+1)/k$ cannot contain nonidentity elements in subgroups of type (3)).

Finally, if f is not a prime then N is smaller than $|G|/2$ if $q^2(q - 8p(\sqrt{q}-1)/(p-1)) > q+4$ holds, which is true for $q \geq 256$. The only values of p^f with non-prime f that do not satisfy this condition are 2^4, 2^6, and 3^4. In all three cases, we have in fact $N < |G|/2$, where we have to use the better bound $n_{sf}(q) < 16q^2$ in the third case. $\qquad\square$

In order to show that the generating graph of G satisfies Pósa's criterion, it suffices to show that the vertex degrees of involutions is larger than the number of involutions, and that the vertex degrees of elements of orders 2, 3, 4, and 5 are larger than the number of elements whose order is at most 5.

Lemma 4: Let $n(q,m)$ denote the number of elements of order m in G, and let $\varphi(m)$ denote the number of prime residues modulo m.

- We have $n(q,2) = q^2 - 1$ if q is even and $n(q,2) \leq q(q+1)/2$ if q is odd.

- For $m \in \{3,4,5\}$, we have $n(q,m) \leq \varphi(m)q(q+1)/2$.

- We have $n(q,(q+1)/k) = \varphi((q+1)/k)q(q-1)/2$.

Lemma 5: If $q > 11$ then each involution in G has vertex degree larger than $n(q,2)$.

If $\varphi((q+1)/k) \geq 12$ then each element of order 3, 4, or 5 has vertex degree larger than $\sum_{m=2}^{5} n(q,m)$.

Proof: Let $s \in G$ of order at most 5. For each element $g \in G$ of order $(q+1)/k$, $U = \langle g,s \rangle$ is either G or contained in the dihedral group of order $2(q+1)/k$ that normalizes $\langle g \rangle$.

If s is an involution then the number of such dihedral groups that contain s is at most $(q+3)/2$, and at least $n(q,(q+1)/k) - \varphi((q+1)/k)(q+3)/2 = \varphi((q+1)/k)(q^2 - 2q - 3)/2$ elements of order $(q+1)/k$ contribute to the vertex degree of s. This number is larger than $q^2 - 1 \geq n(q,2)$ if $q > 11$ (and hence $\varphi((q+1)/k) \geq 3$) holds.

If s is an element of order 3, 4, or 5 then $U \neq G$ means that $s \in \langle g \rangle$, so at least $n(q,(q+1)/k) - 4$ elements of order $(q+1)/k$ contribute to the vertex degree of s. This number is larger than $5q(q+1) > \sum_{m=2}^{5} n(q,m)$ if $\varphi((q+1)/k) \geq 12$. $\qquad\square$

It remains to deal with the values q where $\varphi((q+1)/k) < 12$, that is, $(q+1)/k \leq 30$. We compute that the statement of Lemma 5 is true also for prime powers q with $11 < q \leq 59$.

```
                            ─── Example ───
  gap> TestL2q:= function( t )
  >      local name, orders, nccl, cl, prim, bds, n, ord;
  >
  >      name:= Identifier( t );
  >      orders:= OrdersClassRepresentatives( t );
```

```
>    nccl:= Length( orders );
>    cl:= SizesConjugacyClasses( t );
>    prim:= PrimitivePermutationCharacters( t );
>    bds:= List( LowerBoundsVertexDegrees( cl, prim ), Sum );
>    n:= List( [ 1 .. 5 ], i -> Sum( cl{ Filtered( [ 1 .. nccl ],
>                                        x -> orders[x] = i ) } ) );
>    if ForAny( Filtered( [ 1 .. nccl ], i -> orders[i] > 5 ),
>             i -> bds[i-1] <= Size( t ) / 2 ) then
>      Error( "problem with large orders for ", name );
>    elif ForAny( Filtered( [ 1 .. nccl ], i -> orders[i] = 2 ),
>               i -> bds[i-1] <= n[2] ) then
>      Error( "problem with order 2 for ", name, "\n" );
>    elif ForAny( Filtered( [ 1 .. nccl ],
>                       i -> orders[i] in [ 3 .. 5 ] ),
>               i -> bds[i-1] <= Sum( n{ [ 2 .. 5 ] } ) ) then
>      Error( "problem with order in [ 3 .. 5 ] for ", name );
>    fi;
> end;;
gap> for q in Filtered( [ 13 .. 59 ], IsPrimePowerInt ) do
>       TestL2q( CharacterTable(
>               Concatenation( "L2(", String( q ), ")" ) ) );
>    od;
```

For $2 \leq q \leq 11$, the statement of Lemma 5 is not true but Pósa's criterion is satisfied for the generating graphs of the groups $\mathrm{PSL}(2,q)$ with $2 \leq q \leq 11$.

```
────────── Example ──────────
gap> for q in Filtered( [ 2 .. 11 ], IsPrimePowerInt ) do
>       info:= HamiltonianCycleInfoFromGroup( PSL( 2, q ) );
>       if info <> "Posa for 0th closure" then
>         Print( q, ": ", info, "\n" );
>       fi;
>    od;
```

Chapter 5

GAP Computations with $O_8^+(5).S_3$ and $O_8^+(2).S_3$

Date: October 08th, 2006

This chapter shows how to construct a representation of the automorphic extension G of the simple group $S = O_8^+(5)$ by a symmetric group on three points, together with an embedding of the normalizer H of an $O_8^+(2)$ type subgroup of $O_8^+(5)$.

As an application, it is shown that the permutation representation of G on the cosets of H has a base of length two. This question arose in [BGS11].

5.1 Overview

Let S denote the simple group $O_8^+(5) \cong P\Omega^+(8,5)$, that is, the nonabelian simple group that occurs as a composition factor of the general orthogonal group $GO^+(8,5)$ of 8×8 matrices over the field with five elements.

The outer automorphism group of S is isomorphic to the symmetric group on four points. Let G be an automorphic extension of S by the symmetric group on three points. By [Kle87], the group S contains a maximal subgroup M of the type $O_8^+(2)$ such that the normalizer H, say, of M in G is an automorphic extension of M by a symmetric group on three points. (In fact, H is isomorphic to the full automorphism group of $O_8^+(2)$.)

Let $S.2$ and $S.3$ denote intermediate subgroups between S and G, in which S has the indices 2 and 3, respectively. Analogously, let $M.2 = H \cap S.2$ and $M.3 = H \cap S.3$.

In Section 5.2, we use the following approach to construct representations of $M.2$ and $S.2$. By [CCN+85, p. 85], the Weyl group W of type E_8 is a double cover of $M.2$, and the reduction of its rational 8-dimensional representation modulo 5 embeds into the general orthogonal group $GO^+(8,5)$, which has the structure $2.O_8^+(5).2^2$. Then the actions of $GO^+(8,5)$ and of an isomorphic image of W in $GO^+(8,5)$ on 1-spaces in the natural module of $GO^+(8,5)$ yield $M.2$ as a subgroup of (a supergroup of) $S.2$, where both groups are represented as permutation groups on $N = 19\,656$ points.

In Section 5.3, first we use GAP to compute the automorphism group of M. Then we take an outer automorphism α of M, of order three, and extend α to an automorphism of S. Concretely, we compute the images of generating sets of S and M under α and α^2. This yields permutation representations of $S.3$ and its subgroup $M.3$ on $3N = 58\,968$ points.

In Section 5.4, we put the above information together, in order to construct permutation representations of G and M, on $3N$ points.

As an application, it is shown in Section 5.5 that the permutation representation of G on the cosets of H has a base of length two; this question arose in [BGS11].

In two appendices, it is discussed how to derive a part of this result from the permutation character $(1_H^G)_H$ (see Section 5.6), and a file containing the data used in the earlier sections is described (see Section 5.7).

5.2   Constructing Representations of $M.2$ and $S.2$

5.2.1   A Matrix Representation of the Weyl Group of Type E_8

Following the recipe listed in [CCN+85, p. 85, Section Weyl], we can generate the Weyl group W of type E_8 as a group of rational 8×8 matrices generated by the reflections in the vectors

$$(\pm 1/2, \pm 1/2, 0, 0, 0, 0, 0, 0)$$

plus the vectors obtained from these by permuting the coordinates, plus those those vectors of the form

$$(\pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2)$$

that have an even number of negative signs. (Clearly it is sufficient to consider only one vector form a pair $\pm v$.)

```
———————————————————————— Example ————————————————————————
  gap> rootvectors:= [];;
  gap> for i in Combinations( [ 1 .. 8 ], 2 ) do
  >        v:= 0 * [ 1 .. 8 ];
  >        v{i}:= [ 1, 1 ];
  >        Add( rootvectors, v );
  >        v:= 0 * [ 1 .. 8 ];
  >        v{i}:= [ 1, -1 ];
  >        Add( rootvectors, v );
  >     od;
  gap> Append( rootvectors,
  >        1/2 * Filtered( Tuples( [ -1, 1 ], 8 ),
  >              x -> x[1] = 1 and Number( x, y -> y = 1 ) mod 2 = 0 ) );
  gap> we8:= Group( List( rootvectors, ReflectionMat ) );
  <matrix group with 120 generators>
```

5.2.2   Embedding the Weyl group of Type E_8 into $\mathrm{GO}^+(8,5)$

The elements in the group constructed above respect the symmetric bilinear form that is given by the identity matrix.

```
———————————————————————— Example ————————————————————————
  gap> I:= IdentityMat( 8 );;
  gap> ForAll( GeneratorsOfGroup( we8 ), x -> x * TransposedMat(x) = I );
  true
```

So the reduction of the matrices modulo 5 yields a group W^* of orthogonal matrices w. r. t. the identity matrix. The group $\mathrm{GO}^+(8,5)$ returned by the GAP function GO (**Reference: GO**) leaves a different bilinear form invariant.

```
——————————————————— Example ———————————————————
gap> largegroup:= GO(1,8,5);;
gap> Display( InvariantBilinearForm( largegroup ).matrix );
 . 1 . . . . . .
 1 . . . . . . .
 . . 2 . . . . .
 . . . 2 . . . .
 . . . . 2 . . .
 . . . . . 2 . .
 . . . . . . 2 .
 . . . . . . . 2
```

In order to conjugate W^* into this group, we need a 2×2 matrix T over the field with five elements with the property that TT^{tr} is half of the upper left 2×2 matrix in the above matrix.

```
——————————————————— Example ———————————————————
gap> T:= [ [ 1, 2 ], [ 4, 2 ] ] * One( GF(5) );;
gap> Display( 2 * T * TransposedMat( T ) );
 . 1
 1 .
gap> I:= IdentityMat( 8, GF(5) );;
gap> I{ [ 1, 2 ] }{ [ 1, 2 ] }:= T;;
gap> conj:= List( GeneratorsOfGroup( we8 ), x -> I * x * I^-1 );;
gap> IsSubset( largegroup, conj );
true
```

5.2.3 Compatible Generators of M, $M.2$, S, and $S.2$

For the next computations, we switch from the natural matrix representation of $\mathrm{GO}^+(8,5)$ to a permutation representation of $\mathrm{PGO}^+(8,5)$, of degree $N = 19656$, which is given by the action of $\mathrm{GO}^+(8,5)$ on the smallest orbit of 1-spaces in its natural module.

```
——————————————————— Example ———————————————————
gap> orbs:= OrbitsDomain( largegroup, NormedRowVectors( GF(5)^8 ),
>                         OnLines );;
gap> List( orbs, Length );
[ 39000, 39000, 19656 ]
gap> N:= Length( orbs[3] );
19656
gap> orbN:= SortedList( orbs[3] );;
gap> largepermgroup:= Action( largegroup, orbN, OnLines );;
```

In the same way, permutation representations of the subgroup $M.2 \cong \mathrm{SO}^+(8,2)$ and of its derived subgroup M are obtained. But first we compute a smaller generating set of the simple group M, using a permutation representation on 120 points.

```
——————————————————— Example ———————————————————
gap> orbwe8:= SortedList( Orbit( we8, rootvectors[1], OnLines ) );;
gap> Length( orbwe8 );
120
gap> we8_to_m2:= ActionHomomorphism( we8, orbwe8, OnLines );;
gap> m2_120:= Image( we8_to_m2 );;
gap> m_120:= DerivedSubgroup( m2_120 );;
```

```
gap> sml:= SmallGeneratingSet( m_120 );;  Length( sml );
2
gap> gens_m:= List( sml, x -> PreImagesRepresentative( we8_to_m2, x ) );;
```

Now we compute the actions of M and $M.2$ on the above orbit of length N. For generating $M.2$, we choose an element $b_N \in M.2 \setminus M$, which is obtained from the action of a matrix $b \in 2.M.2 \setminus 2.M$.

```
——————————————————————————— Example ———————————————————————————
gap> gens_m_N:= List( gens_m,
>        x -> Permutation( I * x * I^-1, orbN, OnLines ) );;
gap> m_N:= Group( gens_m_N );;
gap> b:= I * we8.1 * I^-1;;
gap> DeterminantMat( b );
Z(5)^2
gap> b_N:= Permutation( b, orbN, OnLines );;
gap> m2_N:= ClosureGroup( m_N, b_N );;
```

(Note that $M.2$ is not contained in $\mathrm{PSO}^+(8,5)$, since the determinant of b is -1 in the field with five elements.)

The group S is the derived subgroup of $\mathrm{PSO}^+(8,5)$, and $S.2$ is generated by S together with b_N.

```
——————————————————————————— Example ———————————————————————————
gap> s_N:= DerivedSubgroup( largepermgroup );;
gap> s2_N:= ClosureGroup( s_N, b_N );;
```

5.3 Constructing Representations of $M.3$ and $S.3$

5.3.1 The Action of $M.3$ on M

Let α be an automorphism of M, of order three. Then a representation of the semidirect product $M.3$ of M by $\langle \alpha \rangle$ can be constructed as follows.

If M is given by a matrix representation then we map $g \in M$ to the block diagonal matrix

$$
\begin{bmatrix}
g & & \\
& g^\alpha & \\
& & g^{(\alpha^2)}
\end{bmatrix},
$$

and we represent α by the block permutation matrix

$$
\begin{bmatrix}
 & & I \\
I & & \\
& I &
\end{bmatrix},
$$

where I is the identity element in M.

We need the action of α on M. More precisely, we need images of the chosen generators of M under α and α^2.

The group M is small enough for asking GAP to compute its automorphism group, which is isomorphic with $O_8^+(2).S_3$; for that, we use the degree 120 permutation representation constructed in Section 5.2.3.

```
                          Example
  gap> aut_m:= AutomorphismGroup( m_120 );;
```

We pick an outer automorphism α of order three.

```
                          Example
  gap> nice_aut_m:= NiceMonomorphism( aut_m );;
  gap> der:= DerivedSubgroup( Image( nice_aut_m ) );;
  gap> der2:= DerivedSubgroup( der );;
  gap> repeat x:= Random( der );
  >       ord:= Order( x );
  >     until ord mod 3 = 0 and ord mod 9 <> 0 and not x in der2;
  gap> x:= x^( ord / 3 );;
  gap> alpha_120:= PreImagesRepresentative( nice_aut_m, x );;
```

Next we compute the images of the generators `sml` under α and α^2, and the corresponding elements in the action of M on N points.

```
                          Example
  gap> sml_alpha:= List( sml, x -> Image( alpha_120, x ) );;
  gap> sml_alpha_2:= List( sml_alpha, x -> Image( alpha_120, x ) );;
  gap> gens_m_alpha:= List( sml_alpha,
  >                     x -> PreImagesRepresentative( we8_to_m2, x ) );;
  gap> gens_m_alpha_2:= List( sml_alpha_2,
  >                       x -> PreImagesRepresentative( we8_to_m2, x ) );;
  gap> gens_m_N_alpha:= List( gens_m_alpha,
  >      x -> Permutation( I * x * I^-1, orbN, OnLines ) );;
  gap> gens_m_N_alpha_2:= List( gens_m_alpha_2,
  >      x -> Permutation( I * x * I^-1, orbN, OnLines ) );;
```

Finally, we use the construction descibed in the beginning of this section, and obtain a permutation representation of $M.3$ on $3N = 58\,968$ points.

```
                          Example
  gap> alpha_3N:= PermList( Concatenation( [ [ 1 .. N ] + 2*N,
  >                                          [ 1 .. N ],
  >                                          [ 1 .. N ] + N ] ) );;
  gap> gens_m_3N:= List( [ 1 .. Length( gens_m_N ) ],
  >      i -> gens_m_N[i] *
  >          ( gens_m_N_alpha[i]^alpha_3N ) *
  >          ( gens_m_N_alpha_2[i]^(alpha_3N^2) ) );;
  gap> m_3N:= Group( gens_m_3N );;
  gap> m3_3N:= ClosureGroup( m_3N, alpha_3N );;
```

5.3.2 The Action of $S.3$ on S

Our approach is to extend the automorphism α of M to S; we can do this because in the full automorphism group of S, *any* $O_8^+(2)$ type subgroup extends to a group of the type $O_8^+(2).3$, and this extension lies in a subgroup of the type $O_8^+(5).3$ (see [Kle87]).

The group M is maximal in S, so S is generated by M together with any element $s \in S \setminus M$. Having fixed such an element s, what we have to is to find the images of s under the automorphisms that extend α and α^2.

For that, we first choose $x \in M$ such that $C_S(x)$ is a small group that is not contained in M. Then we choose $s \in C_S(x) \setminus M$, and using that s^α must lie in $C_S(C_M(s)^\alpha)$, we then check which elements of this small subgroup can be the desired image.

Each element x of order nine in M has a root s of order 63 in S, and $C_S(x)$ has order 189. For suitable such x, exactly one element $y \in C_S(C_M(s)^\alpha)$ has order 63 and satisfies the necessary conditions that the orders of the products of s and the generators of M are equal to the orders of the product of y and the images of these generators under α. In other words, we have $s^\alpha = y$.

```
─────────────────────────── Example ───────────────────────────
  gap> alpha:= GroupHomomorphismByImagesNC( m_N, m_N,
  >                gens_m_N, gens_m_N_alpha );;
  gap> CheapTestForHomomorphism:= function( gens, genimages, x, cand )
  >        return Order( x ) = Order( cand ) and
  >               ForAll( [ 1 .. Length( gens ) ],
  >           i -> Order( gens[i] * x ) = Order( genimages[i] * cand ) );
  > end;;
  gap> repeat
  >       repeat
  >         x:= Random( m_N );
  >       until Order( x ) = 9;
  >       c_s:= Centralizer( s_N, x );
  >       repeat
  >         s:= Random( c_s );
  >       until Order( s ) = 63;
  >       c_m_alpha:= Images( alpha, Centralizer( m_N, s ) );
  >       good:= Filtered( Elements( Centralizer( s_N, c_m_alpha ) ),
  >              x -> CheapTestForHomomorphism( gens_m_N,
  >                    gens_m_N_alpha, s, x ) );
  >    until Length( good ) = 1;
  gap> s_alpha:= good[1];;
  gap> c_m_alpha_2:= Images( alpha, c_m_alpha );;
  gap> good:= Filtered( Elements( Centralizer( s_N, c_m_alpha_2 ) ),
  >       x -> CheapTestForHomomorphism( gens_m_N_alpha, gens_m_N_alpha_2,
  >                                       s_alpha, x ) );;
  gap> s_alpha_2:= good[1];;
```

Using the notation of the previous section, this means that the permutation representation of $M.3$ on $3N$ points can be extended to $S.3$ by choosing the permutation corresponding to the block diagonal matrix

$$\begin{bmatrix} s & & \\ & s^\alpha & \\ & & s^{(\alpha^2)} \end{bmatrix},$$

as an additional generator.

```
─────────────────────────── Example ───────────────────────────
  gap> outer:= s * ( s_alpha^alpha_3N ) * ( s_alpha_2^(alpha_3N^2) );;
  gap> s3_3N:= ClosureGroup( m3_3N, outer );;
```

(And of course we have $S = \langle M, s \rangle$, which yields generators for S that are compatible with those of M.)

```
─────────────────────────── Example ───────────────────────────
  gap> s_3N:= ClosureGroup( m_3N, outer );;
```

5.4 Constructing Compatible Generators of H and G

After having constructed compatible representations of $M.2$ and $G.2$ on N points (see Section 5.2.3) and of $M.3$ and $S.3$ on $3N$ points (see Section 5.3.2), the last construction step is to find a permutation on $3N$ points with the following properties:

- The induced automorphism β of M extends to $M.3$ such that the automorphism α of M is inverted, modulo inner automorphisms of M.

- The action on the first N points coincides with that of the element $b_N \in M.2 \setminus M$ that was constructed in Section 5.2.3.

Using the notation of the previous sections, we represent β by a block matrix

$$\begin{bmatrix} b & & \\ & & bd \\ & bg & \end{bmatrix},$$

where b describes the action of β on M (on N points), g describes the inner automorphism γ of M that is defined by the condition $\beta\alpha = \alpha^2\beta\gamma$, and d describes $\gamma\gamma^\alpha$.

So we compute an element in M that induces the conjugation automorphism γ, and its image under α. We do this in the representation of M on 120 points, and carry over the result to the representation on N points, via the rational matrix representation; this approach had been used already in Section 5.2.3.

```
─────────── Example ───────────
 gap> b_120:= Permutation( we8.1, orbwe8, OnLines );;
 gap> g_120:= RepresentativeAction( m_120,
 >            List( sml_alpha_2, x -> x^b_120 ),
 >            List( sml, x -> (x^b_120)^alpha_120 ), OnTuples );;
 gap> g_120_alpha:= g_120^alpha_120;;
 gap> g_N:= Permutation( I * PreImagesRepresentative( we8_to_m2, g_120 )
 >                     * I^-1, orbN, OnLines );;
 gap> g_N_alpha:= Permutation( I * PreImagesRepresentative( we8_to_m2,
 >            g_120_alpha ) * I^-1, orbN, OnLines );;
 gap> inv:= PermList( Concatenation(
 >            ListPerm( b_N ),
 >            ListPerm( b_N * g_N ) + 2*N,
 >            ListPerm( b_N * g_N * g_N_alpha ) + N ) );;
```

So we have constructed compatible generators for H and G.

```
─────────── Example ───────────
 gap> h:= ClosureGroup( m3_3N, inv );;
 gap> g:= ClosureGroup( s3_3N, inv );;
```

5.5 Application: Regular Orbits of H on G/H

We want to show that H has regular orbits on the right cosets G/H. The stabilizer in H of the coset Hg is $H \cap H^g$, so we compute that there are elements $s \in S$ with the property $|H \cap H^s| = 1$.

(Of course this implies that also in the permutation representations of the subgroups S, $S.2$, and $S.3$ of G on the cosets of the intersection with H, the point stabilizers have regular orbits.)

```
                          ————— Example —————
  gap> repeat
  >       conj:= Random( s_3N );
  >       inter:= Intersection( h, h^conj );
  >     until Size( inter ) = 1;
```

Eventually GAP will return from this loop, so there are elements *s* with the required property.

(Computing one such intersection takes about six minutes on a 2.5 GHz Pentium 4, so one may have to be a bit patient.)

5.6 Appendix: The Permutation Character $(1_H^G)_H$

As an alternative to the computation of $|H \cap H^s|$ for suitable $s \in S$, we can try to derive information from the permutation character $(1_H^G)_H$. Unfortunately, there seems to be no easy way to prove the existence of regular *H*-orbits on G/H (cf. Section 5.5) only by means of this character.

However, it is not difficult to show that regular orbits of *M*, *M*.2, and *M*.3 exist. For that, we compute $(1_H^G)_H$, by computing class representatives of *H*, their centralizer orders in *G*, and the class fusion of *H*-classes in *G*.

We want to compute the class representatives in a small permutation representation of *H*; this could be done using the degree 360 representation that was implicitly constructed above, but it is technically easier to use a degree 405 representation that is obtained from the degree 58 968 representation by the action of *H* on blocks in an orbit of length 22 680. (One could get this also using the GAP function `SmallerDegreePermutationRepresentation` (**Reference: SmallerDegreePermutationRepresentation**).)

```
                          ————— Example —————
  gap> orbs:= Orbits( h, MovedPoints( h ) );;
  gap> List( orbs, Length );
  [ 22680, 36288 ]
  gap> orb:= orbs[1];;
  gap> bl:= Blocks( h, orb );;  Length( bl[1] );
  2
  gap> actbl:= Action( h, bl, OnSets );;
  gap> bll:= Blocks( actbl, MovedPoints( actbl ) );;  Length( bll );
  405
  gap> oneblock:= Union( bl{ bll[1] } );;
  gap> orb:= SortedList( Orbit( h, oneblock, OnSets ) );;
  gap> acthom:= ActionHomomorphism( h, orb, OnSets );;
  gap> ccl:= ConjugacyClasses( Image( acthom ) );;
  gap> reps:= List( ccl, x -> PreImagesRepresentative( acthom,
  >                           Representative( x ) ) );;
```

Then we carry back class representatives to the degree 58 968 representation, and compute the class fusion and the centralizer orders in *G*.

```
                          ————— Example —————
  gap> reps:= List( ccl, x -> PreImagesRepresentative( acthom,
  >                           Representative( x ) ) );;
  gap> fusion:= [];;
  gap> centralizers:= [];;
  gap> fusreps:= [];;
```

```
gap> for i in [ 1 .. Length( reps ) ] do
>       found:= false;
>       cen:= Size( Centralizer( g, reps[i] ) );
>       for j in [ 1 .. Length( fusreps ) ] do
>         if cen = centralizers[j] and
>            IsConjugate( g, fusreps[j], reps[i] ) then
>           fusion[i]:= j;
>           found:= true;
>           break;
>         fi;
>       od;
>       if not found then
>         Add( fusreps, reps[i] );
>         Add( fusion, Length( fusreps ) );
>         Add( centralizers, cen );
>       fi;
>     od;
```

Next we compute the permutation character values, using the formula

$$(1_H)^G(g) = (|C_G(g)| \sum_h |h^H|)/|H|,$$

where the summation runs over class representatives $h \in H$ that are G-conjugate to g.

```
———————————————— Example ————————————————
gap> pi:= 0 * [ 1 .. Length( fusreps ) ];;
gap> for i in [ 1 .. Length( ccl ) ] do
>       pi[ fusion[i] ]:= pi[ fusion[i] ] + centralizers[ fusion[i] ] *
>                                            Size( ccl[i] );
>     od;
gap> pi:= pi{ fusion } / Size( h );;
```

In order to write the permutation character w.r.t. the ordering of classes in the GAP character table, we use the GAP function `CompatibleConjugacyClasses` (**Reference: CompatibleConjugacyClasses**).

```
———————————————— Example ————————————————
gap> tblh:= CharacterTable( "O8+(2).S3" );;
gap> map:= CompatibleConjugacyClasses( Image( acthom ), ccl, tblh );;
gap> pi:= pi{ map };
[ 51162109375, 69375, 1259375, 69375, 568750, 1750, 4000, 375, 135,
  975, 135, 625, 150, 650, 30, 72, 80, 72, 27, 27, 3, 7, 25, 30, 6,
  12, 25, 484375, 1750, 375, 375, 30, 40, 15, 15, 15, 6, 6, 3, 3, 3,
  157421875, 121875, 4875, 475, 75, 3875, 475, 13000, 1750, 300, 400,
  30, 60, 15, 15, 15, 125, 10, 30, 4, 8, 6, 9, 7, 5, 6, 5 ]
```

Now we consider the restrictions of this permutation character to M, $M.2$, and $M.3$. Note that $(1_H^G)_M = (1_M^S)_M$, $(1_H^G)_{M.2} = (1_{M.2}^{S.2})_{M.2}$, and $(1_H^G)_{M.3} = (1_{M.3}^{S.3})_{M.3}$.

```
———————————————— Example ————————————————
gap> tblm2:= CharacterTable( "O8+(2).2" );;
gap> tblm3:= CharacterTable( "O8+(2).3" );;
```

```
gap> tblm:= CharacterTable( "O8+(2)" );;
gap> pi_m2:= pi{ GetFusionMap( tblm2, tblh ) };;
gap> pi_m3:= pi{ GetFusionMap( tblm3, tblh ) };;
gap> pi_m:= pi_m3{ GetFusionMap( tblm, tblm3 ) };;
```

The permutation character $(1_M^S)_M$ decomposes into 483 transitive permutation characters, and regular M-orbits on S/M correspond to regular constituents in this decomposition. If there is no regular transitive constituent in $(1_M^S)_M$ then the largest degree of a transitive constituent is $|M|/2$; but then the degree of 1_M^S is less than $483|M|/2$, which is smaller than $[S:M]$.

```
————————— Example —————————
gap> n:= ScalarProduct( tblm, pi_m, TrivialCharacter( tblm ) );
483
gap> n * Size( tblm ) / 2;
42065049600
gap> pi[1];
51162109375
```

For the case of $M.2 < S.2$, this argument turns out to be not sufficient. So we first compute a lower bound on the number of regular M-orbits on S/M. For involutions $g \in M$, the number of transitive constituents $1_{\langle g \rangle}^M$ in $(1_M^S)_M$ is at most the integral part of $1_M^S(g)/1_{\langle g \rangle}^M(g) = 2 \cdot 1_M^S(g)/|C_M(g)|$; from this we compute that there are at most 208 such constituents.

```
————————— Example —————————
gap> inv:= Filtered( [ 1 .. NrConjugacyClasses( tblm ) ],
>             i -> OrdersClassRepresentatives( tblm )[i] = 2 );
[ 2, 3, 4, 5, 6 ]
gap> n2:= List( inv,
>           i -> Int( 2 * pi_m[i] / SizesCentralizers( tblm )[i] ) );
[ 1, 54, 54, 54, 45 ]
gap> Sum( n2 );
208
```

As a consequence, M has at least 148 regular orbits on S/M.

```
————————— Example —————————
gap> First( [ 1 .. 483 ],
>     i -> i * Size( tblm ) + 208 * Size( tblm ) / 2
>           + ( 483 - i - 208 - 1 ) * Size( tblm ) / 3 + 1 >= pi[1] );
148
```

Now we consider the action of $M.2$ on $S.2/M.2$. If $M.2$ has no regular orbit then the 148 regular orbits of M must arise from the restriction of transitive constituents $1_U^{M.2}$ to M with $|U| = 2$ and such that U is not contained in M. (This follows from the fact that the restriction of a transitive constituent of $(1_{M.2}^{S.2})_{M.2}$ to M is either itself a transitive constituent of $(1_M^S)_M$ or the sum of two such constituents; the latter case occurs if and only if the point stabilizer is contained in M.) However, the number of these constituents is at most 134.

```
————————— Example —————————
gap> inv:= Filtered( [ 1 .. NrConjugacyClasses( tblm2 ) ],
>             i -> OrdersClassRepresentatives( tblm2 )[i] = 2 and
>                 not i in ClassPositionsOfDerivedSubgroup( tblm2 ) );
```

```
[ 41, 42 ]
gap> n2:= List( inv,
>            i -> Int( 2 * pi_m2[i] / SizesCentralizers( tblm2 )[i] ) );
[ 108, 26 ]
gap> Sum( n2 );
134
```

Finally, we consider the action of $M.3$ on $S.3/M.3$. We compute that $(1_{M.3}^{S.3})_{M.3}$ has 205 transitive constituents, and at most 69 of them can be induced from subgroups of order two. This is already sufficient to show that there must be regular constituents.

```
─────────── Example ───────────
gap> n:= ScalarProduct( tblm3, pi_m3, TrivialCharacter( tblm3 ) );
205
gap> inv:= Filtered( [ 1 .. NrConjugacyClasses( tblm3 ) ],
>            i -> OrdersClassRepresentatives( tblm3 )[i] = 2 );
[ 2, 3, 4 ]
gap> n2:= List( inv,
>            i -> Int( 2 * pi_m3[i] / SizesCentralizers( tblm3 )[i] ) );
[ 0, 54, 15 ]
gap> Sum( n2 );
69
gap> 69 * Size( tblm3 ) / 2 + ( n - 69 - 1 ) * Size( tblm3 ) / 3 + 1;
41542502401
gap> pi[1];
51162109375
```

5.7 Appendix: The Data File

The file o8p2s3_o8p5s3.g that can be found at

http://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib/data/o8p2s3_o8p5s3.g

contains the relevant data used in the above computations. This covers the representations for the groups and the permutation character of $O_8^+(2).S_3$ computed in Section 5.6.

Reading the file into GAP will define a global variable o8p2s3_o8p5s3_data, a record with the following components.

pi   the list of values of the permutation character of $G = O_8^+(5).S_3$ on the cosets of its subgroup $H = O_8^+(2).S_3$, restricted to H, corresponding to the ordering of classes in the character table of H in the GAP Character Table Library (this table has the Identifier (**Reference: Identifier for tables of marks**) value "O8+(2).3.2"),

dim8Q
      a record with generators for $2.M$ and $2.M.2$, matrices of dimension eight over the Rationals,

deg120
      a record with generators for M and $M.2$, permutations of degree 120,

deg360
      a record with generators for M, $M.2$, $M.3$, and H, permutations of degree 360,

`dim8f5`

    a record with generators for $2.M$, $2.M.2$, $2.S$, and $2.S.2$, matrices of dimension eight over the field with five elements,

`deg19656`

    a record with generators for M, $M.2$, S, and $S.2$, permutations of degree $19\,656$,

`deg58968`

    a record with generators for M, $M.2$, $M.3$, H, S, $S.2$, $S.3$, and G, permutations of degree $58\,968$,

`seed405`

    a block whose H-orbit in the representation on $58\,968$ points, w.r.t. the action `OnSets` (**Reference: OnSets**), yields a representation of H on 405 points.

For each of the permutation representations, we have (where applicable)

$$
\begin{aligned}
M &\cong \langle a_1, a_2 \rangle, \\
M.2 &\cong \langle a_1, a_2, b \rangle, \\
M.3 &\cong \langle a_1, a_2, t \rangle, \\
H &\cong \langle a_1, a_2, t, b \rangle, \\
S &\cong \langle a_1, a_2, c \rangle, \\
S.2 &\cong \langle a_1, a_2, c, b \rangle, \\
S.3 &\cong \langle a_1, a_2, c, t \rangle, \\
G &\cong \langle a_1, a_2, c, t, b \rangle,
\end{aligned}
$$

where a_1, a_2, b, t, c are the values of the record components `a1`, `a2`, `b`, `t`, and `c`.
Analogously, for the matrix representations, we have (where applicable)

$$
\begin{aligned}
2.M &\cong \langle a_1, a_2 \rangle, \\
2.M.2 &\cong \langle a_1, a_2, b \rangle, \\
2.S &\cong \langle a_1, a_2, c \rangle, \\
2.S.2 &\cong \langle a_1, a_2, c, b \rangle,
\end{aligned}
$$

Additional components are used for deriving the representations from initial data, as in the constructions in the previous sections.

For example, most of the permutations needed arise as the induced actions of matrices on orbits of vectors; these orbits are computed when the file is read, and are then stored in the components `orb120` and `orb19656`.

The file `o8p2s3_o8p5s3.g` does not contain the generators explicitly, but it is self-contained in the sense that only a few GAP functions are actually needed to produce the data; for example, it should not be difficult to translate the contents of the file into the language of other computer algebra systems.

Advantages of this way to store the data are that the relations between the representations become explicit, and also that only very little space is needed to describe the representations –the size of the file is less than 10 kB, whereas storing (explicitly) one of the permutations on $58\,968$ points requires already about 350 kB.

Chapter 6

Solvable Subgroups of Maximal Order in Sporadic Simple Groups

Date: May 14th, 2012

We determine the orders of solvable subgroups of maximal orders in sporadic simple groups and their automorphism groups, using the information in the Atlas of Finite Groups [CCN+85] and the GAP system [GAP24], in particular its Character Table Library [Bre25] and its library of Tables of Marks [MNP19].

We also determine the conjugacy classes of these solvable subgroups in the big group, and the maximal overgroups.

A first version of this document, which was based on GAP 4.4.10, had been accessible in the web since August 2006. The differences to the current version are as follows.

- The format of the GAP output was adjusted to the changed behaviour of GAP 4.5.

- The (too wide) table of results was split into two tables, the first one lists the orders and indices of the subgroups, the second one lists the structure of subgroups and the maximal overgroups.

- The distribution of the solvable subgroups of maximal orders in the Baby Monster group and the Monster group to conjugacy classes is now proved.

- The sporadic simple Monster group has exactly one class of maximal subgroups of the type $\mathrm{PSL}(2,41)$ (see [NW13]), and has no maximal subgroups which have the socle $\mathrm{PSL}(2,27)$ (see [Wil10]). This does not affect the arguments in Section 6.4.14, but some statements in this section had to be corrected.

6.1 The Result

The tables I and II list information about solvable subgroups of maximal order in sporadic simple groups and their automorphism groups. The first column in each table gives the names of the almost simple groups G, in alphabetical order. The remaining columns of Table I contain the order and the index of a solvable subgroup S of maximal order in G, the value $\log_{|G|}(|S|)$, and the page number in the Atlas [CCN+85] where the information about maximal subgroups of G is listed. The second and third columns of Table II show a structure description of S and the structures of the maximal subgroups that contain S; the value "S" in the third column means that S is itself maximal in G. The fourth and fifth

columns list the pages in the Atlas with the information about the maximal subgroups of G and the section in this note with the proof of the table row, respectively. In the fourth column, page numbers in brackets refer to the Atlas pages with information about the maximal subgroups of nonsolvable quotients of the maximal subgroups of G listed in the third column.

Note that in the case of nonmaximal subgroups S, we do not claim to describe the *module* structure of S in the third column of the table; we have kept the Atlas description of the normal subgroups of the maximal overgroups of S. For example, the subgroup S listed for Co_2 is contained in maximal subgroups of the types $2^{1+8}_+ : S_6(2)$ and $2^{4+10}(S_4 \times S_3)$, so S has normal subgroups of the orders 2, 2^4, 2^9, 2^{14}, and 2^{16}; more Atlas conformal notations would be $2^{[14]}(S_4 \times S_3)$ or $2^{[16]}(S_3 \times S_3)$.

As a corollary (see Section 6.5), we read off the following.

Corollary:

Exactly the following almost simple groups G with sporadic simple socle contain a solvable subgroup S with the property $|S|^2 \geq |G|$.

$$Fi_{23}, J_2, J_2.2, M_{11}, M_{12}, M_{22}.2.$$

The existence of the subgroups S of G with the structure and the order stated in Table I and II follows from the Atlas: It is obvious in the cases where S is maximal in G, and in the other cases, the Atlas information about a nonsolvable factor group of a maximal subgroup of G suffices.

In order to show that the table rows for the group G are correct, we have to show the following.

- G does not contain solvable subgroups of order larger than $|S|$.

- G contain exactly the conjugacy classes of solvable subgroups of order $|S|$ that are listed in the second column of Table II.

- S is contained exactly in the maximal subgroups listed in the third column of Table II.

Remark:

- Each of the groups M_{12} and He contains two classes of isomorphic solvable subgroups of maximal order.

- Each of the groups Ru, Th, and M contains two classes of nonisomorphic solvable subgroups of maximal order.

- The solvable subgroups of maximal order in $McL.2$ have the structure $3^{1+4}_+ : 4S_4$, the subgroups are maximal in the maximal subgroups of the structures $3^{1+4}_+ : 4S_5$ and $U_4(3).2_3$ in $McL.2$. Note that the Atlas claims another structure for these maximal subgroups of $U_4(3).2_3$, see [CCN$^+$85, p. 52].

- The solvable subgroups of maximal order in Co_3 are the normalizers of Sylow 3–subgroups of Co_3.

G	$|S|$	$|G/S|$	$\log_{|G|}(|S|)$	p.
M_{11}	144	55	0.5536	18
M_{12}	432	220	0.5294	33
$M_{12}.2$	432	440	0.4992	33
J_1	168	1 045	0.4243	36
M_{22}	576	770	0.4888	39
$M_{22}.2$	1 152	770	0.5147	39
J_2	1 152	525	0.5295	42
$J_2.2$	2 304	525	0.5527	42
M_{23}	1 152	8 855	0.4368	71
HS	2 000	22 176	0.4316	80
$HS.2$	4 000	22 176	0.4532	80
J_3	1 944	25 840	0.4270	82
$J_3.2$	3 888	25 840	0.4486	82
M_{24}	13 824	17 710	0.4935	96
McL	11 664	77 000	0.4542	100
$McL.2$	23 328	77 000	0.4719	100
He	13 824	291 550	0.4310	104
$He.2$	18 432	437 325	0.4305	104
Ru	49 152	2 968 875	0.4202	126
Suz	139 968	3 203 200	0.4416	131
$Suz.2$	279 936	3 203 200	0.4557	131
$O'N$	25 920	17 778 376	0.3784	132
$O'N.2$	51 840	17 778 376	0.3940	132
Co_3	69 984	7 084 000	0.4142	134
Co_2	2 359 296	17 931 375	0.4676	154
Fi_{22}	5 038 848	12 812 800	0.4853	163
$Fi_{22}.2$	10 077 696	12 812 800	0.4963	163
HN	2 000 000	136 515 456	0.4364	166
$HN.2$	4 000 000	136 515 456	0.4479	166
Ly	900 000	57 516 865 560	0.3562	174
Th	944 784	96 049 408 000	0.3523	177
Fi_{23}	3 265 173 504	1 252 451 200	0.5111	177
Co_1	84 934 656	48 952 653 750	0.4258	183
J_4	28 311 552	3 065 023 459 190	0.3737	190
Fi'_{24}	29 386 561 536	42 713 595 724 800	0.4343	207
$Fi'_{24}.2$	58 773 123 072	42 713 595 724 800	0.4413	207
B	29 686 813 949 952	139 953 768 303 693 093 750	0.4007	217
M	2 849 934 139 195 392	283 521 437 805 098 363 752		
		344 287 234 566 406 250	0.2866	234

Table: Table I: Solvable subgroups of maximal order – orders and indices

G	S	Max. overgroups	[CCN$^+$85]		see
M_{11}	$3^2 : Q_8.2$	S	18		6.3
M_{12}	$3^2 : 2S_4$	S	33		6.3
	$3^2 : 2S_4$	S	33		6.3
$M_{12}.2$	$3^2 : 2S_4$	M_{12}	33		6.3
J_1	$2^3 : 7 : 3$	S	36		6.3
M_{22}	$2^4 : 3^2 : 4$	$2^4 : A_6$	39	(4)	6.3
$M_{22}.2$	$2^4 : 3^2 : D_8$	$2^4 : S_6$	39	(4)	6.3
J_2	$2^{2+4} : (3 \times S_3)$	S	42		6.3
$J_2.2$	$2^{2+4} : (S_3 \times S_3)$	S	42		6.3
M_{23}	$2^4 : (3 \times A_4) : 2$	$2^4 : (3 \times A_5) : 2,$	71	(2)	6.3
		$2^4 : A_7$		(10)	
HS	$5^{1+2}_+ : 8 : 2$	$U_3(5).2$	80	(34)	6.3
		$U_3(5).2$			6.3
$HS.2$	$5^{1+2}_+ : [2^5]$	S	80	(34)	6.3
J_3	$3^2.3^{1+2}_+ : 8$	S	82		6.3
$J_3.2$	$3^2.3^{1+2}_+ : QD_{16}$	S	82		6.3
M_{24}	$2^6 : 3^{1+2}_+ : D_8$	$2^6 : 3.S_6$	96	(4)	6.3
McL	$3^{1+4}_+ : 2S_4$	$3^{1+4}_+ : 2S_5,$	100	(2)	6.3
		$U_4(3)$		(52)	6.3
$McL.2$	$3^{1+4}_+ : 4S_4$	$3^{1+4}_+ : 4S_5,$	100	(2)	6.3
		$U_4(3).2_3$		(52)	6.3
He	$2^6 : 3^{1+2}_+ : D_8$	$2^6 : 3.S_6$	104	(4)	6.3
	$2^6 : 3^{1+2}_+ : D_8$	$2^6 : 3.S_6$		(4)	6.3
$He.2$	$2^{4+4}.(S_3 \times S_3).2$	S	104		6.3
Ru	$2.2^{4+6} : S_4$	$2^{3+8} : L_3(2),$	126	(3)	6.4.1
		$2.2^{4+6} : S_5$		(2)	
	$2^{3+8} : S_4$	$2^{3+8} : L_3(2),$		(3)	6.4.1
Suz	$3^{2+4} : 2(A_4 \times 2^2).2$	S	131		6.4.2
$Suz.2$	$3^{2+4} : 2(S_4 \times D_8)$	S	131		6.4.2
$O'N$	$3^4 : 2^{1+4}_- D_{10}$	S	132		6.4.3
$O'N.2$	$3^4 : 2^{1+4}_-.(5 : 4)$	S	132		6.4.3
Co_3	$3^{1+4}_+ : 4.3^2 : D_8$	$3^{1+4}_+ : 4S_6$	134	(4)	6.3
		$3^5 : (2 \times M_{11})$		(18)	
Co_2	$2^{4+10}(S_4 \times S_3)$	$2^{1+8}_+ : S_6(2),$	154	(46)	6.4.4
		$2^{4+10}(S_5 \times S_3)$		(2)	
Fi_{22}	$3^{1+6}_+ : 2^{3+4} : 3^2 : 2$	S	163		6.4.5
$Fi_{22}.2$	$3^{1+6}_+ : 2^{3+4} : (S_3 \times S_3)$	S	163		6.4.5
HN	$5^{1+4}_+ : 2^{1+4}_-.5.4$	S	166		6.4.6
$HN.2$	$5^{1+4}_+ : (4Y2^{1+4}_-.5.4)$	S	166		6.4.6
Ly	$5^{1+4}_+ : 4.3^2 : D_8$	$5^{1+4}_+ : 4S_6$	174	(4)	6.4.7
Th	$[3^9].2S_4$	S	177		6.4.8
	$3^2.[3^7].2S_4$	S			
Fi_{23}	$3^{1+8}_+.2^{1+6}_-.3^{1+2}_+.2S_4$	S	177		6.4.9
Co_1	$2^{4+12}.(S_3 \times 3^{1+2}_+ : D_8)$	$2^{4+12}.(S_3 \times 3S_6)$	183		6.4.10

Table: Table II: Solvable subgroups of maximal order – structures and overgroups

G	S	Max. overgroups	[CCN$^+$85]		see
J_4	$2^{11} : 2^6 : 3^{1+2}_+ : D_8$	$2^{11} : M_{24}$,	190	(96)	6.4.11
		$2^{1+12}_+.3M_{22} : 2$		(39)	
Fi'_{24}	$3^{1+10}_+ : 2^{1+6}_- : 3^{1+2}_+ : 2S_4$	$3^{1+10}_+ : U_5(2) : 2$	207	(73)	6.4.12
$Fi'_{24}.2$	$3^{1+10}_+ : (2 \times 2^{1+6}_- : 3^{1+2}_+ : 2S_4)$	$3^{1+10}_+ : (2 \times U_5(2) : 2)$	207	(73)	6.4.12
B	$2^{2+10+20}(2^4 : 3^2 : D_8 \times S_3)$	$2^{2+10+20}(M_{22} : 2 \times S_3)$,	217	(39)	6.4.13
		$2^{9+16}S_8(2)$		(123)	
M	$2^{1+2+6+12+18}.(S_4 \times 3^{1+2}_+ : D_8)$	$2^{[39]}.(L_3(2) \times 3S_6)$,	234	(3, 4)	6.4.14
		$2^{1+24}_+.Co_1$		(183)	
	$2^{2+1+6+12+18}.(S_4 \times 3^{1+2}_+ : D_8)$	$2^{[39]}.(L_3(2) \times 3S_6)$,		(3, 4)	6.4.14
		$2^{2+11+22}.(M_{24} \times S_3)$		(96)	

Table: Table II: Solvable subgroups of maximal order – structures and overgroups (continued)

6.2 The Approach

We combine the information in the Atlas [CCN$^+$85] with explicit computations using the GAP system [GAP24], in particular its Character Table Library [Bre25] and its library of Tables of Marks [MNP19]. First we load these two packages.

```
─────────── Example ───────────
gap> LoadPackage( "CTblLib", "1.2", false );
true
gap> LoadPackage( "TomLib", false );
true
```

The orders of solvable subgroups of maximal order will be collected in a global record `MaxSolv`.

```
─────────── Example ───────────
gap> MaxSolv:= rec();;
```

6.2.1 Use the Table of Marks

If the GAP library of Tables of Marks [MNP19] contains the table of marks of a group G then we can easily inspect all conjugacy classes of subgroups of G. The following small GAP function can be used for that. It returns `false` if the table of marks of the group with the name `name` is not available, and the list `[ name, n, super ]` otherwise, where `n` is the maximal order of solvable subgroups of G, and `super` is a list of lists; for each conjugacy class of solvable subgroups S of order n, `super` contains the list of orders of representatives M of the classes of maximal subgroups of G such that M contains a conjugate of S.

Note that a subgroup in the i-th class of a table of marks contains a subgroup in the j-th class if and only if the entry in the position (i, j) of the table of marks is nonzero. For tables of marks objects in GAP, this is the case if and only if j is contained in the i-th row of the list that is stored as the value of the attribute `SubsTom` of the table of marks object; for this test, one need not unpack the matrix of marks.

```
  ──────── Example ────────
gap> MaximalSolvableSubgroupInfoFromTom:= function( name )
>     local tom,          # table of marks for 'name'
>           n,            # maximal order of a solvable subgroup
>           maxsubs,      # numbers of the classes of subgroups of order 'n'
>           orders,       # list of orders of the classes of subgroups
>           i,            # loop over the classes of subgroups
>           maxes,        # list of positions of the classes of max. subgroups
>           subs,         # 'SubsTom' value
>           cont;         # list of list of positions of max. subgroups
>
>     tom:= TableOfMarks( name );
>     if tom = fail then
>       return false;
>     fi;
>     n:= 1;
>     maxsubs:= [];
>     orders:= OrdersTom( tom );
>     for i in [ 1 .. Length( orders ) ] do
>       if IsSolvableTom( tom, i ) then
>         if orders[i] = n then
>           Add( maxsubs, i );
>         elif orders[i] > n then
>           n:= orders[i];
>           maxsubs:= [ i ];
>         fi;
>       fi;
>     od;
>     maxes:= MaximalSubgroupsTom( tom )[1];
>     subs:= SubsTom( tom );
>     cont:= List( maxsubs, j -> Filtered( maxes, i -> j in subs[i] ) );
>
>     return [ name, n, List( cont, l -> orders{ l } ) ];
> end;;
```

6.2.2 Use Information from the Character Table Library

The GAP Character Table Library contains the character tables of all maximal subgroups of sporadic simple groups, except for the Monster group. This information can be used as follows.

   We start, for a sporadic simple group G, with a known solvable subgroup of order n, say, in G. In order to show that G contains no solvable subgroup of larger order, it suffices to show that no maximal subgroup of G contains a larger solvable subgroup.

   The point is that usually the orders of the maximal subgroups of G are not much larger than n, and that a maximal subgroup M contains a solvable subgroup of order n only if the factor group of M by its largest solvable normal subgroup N contains a solvable subgroup of order $n/|N|$. This reduces the question to relatively small groups.

   What we can check *automatically* from the character table of M/N is whether M/N can contain subgroups (solvable or not) of indices between five and $|M|/n$, by computing possible permutation characters of these degrees. (Note that a solvable subgroup of a nonsolvable group has index at least five. This lower bound could be improved for example by considering the smallest degree of a non-

trivial character, but this is not an issue here.)

Then we are left with a –hopefully short– list of maximal subgroups of *G*, together with upper bounds on the indices of possible solvable subgroups; excluding these possibilities then yields that the initially chosen solvable subgroup of *G* is indeed the largest one.

The following GAP function can be used to compute this information for the character table `tblM` of *M* and a given order `minorder`. It returns `false` if *M* cannot contain a solvable subgroup of order at least `minorder`, otherwise a list `[ tblM, m, k ]` where `m` is the maximal index of a subgroup that has order at least `minorder`, and `k` is the minimal index of a possible subgroup of *M* (a proper subgroup if *M* is nonsolvable), according to the GAP function `PermChars` (**Reference: PermChars**).

```
 ─────────────────────────── Example ───────────────────────────
 gap> SolvableSubgroupInfoFromCharacterTable:= function( tblM, minorder )
 >     local maxindex,  # index of subgroups of order 'minorder'
 >           N,         # class positions describing a solvable normal subgroup
 >           fact,      # character table of the factor by 'N'
 >           classes,   # class sizes in 'fact'
 >           nsg,       # list of class positions of normal subgroups
 >           i;         # loop over the possible indices
 >
 >     maxindex:= Int( Size( tblM ) / minorder );
 >     if   maxindex = 0 then
 >       return false;
 >     elif IsSolvableCharacterTable( tblM ) then
 >       return [ tblM, maxindex, 1 ];
 >     elif maxindex < 5 then
 >       return false;
 >     fi;
 >
 >     N:= [ 1 ];
 >     fact:= tblM;
 >     repeat
 >       fact:= fact / N;
 >       classes:= SizesConjugacyClasses( fact );
 >       nsg:= Difference( ClassPositionsOfNormalSubgroups( fact ), [ [ 1 ] ] );
 >       N:= First( nsg, x -> IsPrimePowerInt( Sum( classes{ x } ) ) );
 >     until N = fail;
 >
 >     for i in Filtered( DivisorsInt( Size( fact ) ),
 >                        d -> 5 <= d and d <= maxindex ) do
 >       if Length( PermChars( fact, rec( torso:= [ i ] ) ) ) > 0 then
 >         return [ tblM, maxindex, i ];
 >       fi;
 >     od;
 >
 >     return false;
 > end;;
```

6.3   Cases where the Table of Marks is available in GAP

For twelve sporadic simple groups, the GAP library of Tables of Marks knows the tables of marks, so we can use `MaximalSolvableSubgroupInfoFromTom`.

```
                            ─── Example ───
gap> solvinfo:= Filtered( List(
>         AllCharacterTableNames( IsSporadicSimple, true,
>                                 IsDuplicateTable, false ),
>         MaximalSolvableSubgroupInfoFromTom ), x -> x <> false );;
gap> for entry in solvinfo do
>       MaxSolv.( entry[1] ):= entry[2];
>    od;
gap> for entry in solvinfo do
>       Print( String( entry[1], 5 ), String( entry[2], 7 ),
>              String( entry[3], 28 ), "\n" );
>    od;
  Co3  69984      [ [ 3849120, 699840 ] ]
   HS   2000      [ [ 252000, 252000 ] ]
   He  13824  [ [ 138240 ], [ 138240 ] ]
   J1    168                [ [ 168 ] ]
   J2   1152               [ [ 1152 ] ]
   J3   1944               [ [ 1944 ] ]
  M11    144                [ [ 144 ] ]
  M12    432       [ [ 432 ], [ 432 ] ]
  M22    576               [ [ 5760 ] ]
  M23   1152        [ [ 40320, 5760 ] ]
  M24  13824            [ [ 138240 ] ]
  McL  11664    [ [ 3265920, 58320 ] ]
```

We see that for J_1, J_2, J_3, M_{11}, and M_{12}, the subgroup S is maximal. For M_{12} and He, there are two classes of subgroups S. For the other groups, the class of subgroups S is unique, and there are one or two classes of maximal subgroups of G that contain S. From the shown orders of these maximal subgroups, their structures can be read off from the Atlas, on the pages listed in Table II.

Similarly, the Atlas tells us about the extensions of the subgroups S in $\mathrm{Aut}(G)$. In particular,

- the order $2\,000$ subgroups of HS are contained in maximal subgroups of the type $U_3(5).2$ (two classes) which do not extend to $HS.2$, but there are novelties of the type $5^{1+2}_+ : [2^5]$ and of the order $4\,000$, so the solvable subgroups of maximal order in HS do in fact extend to $HS.2$.

- the order $13\,824$ subgroups of He are contained in maximal subgroups of the type $2^6 : 3S_6$ (two classes) which do not extend to $He.2$, but there are novelties of the type $2^{4+4}.(S_3 \times S_3).2$ and of the order $18\,432$. (So the solvable subgroups S of maximal order in He do not extend to $He.2$ but there are larger solvable subgroups in $He.2$.)

  We inspect the maximal subgroups of $He.2$ in order to show that these are in fact the solvable subgroups of maximal order (see [CCN$^+$85, p. 104]): Any other solvable subgroup of order at least n in $He.2$ must be contained in a subgroup of one of the types $S_4(4).4$ (of index at most 212), $2^2.L_3(4).D_{12}$ (of index at most 52), or $2^{1+6}_+.L_3(2).2$ (of index at most 2). By [CCN$^+$85, pp. 44, 23, 3], this is not the case.

- the maximal subgroups of order $1\,152$ in J_2 extend to subgroups of order $2\,304$ in $J_2.2$.

- the maximal subgroups of order $1\,944$ in J_3 extend to subgroups of the type $3^2.3^{1+2}_+ : 8.2$ and of order 3888 in $J_3.2$. (The structure stated in [CCN$^+$85, p. 82] is not correct, see [BN95].)

- the maximal subgroups of order 432 in M_{12} (two classes) do *not* extend in $M_{12}.2$, and we see from the table of marks of $M_{12}.2$ that there are no larger solvable subgroups in this group, i. e., the solvable subgroups of maximal order in $M_{12}.2$ lie in M_{12}.

- the order 576 subgroups of M_{22} are contained in maximal subgroups of the type $2^4 : A_6$ which extend to subgroups of the type $2^4 : S_6$ in $M_{22}.2$, so the solvable subgroups of maximal order in $M_{22}.2$ have the type $2^4 : 3^2 : D_8$ and the order 1 152. In fact the structure is $S_4 \wr S_2$.

- the order 11 664 subgroups of McL are contained in maximal subgroups of the type $3^{1+4}_+ : 2S_5$ which extend to subgroups of the type $3^{1+4} : 4S_5$ in $McL.2$, so the solvable subgroups of maximal order in $McL.2$ have the type $3^{1+4} : 4S_4$ and the order 23 328.

```
_____ Example _____
gap> MaxSolv.( "HS.2" ):= 2 * MaxSolv.( "HS" );;
gap> n:= 2^(4+4) * ( 6 * 6 ) * 2;  MaxSolv.( "He.2" ):= n;;
18432
gap> List( [ Size( CharacterTable( "S4(4).4" ) ),
>           Factorial( 5 )^2 * 2,
>           Size( CharacterTable( "2^2.L3(4).D12" ) ),
>           2^7 * Size( CharacterTable( "L3(2)" ) ) * 2,
>           7^2 * 2 * Size( CharacterTable( "L2(7)" ) ) * 2,
>           3 * Factorial( 7 ) * 2 ], i -> Int( i / n ) );
[ 212, 1, 52, 2, 1, 1 ]
gap> MaxSolv.( "J2.2" ):= 2 * MaxSolv.( "J2" );;
gap> MaxSolv.( "J3.2" ):= 2 * MaxSolv.( "J3" );;
gap> info:= MaximalSolvableSubgroupInfoFromTom( "M12.2" );
[ "M12.2", 432, [ [ 95040 ] ] ]
gap> MaxSolv.( "M12.2" ):= info[2];;
gap> MaxSolv.( "M22.2" ):= 2 * MaxSolv.( "M22" );;
gap> MaxSolv.( "McL.2" ):= 2 * MaxSolv.( "McL" );;
```

6.4   Cases where the Table of Marks is not available in GAP

We use the GAP function `SolvableSubgroupInfoFromCharacterTable`, and individual arguments. In several cases, information about smaller sporadic simple groups is needed, so we deal with the groups in increasing order.

6.4.1 $G = Ru$

The group Ru contains exactly two conjugacy classes of nonisomorphic solvable subgroups of order $n = 49\,152$, and no larger solvable subgroups.

```
_____ Example _____
gap> t:= CharacterTable( "Ru" );;
gap> mx:= List( Maxes( t ), CharacterTable );;
gap> n:= 49152;;
gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
gap> info:= Filtered( info, IsList );
[ [ CharacterTable( "2^3+8:L3(2)" ), 7, 7 ],
  [ CharacterTable( "2.2^4+6:S5" ), 5, 5 ] ]
```

The maximal subgroups of the structure $2.2^{4+6} : S_5$ in *Ru* contain one class of solvable subgroups of order n and with the structure $2.2^{4+6} : S_4$, see [CCN+85, p. 126, p. 2].

The maximal subgroups of the structure $2^{3+8} : L_3(2)$ in *Ru* contain two classes of solvable subgroups of order n and with the structure $2^{3+8} : S_4$, see [CCN+85, p. 126, p. 3]. These groups are the stabilizers of vectors and two-dimensional subspaces, respectively, in the three-dimensional submodule; note that each $2^{3+8} : L_3(2)$ type subgroup H of *Ru* is the normalizer of an elementary abelian group of order eight all of whose involutions are in the *Ru*-class 2A and are conjugate in H. Since the $2.2^{4+6} : S_5$ type subgroups of *Ru* are the normalizers of 2A-elements in *Ru*, the groups in one of the two classes in question coincide with the largest solvable subgroups in the $2.2^{4+6} : S_5$ type subgroups. The groups in the other class do not centralize a 2A-element in *Ru* and are therefore not isomorphic with the $2.2^{4+6} : S_4$ type groups.

```
                         ──────────── Example ────────────
  gap> MaxSolv.( "Ru" ):= n;;
  gap> s:= info[1][1];;
  gap> cls:= SizesConjugacyClasses( s );;
  gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( s ),
  >                    x -> Sum( cls{ x } ) = 2^3 );
  [ [ 1, 2 ] ]
  gap> cls{ nsg[1] };
  [ 1, 7 ]
  gap> GetFusionMap( s, t ){ nsg[1] };
  [ 1, 2 ]
```

6.4.2 $G = Suz$

The group *Suz* contains a unique conjugacy class of solvable subgroups of order $n = 139\,968$, and no larger solvable subgroups.

```
                         ──────────── Example ────────────
  gap> t:= CharacterTable( "Suz" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 139968;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "G2(4)" ), 1797, 416 ],
    [ CharacterTable( "3_2.U4(3).2_3'" ), 140, 72 ],
    [ CharacterTable( "3^5:M11" ), 13, 11 ],
    [ CharacterTable( "2^4+6:3a6" ), 7, 6 ],
    [ CharacterTable( "3^2+4:2(2^2xa4)2" ), 1, 1 ] ]
```

The maximal subgroups S of the structure $3^{2+4} : 2(A_4 \times 2^2).2$ in *Suz* are solvable and have order n, see [CCN+85, p. 131].

In order to show that *Suz* contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $G_2(4)$ of index at most $1\,797$ (see [CCN+85, p. 97]), in $U_4(3).2_3'$ of index at most 140 (see [CCN+85, p. 52]), in M_{11} of index at most 13 (see [CCN+85, p. 18]), and in A_6 of index at most 7 (see [CCN+85, p. 4]).

The group S extends to a group of the structure $3^{2+4} : 2(S_4 \times D_8)$ in the automorphism group *Suz.2*.

```
                         ──────────── Example ────────────
  gap> MaxSolv.( "Suz" ):= n;;
  gap> MaxSolv.( "Suz.2" ):= 2 * n;;
```

6.4.3 $G = ON$

The group *ON* contains a unique conjugacy class of solvable subgroups of order $25\,920$, and no larger solvable subgroups.

```
─────────────────────────── Example ───────────────────────────
  gap> t:= CharacterTable( "ON" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 25920;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "L3(7).2" ), 144, 114 ],
    [ CharacterTable( "ONM2" ), 144, 114 ],
    [ CharacterTable( "3^4:2^(1+4)D10" ), 1, 1 ] ]
```

The maximal subgroups *S* of the structure $3^4 : 2^{1+4}_{-} D_{10}$ in *ON* are solvable and have order *n*, see [CCN+85, pp. 132].

In order to show that *ON* contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $L_3(7).2$ of index at most 144 (see [CCN+85, p. 50]); note that the groups in the second class of maximal subgroups of *ON* are isomorphic with $L_3(7).2$.

The group *S* extends to a group of order $|S.2|$ in the automorphism group *ON*.2.

```
─────────────────────────── Example ───────────────────────────
  gap> MaxSolv.( "ON" ):= n;;
  gap> MaxSolv.( "ON.2" ):= 2 * n;;
```

6.4.4 $G = Co_2$

The group Co_2 contains a unique conjugacy class of solvable subgroups of order $2\,359\,296$, and no larger solvable subgroups.

```
─────────────────────────── Example ───────────────────────────
  gap> t:= CharacterTable( "Co2" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 2359296;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "U6(2).2" ), 7796, 672 ],
    [ CharacterTable( "2^10:m22:2" ), 385, 22 ],
    [ CharacterTable( "McL" ), 380, 275 ],
    [ CharacterTable( "2^1+8:s6f2" ), 315, 28 ],
    [ CharacterTable( "2^1+4+6.a8" ), 17, 8 ],
    [ CharacterTable( "U4(3).D8" ), 11, 8 ],
    [ CharacterTable( "2^(4+10)(S5xS3)" ), 5, 5 ] ]
```

The maximal subgroups of the structure $2^{4+10}(S_5 \times S_3)$ in Co_2 contain solvable subgroups *S* of order *n* and with the structure $2^{4+10}(S_4 \times S_3)$, see [CCN+85, p. 154].

The subgroups *S* are contained also in the maximal subgroups of the type $2^{1+8}_{+} : S_6(2)$; note that the $2^{1+8}_{+} : S_6(2)$ type subgroups are described as normalizers of elements in the Co_2-class 2A, and *S* normalizes an elementary abelian group of order 16 containing an *S*-class of length five that is contained in the Co_2-class 2A.

```
                          ─── Example ───
  gap> s:= info[7][1];
  CharacterTable( "2^(4+10)(S5xS3)" )
  gap> cls:= SizesConjugacyClasses( s );;
  gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( s ),
  >                    x -> Sum( cls{ x } ) = 2^4 );
  [ [ 1 .. 3 ] ]
  gap> cls{ nsg[1] };
  [ 1, 5, 10 ]
  gap> GetFusionMap( s, t ){ nsg[1] };
  [ 1, 2, 3 ]
```

The stabilizers of these involutions in $2^{4+10}(S_5 \times S_3)$ have index five, they are solvable, and they are contained in $2^{1+8}_+ : S_6(2)$ type subgroups, so they are Co_2-conjugates of S. (The corresponding subgroups of $S_6(2)$ are maximal and have the type $2.[2^6] : (S_3 \times S_3)$.)

In order to show that G contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $U_6(2)$ of index at most 7796 (see [CCN+85, p. 115]), in $M_{22}.2$ of index at most 385 (see [CCN+85, p. 39] or Section 6.3), in McL of index at most 380 (see [CCN+85, p. 100] or Section 6.3), in A_8 of index at most 17 (see [CCN+85, p. 20]), and in $U_4(3).D_8$ of index at most 11 (see [CCN+85, p. 52]).

```
                          ─── Example ───
  gap> MaxSolv.( "Co2" ):= n;;
```

6.4.5 $G = Fi_{22}$

The group Fi_{22} contains a unique conjugacy class of solvable subgroups of order $5\,038\,848$, and no larger solvable subgroups.

```
                          ─── Example ───
  gap> t:= CharacterTable( "Fi22" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 5038848;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "2.U6(2)" ), 3650, 672 ],
    [ CharacterTable( "O7(3)" ), 910, 351 ],
    [ CharacterTable( "Fi22M3" ), 910, 351 ],
    [ CharacterTable( "O8+(2).3.2" ), 207, 6 ],
    [ CharacterTable( "2^10:m22" ), 90, 22 ],
    [ CharacterTable( "3^(1+6):2^(3+4):3^2:2" ), 1, 1 ] ]
```

The maximal subgroups S of the structure $3^{1+6} : 2^{3+4} : 3^2 : 2$ in Fi_{22} are solvable and have order n, see [CCN+85, p. 163].

In order to show that Fi_{22} contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $U_6(2)$ of index at most 3650 (see [CCN+85, p. 115]), in $O_7(3)$ of index at most 910 (see [CCN+85, p. 109]), in $O_8^+(2).S_3$ of index at most 207 (see [CCN+85, p. 85]), and in $M_{22}.2$ of index at most 90 (see [CCN+85, p. 39] or Section 6.3); note that the groups in the third class of maximal subgroups of Fi_{22} are isomorphic with $O_7(3)$.

The group S extends to a group of order $|S.2|$ in the automorphism group $Fi_{22}.2$.

```
                          ─── Example ───
  gap> MaxSolv.( "Fi22" ):= n;;
  gap> MaxSolv.( "Fi22.2" ):= 2 * n;;
```

6.4.6 $G = HN$

The group *HN* contains a unique conjugacy class of solvable subgroups of order $2\,000\,000$, and no larger solvable subgroups.

```
                          ─── Example ───
  gap> t:= CharacterTable( "HN" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 2000000;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "A12" ), 119, 12 ],
    [ CharacterTable( "5^(1+4):2^(1+4).5.4" ), 1, 1 ] ]
```

The maximal subgroups *S* of the structure $5^{1+4} : 2^{1+4}.5.4$ in *HN* are solvable and have order *n*, see [CCN$^+$85, p. 166].

In order to show that *HN* contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in A_{12} of index at most 119 (see [CCN$^+$85, p. 91]).

The group *S* extends to a group of order $|S.2|$ in the automorphism group *HN*.2.

```
                          ─── Example ───
  gap> MaxSolv.( "HN" ):= n;;
  gap> MaxSolv.( "HN.2" ):= 2 * n;;
```

6.4.7 $G = Ly$

The group *Ly* contains a unique conjugacy class of solvable subgroups of order $900\,000$, and no larger solvable subgroups.

```
                          ─── Example ───
  gap> t:= CharacterTable( "Ly" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 900000;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "G2(5)" ), 6510, 3906 ],
    [ CharacterTable( "3.McL.2" ), 5987, 275 ],
    [ CharacterTable( "5^3.psl(3,5)" ), 51, 31 ],
    [ CharacterTable( "2.A11" ), 44, 11 ],
    [ CharacterTable( "5^(1+4):4S6" ), 10, 6 ] ]
```

The maximal subgroups of the structure $5^{(1+4)} : 4S6$ in *Ly* contain solvable subgroups *S* of order *n* and with the structure $5^{1+4} : 4.3^2.D_8$, see [CCN$^+$85, p. 174].

In order to show that *Ly* contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $G_2(5)$ of index at most 6510 (see [CCN$^+$85, p. 114]), in *McL*.2 of index at most 5987 (see [CCN$^+$85, p. 100] or Section 6.3), in $L_3(5)$ of index at most 51 (see [CCN$^+$85, p. 38]), and in A_{11} of index at most 44 (see [CCN$^+$85, p. 75]).

```
                          ─── Example ───
  gap> MaxSolv.( "Ly" ):= n;;
```

6.4.8 $G = Th$

The group Th contains exactly two conjugacy classes of nonisomorphic solvable subgroups of order $n = 944\,784$, and no larger solvable subgroups.

```
─────────────────────── Example ───────────────────────
  gap> t:= CharacterTable( "Th" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 944784;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "2^5.psl(5,2)" ), 338, 31 ],
    [ CharacterTable( "2^1+8.a9" ), 98, 9 ],
    [ CharacterTable( "U3(8).6" ), 35, 6 ],
    [ CharacterTable( "ThN3B" ), 1, 1 ],
    [ CharacterTable( "ThM7" ), 1, 1 ] ]
```

The maximal subgroups S of the structures $[3^9].2S_4$ and $3^2.[3^7].2S_4$ in Th are solvable and have order n, see [CCN$^+$85, p. 177].

In order to show that Th contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $L_5(2)$ of index at most 338 (see [CCN$^+$85, p. 70]), in A_9 of index at most 98 (see [CCN$^+$85, p. 37]), and in $U_3(8).6$ of index at most 35 (see [CCN$^+$85, p. 66]).

```
─────────────────────── Example ───────────────────────
  gap> MaxSolv.( "Th" ):= n;;
```

6.4.9 $G = Fi_{23}$

The group Fi_{23} contains a unique conjugacy class of solvable subgroups of order $n = 3\,265\,173\,504$, and no larger solvable subgroups.

```
─────────────────────── Example ───────────────────────
  gap> t:= CharacterTable( "Fi23" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 3265173504;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "2.Fi22" ), 39545, 3510 ],
    [ CharacterTable( "O8+(3).3.2" ), 9100, 6 ],
    [ CharacterTable( "3^(1+8).2^(1+6).3^(1+2).2S4" ), 1, 1 ] ]
```

The maximal subgroups S of the structure $3^{1+8}_+.2^{1+6}_-.3^{1+2}_+.2S_4$ in Fi_{23} are solvable and have order n, see [CCN$^+$85, p. 177].

In order to show that Fi_{23} contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in Fi_{22} of index at most $39\,545$ (see Section 6.4.5) and in $O_8^+(3).S_3$ of index at most $9\,100$ (see [CCN$^+$85, p. 140]).

```
─────────────────────── Example ───────────────────────
  gap> MaxSolv.( "Fi23" ):= n;;
```

6.4.10 $G = Co_1$

The group Co_1 contains a unique conjugacy class of solvable subgroups of order $n = 84\,934\,656$, and no larger solvable subgroups.

```
──────────────────────────── Example ────────────────────────────
  gap> t:= CharacterTable( "Co1" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 84934656;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "Co2" ), 498093, 2300 ],
    [ CharacterTable( "3.Suz.2" ), 31672, 1782 ],
    [ CharacterTable( "2^11:M24" ), 5903, 24 ],
    [ CharacterTable( "Co3" ), 5837, 276 ],
    [ CharacterTable( "2^(1+8)+.O8+(2)" ), 1050, 120 ],
    [ CharacterTable( "U6(2).3.2" ), 649, 6 ],
    [ CharacterTable( "2^(2+12):(A8xS3)" ), 23, 8 ],
    [ CharacterTable( "2^(4+12).(S3x3S6)" ), 10, 6 ] ]
```

The maximal subgroups of the structure $2^{4+12}.(S_3 \times 3S_6)$ in Co_1 contain solvable subgroups S of order n and with the structure $2^{4+12}.(S_3 \times 3^{1+2}_+ : D_8)$, see [CCN$^+$85, p. 183].

In order to show that Co_1 contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in Co_2 of index at most 498093 (see Section 6.4.4), in *Suz*.2 of index at most 31672 (see Section 6.4.2), in M_{24} of index at most 5903 (see Section 6.3), in Co_3 of index at most 5837 (see [CCN$^+$85, p. 134] or Section 6.3), in $O_8^+(2)$ of index at most 1050 (see [CCN$^+$85, p. 185]), in $U_6(2).S_3$ of index at most 649 (see [CCN$^+$85, p. 115]), and in A_8 of index at most 23 (see [CCN$^+$85, p. 22]).

```
──────────────────────────── Example ────────────────────────────
  gap> MaxSolv.( "Co1" ):= n;;
```

6.4.11 $G = J_4$

The group J_4 contains a unique conjugacy class of solvable subgroups of order $28\,311\,552$, and no larger solvable subgroups.

```
──────────────────────────── Example ────────────────────────────
  gap> t:= CharacterTable( "J4" );;
  gap> mx:= List( Maxes( t ), CharacterTable );;
  gap> n:= 28311552;;
  gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
  gap> info:= Filtered( info, IsList );
  [ [ CharacterTable( "mx1j4" ), 17710, 24 ],
    [ CharacterTable( "c2aj4" ), 770, 22 ],
    [ CharacterTable( "2^10:L5(2)" ), 361, 31 ],
    [ CharacterTable( "J4M4" ), 23, 5 ] ]
```

The maximal subgroups of the structure $2^{11} : M_{24}$ in J_4 contain solvable subgroups S of order n and with the structure $2^{11} : 2^6 : 3^{1+2}_+ : D_8$, see Section 6.3 and [CCN$^+$85, p. 190].

(The subgroups in the first four classes of maximal subgroups of J_4 have the structures $2^{11} : M_{24}$, $2^{1+12}_+.3M_{22} : 2$, $2^{10} : L_5(2)$, and $2^{3+12}.(S_5 \times L_3(2))$, in this order.)

The subgroups S are contained also in the maximal subgroups of the type $2_+^{1+12}.3M_{22}:2$; note that these subgroups are described as normalizers of elements in the J_4-class 2A, and S normalizes an elementary abelian group of order 2^{11} containing an S-class of length 1771 that is contained in the J_4-class 2A.

```
─────────────── Example ───────────────
gap> s:= info[1][1];
CharacterTable( "mx1j4" )
gap> cls:= SizesConjugacyClasses( s );;
gap> nsg:= Filtered( ClassPositionsOfNormalSubgroups( s ),
>                    x -> Sum( cls{ x } ) = 2^11 );
[ [ 1 .. 3 ] ]
gap> cls{ nsg[1] };
[ 1, 276, 1771 ]
gap> GetFusionMap( s, t ){ nsg[1] };
[ 1, 3, 2 ]
```

The stabilizers of these involutions in $2^{11}:M_{24}$ have index 1771, they have the structure $2^{11}:2^6:3.S_6$, and they are contained in $2_+^{1+12}.3M_{22}:2$ type subgroups; so also S, which has index 10 in $2^{11}:2^6:3.S_6$, is contained in $2_+^{1+12}.3M_{22}:2$. (The corresponding subgroups of $M_{22}:2$ are of course the solvable groups of maximal order described in Section 6.3.)

In order to show that G contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in $L_5(2)$ of index at most 361 (see [CCN⁺85, p. 70]) and in $S_5 \times L_3(2)$ of index at most 23 (see [CCN⁺85, pp. 2, 3]).

```
─────────────── Example ───────────────
gap> MaxSolv.( "J4" ):= n;;
```

6.4.12 $G = Fi'_{24}$

The group Fi'_{24} contains a unique conjugacy class of solvable subgroups of order $29\,386\,561\,536$, and no larger solvable subgroups.

```
─────────────── Example ───────────────
gap> t:= CharacterTable( "Fi24'" );;
gap> mx:= List( Maxes( t ), CharacterTable );;
gap> n:= 29386561536;;
gap> info:= List( mx, x -> SolvableSubgroupInfoFromCharacterTable( x, n ) );;
gap> info:= Filtered( info, IsList );
[ [ CharacterTable( "Fi23" ), 139161244, 31671 ],
  [ CharacterTable( "2.Fi22.2" ), 8787, 3510 ],
  [ CharacterTable( "(3xO8+(3):3):2" ), 3033, 6 ],
  [ CharacterTable( "O10-(2)" ), 851, 495 ],
  [ CharacterTable( "3^(1+10):U5(2):2" ), 165, 165 ],
  [ CharacterTable( "2^2.U6(2).3.2" ), 7, 6 ] ]
```

The maximal subgroups of the structure $3_+^{1+10}:U5(2):2$ in Fi'_{24} contain solvable subgroups S of order n and with the structure $3_+^{1+10}:2_-^{1+6}:3_+^{1+2}:2S_4$, see [CCN⁺85, p. 73, p. 207].

In order to show that G contains no other solvable subgroups of order larger than or equal to $|S|$, we check that there are no solvable subgroups in Fi_{23} of order at least n (see Section 6.4.9), in $Fi_{22}.2$ of order at least n (see Section 6.4.5), in $O_8^+(3).S_3$ of index at most 3033 (see [CCN⁺85, p.

140]), in $O_{10}^-(2)$ of index at most 851 (see [CCN$^+$85, p. 147]), and in $U_6(2).S_3$ of index at most 7 (see [CCN$^+$85, p. 115]).

The group S extends to a group of order $|S.2|$ in the automorphism group Fi_{24}.

```
                         ──── Example ────
  gap> MaxSolv.( "Fi24'" ):= n;;
  gap> MaxSolv.( "Fi24'.2" ):= 2 * n;;
```

6.4.13 $G = B$

The group B contains a unique conjugacy class of solvable subgroups of order $n = 29\,686\,813\,949\,952$, and no larger solvable subgroups.

The maximal subgroups of the structure $2^{2+10+20}(M_{22} : 2 \times S_3)$ in B contain solvable subgroups S of order n and with the structure $2^{2+10+20}(2^4 : 3^2 : D_8 \times S_3)$, see [CCN$^+$85, p. 217] and Section 6.3.

```
                         ──── Example ────
  gap> n:= 29686813949952;;
  gap> n = 2^(2+10+20) * 2^4 * 3^2 * 8 * 6;
  true
  gap> n = 2^(2+10+20) * MaxSolv.( "M22.2" ) * 6;
  true
```

By [Wil99, Table 1], the only maximal subgroups of B of order bigger than $|S|$ have the following structures.

$$
\begin{array}{llll}
2.^2E_6(2).2 & 2^{1+22}.Co_2 & Fi_{23} & 2^{9+16}S_8(2) \\
Th & (2^2 \times F_4(2)) : 2 & 2^{2+10+20}(M_{22} : 2 \times S_3) & 2^{5+5+10+10}L_5(2) \\
S_3 \times Fi_{22} : 2 & 2^{[35]}(S_5 \times L_3(2)) & HN : 2 & O_8^+(3) : S_4
\end{array}
$$

(The character tables of the maximal subgroups of B are meanwhile available in GAP.)

```
                         ──── Example ────
  gap> b:= CharacterTable( "B" );;
  gap> mx:= List( Maxes( b ), CharacterTable );;
  gap> Filtered( mx, x -> Size( x ) >= n );
  [ CharacterTable( "2.2E6(2).2" ), CharacterTable( "2^(1+22).Co2" ),
    CharacterTable( "Fi23" ), CharacterTable( "2^(9+16).S8(2)" ),
    CharacterTable( "Th" ), CharacterTable( "(2^2xF4(2)):2" ),
    CharacterTable( "2^(2+10+20).(M22.2xS3)" ),
    CharacterTable( "[2^30].L5(2)" ), CharacterTable( "S3xFi22.2" ),
    CharacterTable( "[2^35].(S5xL3(2))" ), CharacterTable( "HN.2" ),
    CharacterTable( "O8+(3).S4" ) ]
```

For the subgroups $2^{1+22}.Co_2$, Fi_{23}, Th, $S_3 \times Fi_{22} : 2$, and $HN : 2$, the solvable subgroups of maximal order are known from the previous sections or can be derived from known values, and are smaller than n.

```
                         ──── Example ────
  gap> List( [ 2^(1+22) * MaxSolv.( "Co2" ),
  >            MaxSolv.( "Fi23" ),
  >            MaxSolv.( "Th" ),
  >            6 * MaxSolv.( "Fi22.2" ),
  >            MaxSolv.( "HN.2" ) ], i -> Int( i / n ) );
  [ 0, 0, 0, 0, 0 ]
```

If one of the remaining maximal groups U from the above list has a solvable subgroup of order at least n then the index of this subgroup in U is bounded as follows.

```
──────────────────────── Example ────────────────────────
 gap> List( [ Size( CharacterTable( "2.2E6(2).2" ) ),
 >            2^(9+16) * Size( CharacterTable( "S8(2)" ) ),
 >            2^3 * Size( CharacterTable( "F4(2)" ) ),
 >            2^(2+10+20) * Size( CharacterTable( "M22.2" ) ) ) * 6,
 >            2^30 * Size( CharacterTable( "L5(2)" ) ),
 >            2^35 * Factorial(5) * Size( CharacterTable( "L3(2)" ) ) ),
 >            Size( CharacterTable( "O8+(3)" ) ) * 24 ],
 >         i -> Int( i / n ) );
 [ 10311982931, 53550, 892, 770, 361, 23, 4 ]
```

The group $O_8^+(3) : S_4$ is nonsolvable, and its order is less than $5n$, thus its solvable subgroups have orders less than n.

The largest solvable subgroup of $S_5 \times L_3(2)$ has index 35, thus the solvable subgroups of $2^{[35]}(S_5 \times L_3(2))$ have orders less than n.

The groups of type $2^{5+5+10+10}L_5(2)$ cannot contain solvable subgroups of order at least n because $L_5(2)$ has no solvable subgroup of index up to 361 –such a subgroup would be contained in $2^4 : L_4(2)$, of index at most $\lfloor 361/31 \rfloor = 11$ (see [CCN$^+$85, p. 70]), and $L_4(2) \cong A_8$ does not have such subgroups (see [CCN$^+$85, p. 22]).

The largest proper subgroup of $F_4(2)$ has index 69615 (see [CCN$^+$85, p. 170]), which excludes solvable subgroups of order at least n in $(2^2 \times F_4(2)) : 2$.

Ruling out the group $2.^2E_6(2).2$ is more involved. We consider the list of maximal subgroups of $^2E_6(2)$ in [CCN$^+$85, p. 191] (which is complete, see [BN95]), and compute the maximal index of a group of order $n/4$; the possible subgroups of $^2E_6(2)$ to consider are the following

$$2^{1+20} : U_6(2) \quad 2^{8+16} : O_8^-(2) \quad F_4(2) \qquad\qquad 2^2.2^9.2^{18} : (L_3(4) \times S_3)$$
$$Fi_{22} \qquad\quad O_{10}^-(2) \qquad 2^3.2^{12}.2^{15} : (S_5 \times L_3(2))$$

(The order of $S_3 \times U_6(2)$ is already smaller than $n/4$.)

```
──────────────────────── Example ────────────────────────
 gap> List( [ 2^(1+20) * Size( CharacterTable( "U6(2)" ) ),
 >            2^(8+16) * Size( CharacterTable( "O8-(2)" ) ),
 >            Size( CharacterTable( "F4(2)" ) ),
 >            2^(2+9+18) * Size( CharacterTable( "L3(4)" ) ) ) * 6,
 >            Size( CharacterTable( "Fi22" ) ),
 >            Size( CharacterTable( "O10-(2)" ) ),
 >            2^(3+12+15) * 120 * Size( CharacterTable( "L3(2)" ) ),
 >            6 * Size( CharacterTable( "U6(2)" ) ) ],
 >         i -> Int( i / ( n / 4 ) ) );
 [ 2598, 446, 446, 8, 8, 3, 2, 0 ]
```

The indices of the solvable groups of maximal orders in the groups $U_6(2)$, $O_8^-(2)$, $F_4(2)$, $L_3(4)$, and Fi_{22} are larger than the bounds we get for n, see [CCN$^+$85, pp. 115, 89, 170, 23, 163].

It remains to consider the subgroups of the type $2^{9+16}S_8(2)$. The group $S_8(2)$ contains maximal subgroups of the type $2^{3+8} : (S_3 \times S_6)$ and of index 5355 (see [CCN$^+$85, p. 123]), which contain solvable subgroups S' of index 10. This yields solvable subgroups of order $2^{9+16+3+8} \cdot 6 \cdot 72 = n$.

```
_____ Example _____
  gap> 2^(9+16+3+8) * 6 * 72 = n;
  true
```

There are no other solvable subgroups of larger or equal order in $S_8(2)$: We would need solvable subgroups of index at most 446 in $O_8^-(2) : 2$, 393 in $O_8^+(2) : 2$, 210 in $S_6(2)$, or 23 in A_8, which is not the case by [CCN+85, pp. 89, 85, 46, 22].

```
_____ Example _____
  gap> index:= Int( 2^(9+16) * Size( CharacterTable( "S8(2)" ) ) / n );
  53550
  gap> List( [ 120, 136, 255, 2295 ], i -> Int( index / i ) );
  [ 446, 393, 210, 23 ]
  gap> MaxSolv.( "B" ):= n;;
```

So the $2^{9+16}S_8(2)$ type subgroups of B yield solvable subgroups S' of the type $2^{9+16}.2^{3+8} : (S_3 \times 3^2 : D_8)$, and of order n.

We want to show that S' is a B-conjugate of S. For that, we first show the following:

Lemma:

The group B contains exactly two conjugacy classes of Klein four groups whose involutions lie in the class 2B. (We will call these Klein four groups 2B-pure.) Their normalizers in B have the orders $22\,858\,846\,741\,463\,040$ and $292\,229\,574\,819\,840$, respectively.

Proof. Let V be a 2B-pure Klein four group in B, and set $N = N_B(V)$. Let $x \in V$ be an involution and set $H = C_B(x)$, then H is maximal in B and has the structure $2^{1+22}.Co_2$. The index of $C = C_B(V) = C_H(V)$ in N divides 6, and C stabilizes the central involution in H and another 2B involution. The group H contains exactly four conjugacy classes of 2B elements.

```
_____ Example _____
  gap> h:= mx[2];
  CharacterTable( "2^(1+22).Co2" )
  gap> pos:= Positions( GetFusionMap( h, b ), 3 );
  [ 2, 4, 11, 20 ]
```

The B-classes of 2B-pure Klein four groups arise from those of these classes $y^H \subset H$ such that $x \neq y$ holds and xy is a 2B element. We compute this subset.

```
_____ Example _____
  gap> pos:= Filtered( Difference( pos, [ 2 ] ), i -> ForAny( pos,
  >             j -> NrPolyhedralSubgroups( h, 2, i, j ).number <> 0 ) );
  [ 4, 11 ]
```

The two classes have lengths $93\,150$ and $7\,286\,400$, thus the index of C in H is one of these numbers.

```
_____ Example _____
  gap> SizesConjugacyClasses( h ){ pos };
  [ 93150, 7286400 ]
```

Next we compute the number n_0 of 2B-pure Klein four groups in B.

```
─────────────────────── Example ───────────────────────
gap> nr:= NrPolyhedralSubgroups( b, 3, 3, 3 );
rec( number := 14399283809600746875, type := "V4" )
gap> n0:= nr.number;;
```

The *B*-conjugacy class of *V* has length $[B:N] = [B:H] \cdot [H:C]/[N:C]$, where $[N:C]$ divides 6. We see that $[N:C] = 6$ in both cases.

```
─────────────────────── Example ───────────────────────
gap> cand:= List( pos, i -> Size( b ) / SizesCentralizers( h )[i] / 6 );
[ 181758140654146875, 14217525668946600000 ]
gap> Sum( cand ) = n0;
true
```

The orders of the normalizers of the two classes of 2B-pure Klein four groups are as claimed.

```
─────────────────────── Example ───────────────────────
gap> List( cand, x -> Size( b ) / x );
[ 22858846741463040, 292229574819840 ]
```

The subgroup *S* of order *n* is contained in a maximal subgroup *M* of the type $2^{2+10+20}(M_{22} : 2 \times S_3)$ in *B*. The group *M* is the normalizer of a 2B-pure Klein four group in *B*, and the other class of normalizers of 2B-pure Klein four groups does not contain subgroups of order *n*. Thus the conjugates of *S* are uniquely determined by $|S|$ and the property that they normalize 2B-pure Klein four groups.

```
─────────────────────── Example ───────────────────────
gap> m:= mx[7];
CharacterTable( "2^(2+10+20).(M22.2xS3)" )
gap> Size( m );
22858846741463040
gap> nsg:= ClassPositionsOfMinimalNormalSubgroups( m );
[ [ 1, 2 ] ]
gap> SizesConjugacyClasses( m ){ nsg[1] };
[ 1, 3 ]
gap> GetFusionMap( m, b ){ nsg[1] };
[ 1, 3 ]
gap> List( cand, x -> Size( b ) / ( n * x ) );
[ 770, 315/32 ]
```

Now consider the subgroup *S′* of order *n*, which is contained in a maximal subgroup of the type $2^{9+16}S_8(2)$ in *B*. In order to prove that *S′* is *B*-conjugate to *S*, it is enough to show that *S′* normalizes a 2B-pure Klein four group.

The unique minimal normal subgroup *V* of $2^{9+16}S_8(2)$ has order 2^8. Its involutions lie in the class 2B of *B*.

```
─────────────────────── Example ───────────────────────
gap> m:= mx[4];
CharacterTable( "2^(9+16).S8(2)" )
gap> nsg:= ClassPositionsOfMinimalNormalSubgroups( m );
[ [ 1, 2 ] ]
gap> SizesConjugacyClasses( m ){ nsg[1] };
[ 1, 255 ]
gap> GetFusionMap( m, b ){ nsg[1] };
[ 1, 3 ]
```

The group V is central in the normal subgroup $W = 2^{9+16}$, since all nonidentity elements of V lie in one conjugacy class of odd length. As a module for $S_8(2)$, V is the unique irreducible eight-dimensional module in characteristic two.

```
───────────────── Example ─────────────────
gap> CharacterDegrees( CharacterTable( "S8(2)" ) mod 2 );
[ [ 1, 1 ], [ 8, 1 ], [ 16, 1 ], [ 26, 1 ], [ 48, 1 ], [ 128, 1 ],
  [ 160, 1 ], [ 246, 1 ], [ 416, 1 ], [ 768, 1 ], [ 784, 1 ],
  [ 2560, 1 ], [ 3936, 1 ], [ 4096, 1 ], [ 12544, 1 ], [ 65536, 1 ] ] ]
```

Hence we are done if the restriction of the $S_8(2)$-action on V to S'/W leaves a two-dimensional subspace of V invariant. In fact we show that already the restriction of the $S_8(2)$-action on V to the maximal subgroups of the structure $2^{3+8} : (S_3 \times S_6)$ has a two-dimensional submodule.

These maximal subgroups have index 5355 in $S_8(2)$. The primitive permutation representation of degree 5355 of $S_8(2)$ and the irreducible eight-dimensional matrix representation of $S_8(2)$ over the field with two elements are available via the GAP package AtlasRep, see [WPN+22]. We compute generators for an index 5355 subgroup in the matrix group via an isomorphism to the permutation group.

```
───────────────── Example ─────────────────
gap> permg:= AtlasGroup( "S8(2)", NrMovedPoints, 5355 );
<permutation group of size 47377612800 with 2 generators>
gap> matg:= AtlasGroup( "S8(2)", Dimension, 8 );
<matrix group of size 47377612800 with 2 generators>
gap> hom:= GroupHomomorphismByImagesNC( matg, permg,
>           GeneratorsOfGroup( matg ), GeneratorsOfGroup( permg ) );;
gap> max:= PreImages( hom, Stabilizer( permg, 1 ) );;
```

These generators define the action of the index 5355 subgroup of $S_8(2)$ on the eight-dimensional module. We compute the dimensions of the factors of an ascending composition series of this module.

```
───────────────── Example ─────────────────
gap> m:= GModuleByMats( GeneratorsOfGroup( max ), GF(2) );;
gap> comp:= MTX.CompositionFactors( m );;
gap> List( comp, r -> r.dimension );
[ 2, 4, 2 ]
```

6.4.14 $G = M$

The group M contains exactly two conjugacy classes of solvable subgroups of order $n = 2\,849\,934\,139\,195\,392$, and no larger solvable subgroups.

The maximal subgroups of the structure $2^{1+24}_+.Co_1$ in the group M contain solvable subgroups S of order n and with the structure $2^{1+24}_+.2^{4+12}.(S_3 \times 3^{1+2}_+ : D_8)$, see [CCN+85, p. 234] and Section 6.4.10.

```
───────────────── Example ─────────────────
gap> n:= 2^25 * MaxSolv.( "Co1" );
2849934139195392
```

The solvable subgroups of maximal order in groups of the types $2^{2+11+22}.(M_{24} \times S_3)$ and $2^{[39]}.(L_3(2) \times 3S_6)$ have order n.

```
 ─────────────────────────── Example ───────────────────────────
  gap> 2^(2+11+22) * MaxSolv.( "M24" ) * 6 = n;
  true
  gap> 2^39 * 24 * 3 * 72 = n;
  true
```

For inspecting the other maximal subgroups of *M*, we use the description from [NW13], which lists 44 classes of maximal subgroups of *G*, and states that any possible other maximal subgroup of *G* has socle isomorphic to one of $L_2(13)$, $Sz(8)$, $U_3(4)$, $U_3(8)$; so these maximal subgroups are isomorphic to subgroups of the automorphism groups of these groups – the maximum of these group orders is smaller than *n*, hence we may ignore these possible subgroups.

```
 ─────────────────────────── Example ───────────────────────────
  gap> cand:= [ "L2(13)", "Sz(8)", "U3(4)", "U3(8)" ];;
  gap> List( cand, nam -> ExtensionInfoCharacterTable(
  > CharacterTable( nam ) ) );
  [ [ "2", "2" ], [ "2^2", "3" ], [ "", "4" ], [ "3", "(S3x3)" ] ]
  gap> ll:= List( cand, x -> Size( CharacterTable( x ) ) );
  [ 1092, 29120, 62400, 5515776 ]
  gap> 18 * ll[4];
  99283968
  gap> 2^39 * 24 * 3 * 72;
  2849934139195392
```

Remark added in December 2023: The classes of maximal subgroups of *G* are classified in [DLP25]. As a consequence, The result is that there are no maximal subgroups with socle $Sz(8)$ or $U_3(8)$, there is one class of maximal subgroups of each of the isomorphism types $L_2(13).2$ and $U_3(4).4$, and there are no subgroups o the type $L_2(59)$ (hence the subgroups of the type $59 : 29$ are maximal).

Thus only the following maximal subgroups of *M* have order bigger than $|S|$.

$$
\begin{array}{llll}
2.B & 2^{1+24}_{+}.Co_1 & 3.Fi_{24} & 2^2.{}^2E_6(2) : S_3 \\
2^{10+16}.O^+_{10}(2) & 2^{2+11+22}.(M_{24} \times S_3) & 3^{1+12}_{+}.2Suz.2 & 2^{5+10+20}.(S_3 \times L_5(2)) \\
S_3 \times Th & 2^{[39]}.(L_3(2) \times 3S_6) & 3^8.O^-_8(3).2_3 & (D_{10} \times HN).2
\end{array}
$$

For the subgroups $2.B$, $3.Fi_{24}$, $3^{1+12}_{+}.2Suz.2$, $S_3 \times Th$, and $(D_{10} \times HN).2$, the solvable subgroups of maximal order are smaller than *n*.

```
 ─────────────────────────── Example ───────────────────────────
  gap> List( [ 2 * MaxSolv.( "B" ),
  >            6 * MaxSolv.( "Fi24'" ),
  >            3^13 * 2 * MaxSolv.( "Suz" ) * 2,
  >            6 * MaxSolv.( "Th" ),
  >            10 * MaxSolv.( "HN" ) * 2 ], i -> Int( i / n ) );
  [ 0, 0, 0, 0, 0 ]
```

The subgroup $2^2.{}^2E_6(2) : S_3$ can be excluded by the fact that this group is only six times larger than the subgroup $2.{}^2E_6(2) : 2$ of *B*, but *n* is 96 times larger than the maximal solvable subgroup in *B*.

```
 ─────────────────────────── Example ───────────────────────────
  gap> n / MaxSolv.( "B" );
  96
```

The group $3^8.O_8^-(3).2_3$ can be excluded by the fact that a solvable subgroup of order at least n would imply the existence of a solvable subgroup of index at most 46 in $O_8^-(3).2_3$, which is not the case (see [CCN$^+$85, p. 141]).

```
——————————————— Example ———————————————
gap> Int( 3^8 * Size( CharacterTable( "O8-(3)" ) ) * 2 / n );
46
```

Similarly, the existence of a solvable subgroup of order at least n in $2^{5+10+20}.(S_3 \times L_5(2))$ would imply the existence of a solvable subgroup of index at most 723 in $L_5(2)$ and in turn of a solvable subgroup of index at most 23 in $L_4(2)$, which is not the case (see [CCN$^+$85, p. 70]).

```
——————————————— Example ———————————————
gap> Int( 2^(10+16) * Size( CharacterTable( "O10+(2)" ) ) / n );
553350
gap> Int( 2^(5+10+20) * 6 * Size( CharacterTable( "L5(2)" ) ) / n );
723
gap> Int( 723 / 31 );
23
```

It remains to exclude the subgroup $2^{10+16}.O_{10}^+(2)$, which means to show that $O_{10}^+(2)$ does not contain a solvable subgroup of index at most $553\,350$. If such a subgroup would exist then it would be contained in one of the following maximal subgroups of $O_{10}^+(2)$ (see [CCN$^+$85, p. 146]): in $S_8(2)$ (of index at most $1\,115$), in $2^8 : O_8^+(2)$ (of index at most $1\,050$), in $2^{10} : L_5(2)$ (of index at most 241), in $(3 \times O_8^-(2)) : 2$ (of index at most 27), in $(2_+^{1+12} : (S_3 \times A_8)$ (of index at most 23), or in $2^{3+12} : (S_3 \times S_3 \times L_3(2))$ (of index at most 4). By [CCN$^+$85, pp. 123, 85, 70, 89, 22], this is not the case.

```
——————————————— Example ———————————————
gap> index:= Int( 2^(10+16) * Size( CharacterTable( "O10+(2)" ) ) / n );
553350
gap> List( [ 496, 527, 2295, 19840, 23715, 118575 ], i -> Int( index / i ) );
[ 1115, 1050, 241, 27, 23, 4 ]
```

As a consequence, we have shown that the largest solvable subgroups of M have order n.

```
——————————————— Example ———————————————
gap> MaxSolv.( "M" ):= n;;
```

In order to prove the statement about the conjugacy of subgroups of order n in M, we first show the following.

Lemma:

The group M contains exactly three conjugacy classes of 2B-pure Klein four groups. Their normalizers in M have the orders $50\,472\,333\,605\,150\,392\,320$, $259\,759\,622\,062\,080$, and $9\,567\,039\,651\,840$, respectively.

Proof. The idea is the same as for the Baby Monster group, see Section 6.4.13. Let V be a 2B-pure Klein four group in M, and set $N = N_M(V)$. Let $x \in V$ be an involution and set $H = C_M(x)$, then H is maximal in M and has the structure $2_+^{1+24}.Co_1$. The index of $C = C_M(V) = C_H(V)$ in N divides 6, and C stabilizes the central involution in H and another 2B involution.

The group H contains exactly five conjugacy classes of 2B elements, three of them consist of elements that generate a 2B-pure Klein four group together with x.

```
gap> m:= CharacterTable( "M" );;
gap> h:= CharacterTable( "2^1+24.Co1" );
CharacterTable( "2^1+24.Co1" )
gap> pos:= Positions( GetFusionMap( h, m ), 3 );
[ 2, 4, 7, 9, 16 ]
gap> pos:= Filtered( Difference( pos, [ 2 ] ), i -> ForAny( pos,
>              j -> NrPolyhedralSubgroups( h, 2, i, j ).number <> 0 ) );
[ 4, 9, 16 ]
```

The two classes have lengths $93\,150$ and $7\,286\,400$, thus the index of C in H is one of these numbers.

```
gap> SizesConjugacyClasses( h ){ pos };
[ 16584750, 3222483264000, 87495303168000 ]
```

Next we compute the number n_0 of 2B-pure Klein four groups in M.

```
gap> nr:= NrPolyhedralSubgroups( m, 3, 3, 3 );
rec( number := 8756911006698538735755092552182824492187
5,
  type := "V4" )
gap> n0:= nr.number;;
```

The M-conjugacy class of V has length $[M : N] = [M : H] \cdot [H : C]/[N : C]$, where $[N : C]$ divides 6. We see that $[N : C] = 6$ in both cases.

```
gap> cand:= List( pos, i -> Size( m ) / SizesCentralizers( h )[i] / 6 );
[ 1600911562987568400634355094492187
5,
  3110635203347364905168577322802100000000,
  8445845885452239257669834185547520000000000 ]
gap> Sum( cand ) = n0;
true
```

The orders of the normalizers of the three classes of 2B-pure Klein four groups are as claimed.

```
gap> List( cand, x -> Size( m ) / x );
[ 50472333605150392320, 259759622062080, 9567039651840 ]
```

As we have seen above, the group M contains exactly the following (solvable) subgroups of order n.

1. One class in $2_+^{1+24}.Co_1$ type subgroups,

2. one class in $2^{2+11+22}.(M_{24} \times S_3)$ type subgroups, and

3. two classes in $2^{[39]}.(L_3(2) \times 3S_6)$ type subgroups.

Note that $2^{[39]}.(L_3(2) \times 3S_6)$ contains an elementary abelian normal subgroup of order eight whose involutions lie in the class 2B, see [CCN$^+$85, p. 234]. As a module for the group $L_3(2)$, this normal subgroup is irreducible, and the restriction of the action to the two classes of S_4 type subgroups fixes a one- and a two-dimensional subspace, respectively. Hence we have one class of subgroups of order n that centralize a 2B element and one class of subgroups of order n that normalize a 2B-pure Klein four group. Clearly the subgroups in the first class coincide with the subgroups of order n in $2_+^{1+24}.Co_1$ type subgroups. By the above classification of 2B-pure Klein four groups in M, the subgroups in the second class coincide with the subgroups of order n in $2^{2+11+22}.(M_{24} \times S_3)$ type subgroups.

It remains to show that the subgroups of order n do *not* stabilize both a 2B element *and* a 2B–pure Klein four group. We do this by direct computations with a $2^{2+11+22}.(M_{24} \times S_3)$ type group, which is available via the AtlasRep package, see [WPN$^+$22].

First we fetch the group, and factor out the largest solvable normal subgroup, by suitable actions on blocks.

```
──────── Example ────────
gap> g:= AtlasGroup( "2^(2+11+22).(M24xS3)" );
<permutation group of size 50472333605150392320 with 2 generators>
gap> NrMovedPoints( g );
294912
gap> bl:= Blocks( g, MovedPoints( g ) );;
gap> Length( bl );
147456
gap> hom1:= ActionHomomorphism( g, bl, OnSets );;
gap> act1:= Image( hom1 );;
gap> Size( g ) / Size( act1 );
8192
gap> bl2:= Blocks( act1, MovedPoints( act1 ) );;
gap> Length( bl2 );
72
gap> hom2:= ActionHomomorphism( act1, bl2, OnSets );;
gap> act2:= Image( hom2 );;
gap> Size( act2 );
1468938240
gap> Size( MathieuGroup( 24 ) ) * 6;
1468938240
gap> bl3:= AllBlocks( act2 );;
gap> List( bl3, Length );
[ 24, 3 ]
gap> bl3:= Orbit( act2, bl3[2], OnSets );;
gap> hom3:= ActionHomomorphism( act2, bl3, OnSets );;
gap> act3:= Image( hom3 );;
```

Now we compute an isomorphism from the factor group of type M_{24} to the group that belongs to GAP's table of marks. Then we use the information from the table of marks to compute a solvable subgroup of maximal order in M_{24} (which is 13 824), and take the preimage under the isomorphism. Finally, we take the preimage of this group in the original group.

```
──────── Example ────────
gap> tom:= TableOfMarks( "M24" );;
gap> tomgroup:= UnderlyingGroup( tom );;
gap> iso:= IsomorphismGroups( act3, tomgroup );;
gap> pos:= Positions( OrdersTom( tom ), 13824 );
```

```
[ 1508 ]
gap> sub:= RepresentativeTom( tom, pos[1] );;
gap> pre:= PreImages( iso, sub );;
gap> pre:= PreImages( hom3, pre );;
gap> pre:= PreImages( hom2, pre );;
gap> pre:= PreImages( hom1, pre );;
gap> Size( pre ) = n;
true
```

The subgroups stabilizes a Klein four group. It does not stabilize a 2B element because its centre is trivial.

```
——————————————————— Example ———————————————————
gap> pciso:= IsomorphismPcGroup( pre );;
gap> Size( Centre( Image( pciso ) ) );
1
```

6.5 Proof of the Corollary

With the computations in the previous sections, we have collected the information that is needed to show the corollary stated in Section 6.1.

```
——————————————————— Example ———————————————————
gap> Filtered( Set( RecNames( MaxSolv ) ),
>             x -> MaxSolv.( x )^2 >= Size( CharacterTable( x ) ) );
[ "Fi23", "J2", "J2.2", "M11", "M12", "M22.2" ]
```

Chapter 7

Large Nilpotent Subgroups of Sporadic Simple Groups

Date: June 6th, 2009

We show that any nontrivial nilpotent subgroup U in a sporadic simple group G satisfies $|U| \cdot |\mathbf{N}_G(U)| < |G|$. The proof uses the information in the Atlas of Finite Groups [CCN$^+$85] and the GAP system [GAP24], in particular its Character Table Library [Bre25] and its library of Tables of Marks [MNP19]. (In [Vdo00], it is shown that in any finite nonabelian simple group G, any nilpotent subgroup U satisfies $|U|^2 < |G|$.)

7.1 The Result

The aim of this writeup is to show the following statement.

Proposition: Let G be a sporadic simple group, let U be a nontrivial nilpotent subgroup in G, and let $\mathbf{N}_G(U)$ denote the normalizer of U in G. Then $|U| \cdot |\mathbf{N}_G(U)| < |G|$ holds.

The following criteria are sufficient to prove this proposition. Note that we are interested in an argument that uses only information about the character tables of the sporadic simple groups and of their maximal subgroups.

Lemma 1: Let G be a nonabelian finite simple group, and suppose that U is a nontrivial nilpotent subgroup of G such that $|U| \cdot |\mathbf{N}_G(U)| \geq |G|$ holds. Let $\Pi = \{p_1, p_2, \ldots, p_n\}$ be the set of prime divisors of $|U|$, and set $n = \prod_{p \in \Pi} p$.

(a) G contains an element g of order n and a maximal subgroup M with the properties $g \in Z(U)$ and $\mathbf{N}_G(U) \leq M$. Set $c := \gcd(|\mathbf{C}_G(g)|_\Pi, |M|)$, where $|\mathbf{C}_G(g)|_\Pi$ denotes the largest divisor of the order of the centralizer of g in G whose prime divisors are elements of the set Π. Then we have $|U| \leq c$ and hence $c \cdot |M| \geq |G|$, in particular $|M|^2 \geq |G|$.

(b)   If (g, M) is as in part (a) then one of the following holds.

    (b1)

        U is normal in M, and the Fitting subgroup $Fit(M)$ of M satisfies $|Fit(M)| \cdot |M| \geq |G|$.

    (b2)

        U is not normal in M, so $\mathbf{N}_G(U)$ is a proper subgroup of M, in particular $|G| \leq |U| \cdot |M|/2 \leq c \cdot |M|/2$ holds.

(c) Let (g,M) be as in part (b2) and assume that M contains a normal subgroup K such that $\pi(M) := M/K$ is an almost simple group with socle S, i. e., $\pi(M)$ has a nonabelian simple normal subgroup S such that $\mathbf{C}_{\pi(M)}(S)$ is trivial. Then either $U \leq K$ holds, and hence $|K| \cdot |M| \geq |G|$, or we are in the following situation.

The group $\pi(U) := UK/K$ is a nontrivial nilpotent normal subgroup of $\pi(N) := \mathbf{N}_G(U)K/K$, and $H := S \cap \pi(N)$ is a proper subgroup of S. The latter statement holds because otherwise $S \cap \pi(U)$ would be normal in S and thus would be trivial, which would imply that S would centralize $\pi(U)$.

As a consequence, $|\pi(N)|$ divides $|\pi(M)/S| \cdot |H| = |\pi(M)|/[S:H]$, in particular, $[S:H] \leq |\pi(M)|/|\pi(N)| = |M|/|\mathbf{N}_G(U)K| \leq |M|/|\mathbf{N}_G(U)| \leq |M| \cdot |U|/|G| \leq c/[G:M]$ holds.

We will apply Lemma 1 as follows.

From the character tables of G and M, the value $|Fit(M)|$ and the maximal possible c can be computed. If part (a) of the lemma applies then we verify that part (b1) does *not* apply, and that either (b2) or (c) yields a contradiction. Note that we can determine from the character table of M whether M has a normal subgroup K such that M/K is almost simple, and in this case we can compute the order of the socle S of M/K.

For proving the nonexistence of the subgroup H in the situation of part (c), we will show that all subgroups of $\pi(M)$ of index up to $d := c \cdot [\pi(M):S]/[G:M]$ contain S. For that, we will compute the complete list of those possible permutation characters of $\pi(M)$ whose degree is at most d, and then check that the kernels of these characters contain S.

(Note that these computations are cheap because the bound d is small in the cases that occur. There are easier criteria for proving the nonexistence of a subgroup of index at most d in a simple group S, for example in the case $|S| > d!/2$ or if the smallest nontrivial irreducible degree of S is at least d; but these criteria do not suffice in our situation.)

We illustrate the application of Lemma 1 with some examples.

J_1: The first Janko group J_1 (see [CCN$^+$85, p. 36]) has order $175\,560$, and the largest maximal subgroup has order 660. The largest centralizer of a nonidentity element in J_1 has order 120, and $660 \cdot 120 = 79\,200 < |J_1|$. Thus J_1 satisfies the proposition.

$\mathbb{M}$: For the Monster group $\mathbb{M}$ (see [CCN$^+$85, p. 234]), we read off from the list [Wil] of maximal subgroups that the only maximal subgroups M of $\mathbb{M}$ with the property $|M|^2 \geq \mathbb{M}$ have the structure $2.B$. Already for the second largest maximal subgroups, with the structure $2^{1+24}.Co_1$, the order is smaller than the index in the Monster.

Only elements g from the classes 2A, 2B, and 3A have the property that the product of $|2.B|$ and the order of the centralizer of g in M is not smaller than $|M|$. So U can be only a 2- or a 3-subgroup of $2.B$. However, the 2-part and the 3-part of $|2.B|$ are 2^{42} and 3^{13}, respectively, which are smaller than the index of $2.B$ in M. Thus M satisfies the proposition.

Fi_{23}:

We show that no counterexample to the proposition can arise from maximal subgroups M of the type $O_8^+(3):S_3$ in the Fischer group Fi_{23} (see [CCN$^+$85, p. 177]). Several element centralizers in G satisfy Lemma 1 (a), the largest value c arises from elements in the class 6B, whose centralizers have order $2^8 \cdot 3^9$, which divides $|M|$. So $|U| \leq 2^8 \cdot 3^9$, and a possible counterexample to the proposition must satisfy $|\mathbf{N}_G(U)| \geq |G|/(2^8 \cdot 3^9) = 811\,588\,377\,600$. We have $|M| = 29\,713\,078\,886\,400$, which is less than 37 times this minimal order required for $\mathbf{N}_G(U)$.

However, the intersection H of this group with the simple subgroup $S \cong O_8^+(3)$ in M cannot be at most 36, because the largest maximal subgroups in S have index 1 080 (see [CCN+85, p. 140]). Arguing not with S but with M, we can show –using only the character table of M– that all proper subgroups of index less than $37 \cdot 6$ in M contain S.

7.2 The Proof

The following GAP function utilizes Lemma 1. Its input are the GAP character table `tbl` of a group G, say, and a list `maxesinfo` of character tables of maximal subgroups of G, covering at least all those maximal subgroups M for which $|M|^2 \geq |G|$ holds.

The idea is to collect pairs (M, g) that satisfy part (a) of Lemma 1, and then to show that they do not satisfy part (b) or part (c). For each maximal subgroup M that admits elements g as in Lemma 1, information is printed how this candidate is excluded.

The function returns a list of length three. The first entry is `true` if the criteria of Lemma 1 are sufficient to prove that the proposition is true for G, and `false` otherwise. The second entry is the name of G, and the third entry in the number of maximal subgroups M for which an element g as in Lemma 1 (a) exists.

```
─────────────── Example ───────────────
gap> ApplyTheLemma:= function( tbl, maxesinfo )
>     local Gname, Gsize, cents, orders, result, Mtbl, Msize, maxc, i,
>           pi, pipart, c, Mclasslengths, Fit, excluded, Kclasses, Mbar,
>           Ksize, Sclasses, Ssize, d;
>     Gname:= Identifier( tbl );
>     Gsize:= Size( tbl );
>     cents:= SizesCentralizers( tbl );
>     orders:= OrdersClassRepresentatives( tbl );
>     result:= [ true, Gname, 0 ];
>     # Run over the relevant maximal subgroups.
>     for Mtbl in maxesinfo do
>       Msize:= Size( Mtbl );
>       # Run over nonidentity class representatives g of squarefree
>       # order, compute the largest c that occurs.
>       maxc:= 1;
>       for i in [ 2 .. NrConjugacyClasses( tbl ) ] do
>         pi:= Factors( orders[i] );
>         if IsSet( pi ) then
>           # The elements in class 'i' have squarefree order.
>           pipart:= Product( Filtered( Factors( cents[i] ),
>                                       x -> x in pi ) );
>           c:= Gcd( pipart, Msize );
>           if maxc < c then
>             maxc:= c;
>           fi;
>         fi;
>       od;
>       if maxc * Msize >= Gsize then
>         # Criterion (a) is satisfied, try to exclude (b) and (c).
>         result[3]:= result[3] + 1;
>         Print( Gname, ": consider M = ", Identifier( Mtbl ),
>                ", c = ", StringPP( maxc ),
```

```
>                    ", c * |M| / |G| >= ", Int( maxc * Msize / Gsize ),
>                "\n" );
>          Mclasslengths:= SizesConjugacyClasses( Mtbl );
>          Fit:= Mclasslengths{ ClassPositionsOfFittingSubgroup( Mtbl ) };
>          if Sum( Fit ) * Msize >= Gsize then
>            # Criterion (b1) is satisfied.
>            Print( Gname, ": not excludable by (b1)\n" );
>            result[1]:= false;
>          elif maxc * Msize < 2 * Gsize then
>            # Criterion (b2) is not satisfied.
>            Print( Gname, ":      excluded by (b2)\n" );
>          else
>            # Run over the normal subgroups of M.
>            excluded:= false;
>            for Kclasses in ClassPositionsOfNormalSubgroups( Mtbl ) do
>              Mbar:= Mtbl / Kclasses;
>              Ksize:= Sum( Mclasslengths{ Kclasses } );
>              if IsAlmostSimpleCharacterTable( Mbar ) and
>                 Ksize * Msize < Gsize then
>                # We are in the situation of criterion (c).
>                # The socle is the unique minimal normal subgroup.
>                Sclasses:= ClassPositionsOfMinimalNormalSubgroups(
>                               Mbar )[1];
>                Ssize:= Sum( SizesConjugacyClasses( Mbar ){ Sclasses } );
>                d:= Int( maxc * Msize * Size( Mbar )
>                    / ( Gsize * Ssize ) );
>                # Try to show that all subgroups of index up to d
>                # in Mbar contain the socle.
>                if ForAll( [ 2 .. d ],
>                    n -> ForAll( PermChars( Mbar, rec( torso:= [ n ] ) ),
>                          chi -> IsSubset(
>                                    ClassPositionsOfKernel( chi ),
>                                    Sclasses ) ) ) then
>                  Print( Gname, ":      excluded by (c), |K| = ",
>                         StringPP( Ksize ), ", degree bound ", d, "\n" );
>                  excluded:= true;
>                  break;
>                fi;
>              fi;
>            od;
>            if not excluded then
>              Print( Gname, ": not excludable by (c)\n" );
>              result[1]:= false;
>            fi;
>          fi;
>        fi;
>      od;
>      return result;
> end;;
```

So our proof relies on the classifications of maximal subgroups of sporadic simple groups, see [CCN+85] and [BN95].

The GAP Character Table Library [Bre25] contains the character tables of the sporadic simple groups and of their maximal subgroups, except that not all character tables of maximal subgroups of the Monster group are available yet. (See Section 7.1 for the treatment of the Monster group.)

Since the GAP Character Table Library is used for the computations in this section, we first load this package.

```
───────────── Example ─────────────
gap> LoadPackage( "ctbllib", false );
true
```

Now we apply the function to the sporadic simple groups.

```
───────────── Example ─────────────
gap> info:= [];;
gap> for name in AllCharacterTableNames( IsSporadicSimple, true,
>                                        IsDuplicateTable, false ) do
>      tbl:= CharacterTable( name );
>      if HasMaxes( tbl ) then
>        mx:= List( Maxes( tbl ), CharacterTable );
>      elif name = "M" then
>        mx:= [ CharacterTable( "2.B" ) ];
>      else
>        Error( "this should not happen ...");
>      fi;
>      Add( info, ApplyTheLemma( tbl, mx ) );
>    od;
B: consider M = 2.2E6(2).2, c = 2^38, c * |M| / |G| >= 20
B:     excluded by (c), |K| = 2, degree bound 40
Co1: consider M = Co2, c = 2^13*3^5, c * |M| / |G| >= 20
Co1:     excluded by (c), |K| = 1, degree bound 20
Co1: consider M = 3.Suz.2, c = 2^13*3^5, c * |M| / |G| >= 1
Co1:     excluded by (b2)
Co2: consider M = U6(2).2, c = 2^16, c * |M| / |G| >= 28
Co2:     excluded by (c), |K| = 1, degree bound 56
Co2: consider M = 2^10:m22:2, c = 2^18, c * |M| / |G| >= 5
Co2:     excluded by (c), |K| = 2^10, degree bound 11
Co2: consider M = 2^1+8:s6f2, c = 2^18, c * |M| / |G| >= 4
Co2:     excluded by (c), |K| = 2^9, degree bound 4
Co3: consider M = McL.2, c = 2^4*3^4, c * |M| / |G| >= 4
Co3:     excluded by (c), |K| = 1, degree bound 9
F3+: consider M = Fi23, c = 2^9*3^9, c * |M| / |G| >= 32
F3+:     excluded by (c), |K| = 1, degree bound 32
Fi22: consider M = 2.U6(2), c = 2^7*3^6, c * |M| / |G| >= 26
Fi22:     excluded by (c), |K| = 2, degree bound 26
Fi22: consider M = O7(3), c = 2^7*3^6, c * |M| / |G| >= 6
Fi22:     excluded by (c), |K| = 1, degree bound 6
Fi22: consider M = Fi22M3, c = 2^7*3^6, c * |M| / |G| >= 6
Fi22:     excluded by (c), |K| = 1, degree bound 6
Fi22: consider M = O8+(2).3.2, c = 2^7*3^6, c * |M| / |G| >= 1
Fi22:     excluded by (b2)
Fi23: consider M = 2.Fi22, c = 2^8*3^9, c * |M| / |G| >= 159
Fi23:     excluded by (c), |K| = 2, degree bound 159
Fi23: consider M = O8+(3).3.2, c = 2^8*3^9, c * |M| / |G| >= 36
```

```
Fi23:     excluded by (c), |K| = 1, degree bound 219
HS: consider M = M22, c = 2^7, c * |M| / |G| >= 1
HS:     excluded by (b2)
M11: consider M = A6.2_3, c = 2^4, c * |M| / |G| >= 1
M11:      excluded by (b2)
M12: consider M = M11, c = 2^4, c * |M| / |G| >= 1
M12:      excluded by (b2)
M12: consider M = M12M2, c = 2^4, c * |M| / |G| >= 1
M12:      excluded by (b2)
M22: consider M = L3(4), c = 2^6, c * |M| / |G| >= 2
M22:      excluded by (c), |K| = 1, degree bound 2
M22: consider M = 2^4:a6, c = 2^7, c * |M| / |G| >= 1
M22:      excluded by (b2)
M23: consider M = M22, c = 2^7, c * |M| / |G| >= 5
M23:      excluded by (c), |K| = 1, degree bound 5
M24: consider M = M23, c = 2^7, c * |M| / |G| >= 5
M24:      excluded by (c), |K| = 1, degree bound 5
M24: consider M = 2^4:a8, c = 2^10, c * |M| / |G| >= 1
M24:      excluded by (b2)
McL: consider M = U4(3), c = 3^6, c * |M| / |G| >= 2
McL:      excluded by (c), |K| = 1, degree bound 2
Ru: consider M = 2F4(2)'.2, c = 2^12, c * |M| / |G| >= 1
Ru:       excluded by (b2)
Suz: consider M = G2(4), c = 2^12, c * |M| / |G| >= 2
Suz:      excluded by (c), |K| = 1, degree bound 2
```

First of all, we see that Lemma 1 is sufficient to prove the proposition, since all candidates were excluded.

Moreover, we see that for the following ten sporadic simple groups, no candidates had to be considered. (No information was printed about these groups.)

```
––––––––––––––––––– Example –––––––––––––––––––
gap> Filtered( info, x -> x[3] = 0 );
[ [ true, "HN", 0 ], [ true, "He", 0 ], [ true, "J1", 0 ],
  [ true, "J2", 0 ], [ true, "J3", 0 ], [ true, "J4", 0 ],
  [ true, "Ly", 0 ], [ true, "M", 0 ], [ true, "ON", 0 ],
  [ true, "Th", 0 ] ]
```

7.3   Alternative: Use **GAP**'s Tables of Marks

We can easily inspect all conjugacy classes of subgroups of a group G whose table of marks is contained in **GAP**'s Library of Tables of Marks [MNP19]. First we load this **GAP** package.

```
––––––––––––––––––– Example –––––––––––––––––––
gap> LoadPackage( "tomlib", false );
true
```

The following **GAP** function takes the table of marks of a group G and returns the list of pairs $[U, \mathbf{N}_G(U)]$ where U ranges over representatives of conjugacy classes of those nilpotent subgroups of G for which $|U| \cdot |\mathbf{N}_G(U)|$ is maximal.

```
───────────────────── Example ─────────────────────
gap> maximalpairs:= function( tom )
>     local g, max, result, i, u, n, prod;
>     g:= UnderlyingGroup( tom );
>     max:= 1;
>     result:= [];
>     for i in [ 1 .. Length( OrdersTom( tom ) ) ] do
>       u:= RepresentativeTom( tom, i );
>       if not IsTrivial( u ) and IsNilpotent( u ) then
>         n:= Normalizer( g, u );
>         prod:= Size( u ) * Size( n );
>         if max < prod then
>           max:= prod;
>           result:= [ [ u, n ] ];
>         elif max = prod then
>           Add( result, [ u, n ] );
>         fi;
>       fi;
>     od;
>     return result;
> end;;
```

So let us collect the data for those sporadic simple groups for which the table of marks is known.

```
───────────────────── Example ─────────────────────
gap> info:= [];;
gap> for name in AllCharacterTableNames( IsSporadicSimple, true,
>                                        IsDuplicateTable, false ) do
>       tom:= TableOfMarks( name );
>       if tom <> fail then
>         Add( info, [ name, tom, maximalpairs( tom ) ] );
>       fi;
>     od;
gap> Length( info );
12
```

We got results for twelve sporadic simple groups. The following computations show that in ten cases, the simple group G contains a unique class of nontrivial nilpotent subgroups U for which the maximal value of $|U| \cdot |\mathbf{N}_G(U)|$ is attained. The ratio of this value and $|G|$ is less than 21 per cent. The following table shows the name of the group G, the orders of U and $\mathbf{N}_G(U)$, and the integral part of 10^6 times the ratio.

```
───────────────────── Example ─────────────────────
gap> List( info, x -> Length( x[3] ) );
[ 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1 ]
gap> mat:= [];;
gap> for entry in info do
>       pair:= entry[3][1];                        # [ U, N_G(U) ]
>       bound:= Size( pair[1] ) * Size( pair[2] );  # |U|*|N_G(U)|
>       size:= Size( UnderlyingGroup( entry[2] ) ); # |G|
>       Add( mat, [ entry[1],
>                   StringPP( Size( pair[1] ) ),
>                   StringPP( Size( pair[2] ) ),
```

```
>                  Int( 10^6 * bound / size ) ] );
>        if Size( pair[1] ) * Size( pair[2] ) >= 21/100 * size then
>          Error("!");
>        fi;
>    od;
gap> PrintArray( mat );
[ [          Co3,      3^5, 2^5*3^7*5*11,          1886 ],
  [           HS,      2^6,      2^9*3*7,         15515 ],
  [           He,      2^6,   2^10*3^3*5,          2195 ],
  [           J1,       19,       2*3*19,         12337 ],
  [           J2,      2^6,      2^7*3^2,        121904 ],
  [           J3,      3^5,      2^3*3^5,          9404 ],
  [          M11,      3^2,      2^4*3^2,        163636 ],
  [          M12,      2^5,        2^6*3,         64646 ],
  [          M22,      2^4,    2^7*3^2*5,        207792 ],
  [          M23,      2^4,  2^7*3^2*5*7,         63241 ],
  [          M24,      2^6,   2^10*3^3*5,         36137 ],
  [          McL,      3^5,    2^4*3^6*5,         15779 ] ]
```

Moreover, we see that in most cases, the group U for which the maximum is attained is not the largest p-subgroup in the simple group in question.

Chapter 8

Permutation Characters in GAP

Date: April 17th, 1999

This is a loose collection of examples of computations with permutation characters and possible permutation characters in the GAP system [GAP24]. We mainly use the GAP implementation of the algorithms to compute possible permutation characters that are described in [BP98], and information from the Atlas of Finite Groups [CCN+85]. A *possible permutation character* of a finite group G is a character satisfying the conditions listed in Section "Possible Permutation Characters" of the GAP Reference Manual.

- Sections 8.14 and 8.15 were added in October 2001.

- Section 8.16.1 was added in June 2009.

- Section 8.16.2 was added in September 2009.

- Section 8.16.3 was added in October 2009.

- Section 8.16.4 was added in November 2009.

- Section 8.17 was added in June 2012.

- Section 8.18 was added in October 2017.

- Section 8.19 was added in December 2021.

In the following, the GAP Character Table Library [Bre25] will be used frequently.

```
———————————————————— Example ————————————————————
 gap> LoadPackage( "ctbllib", "1.2", false );
 true
```

8.1 Some Computations with M_{24}

We start with the sporadic simple Mathieu group $G = M_{24}$ in its natural action on 24 points.

```
———————————————————— Example ————————————————————
 gap> g:= MathieuGroup( 24 );;
 gap> SetName( g, "m24" );
 gap> Size( g );  IsSimple( g );  NrMovedPoints( g );
```

```
244823040
true
24
```

The conjugacy classes that are computed for a group can be ordered differently in different GAP sessions. In order to make the output shown in the following examples stable, we first sort the conjugacy classes of *G* for our purposes.

```
————————————— Example —————————————
gap> ccl:= AttributeValueNotSet( ConjugacyClasses, g );;
gap> HasConjugacyClasses( g );
false
gap> invariants:= List( ccl, c -> [ Order( Representative( c ) ),
>        Size( c ), Size( ConjugacyClass( g, Representative( c )^2 ) ) ] );;
gap> SortParallel( invariants, ccl );
gap> SetConjugacyClasses( g, ccl );
```

The permutation character pi of *G* corresponding to the action on the moved points is constructed. This action is 5-transitive.

```
————————————— Example —————————————
gap> NrConjugacyClasses( g );
26
gap> pi:= NaturalCharacter( g );
Character( CharacterTable( m24 ),
 [ 24, 8, 0, 6, 0, 0, 4, 0, 4, 2, 0, 3, 3, 2, 0, 2, 0, 0, 1, 1, 1, 1,
   0, 0, 1, 1 ] )
gap> IsTransitive( pi );  Transitivity( pi );
true
5
gap> SetIdentifier( CharacterTable( g ), "M24table" );
gap> Display( pi );
M24table
```

```
    2 10 10  9  3  3  7  7  5  2  3  3  1  1  4  2  .  2  2  1
    3  3  1  1  3  2  1  .  1  1  1  1  1  1  1  .  .  .  1  1  .
    5  1  .  1  1  .  .  .  .  1  .  .  .  .  .  1  .  .  .  .
    7  1  1  .  .  1  .  .  .  .  .  .  1  1  .  .  .  .  .  1
   11  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  1  .  .  .
   23  1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

       1a 2a 2b 3a 3b 4a 4b 4c 5a 6a 6b 7a 7b 8a 10a 11a 12a 12b 14a

Y.1    24  8  .  6  .  .  4  .  4  2  .  3  3  2  .  2  .  .  1

    2  1  .  .  .  .  .  .
    3  .  1  1  1  1  .  .
    5  .  1  1  .  .  .  .
    7  1  .  .  1  1  .  .
   11  .  .  .  .  .  .  .
   23  .  .  .  .  .  1  1

       14b 15a 15b 21a 21b 23a 23b
```

```
Y.1      1   1   1   .   .   1   1
```

(We have set the `Identifier` (**CTblLib: IdentifierOfMainTable**) value of the character table because otherwise some default identifier would be chosen, which depends on the GAP session.)

`pi` determines the permutation characters of the *G*-actions on related sets, for example `piop` on the set of ordered and `piup` on the set of unordered pairs of points.

```
———————————————————— Example ————————————————————
gap> piop:= pi * pi;
Character( CharacterTable( m24 ),
 [ 576, 64, 0, 36, 0, 0, 16, 0, 16, 4, 0, 9, 9, 4, 0, 4, 0, 0, 1, 1,
   1, 1, 0, 0, 1, 1 ] )
gap> IsTransitive( piop );
false
gap> piup:= SymmetricParts( UnderlyingCharacterTable(pi), [ pi ], 2 )[1];
Character( CharacterTable( m24 ),
 [ 300, 44, 12, 21, 0, 4, 12, 0, 10, 5, 0, 6, 6, 4, 2, 3, 0, 1, 2, 2,
   1, 1, 0, 0, 1, 1 ] )
gap> IsTransitive( piup );
false
```

Clearly the action on unordered pairs is not transitive, since the pairs $[i, i]$ form an orbit of their own. There are exactly two *G*-orbits on the unordered pairs, hence the *G*-action on 2-sets of points is transitive.

```
———————————————————— Example ————————————————————
gap> ScalarProduct( piup, TrivialCharacter( g ) );
2
gap> comb:= Combinations( [ 1 .. 24 ], 2 );;
gap> hom:= ActionHomomorphism( g, comb, OnSets );;
gap> pihom:= NaturalCharacter( hom );
Character( CharacterTable( m24 ),
 [ 276, 36, 12, 15, 0, 4, 8, 0, 6, 3, 0, 3, 3, 2, 2, 1, 0, 1, 1, 1,
   0, 0, 0, 0, 0, 0 ] )
gap> Transitivity( pihom );
1
```

In terms of characters, the permutation character `pihom` is the difference of `piup` and `pi`. Note that GAP does not know that this difference is in fact a character; in general this question is not easy to decide without knowing the irreducible characters of *G*, and up to now GAP has not computed the irreducibles.

```
———————————————————— Example ————————————————————
gap> pi2s:= piup - pi;
VirtualCharacter( CharacterTable( m24 ),
 [ 276, 36, 12, 15, 0, 4, 8, 0, 6, 3, 0, 3, 3, 2, 2, 1, 0, 1, 1, 1,
   0, 0, 0, 0, 0, 0 ] )
gap> pi2s = pihom;
true
gap> HasIrr( g );  HasIrr( CharacterTable( g ) );
false
false
```

The point stabilizer in the action on 2-sets is in fact a maximal subgroup of G, which is isomorphic to the automorphism group $M_{22} : 2$ of the Mathieu group M_{22}. Thus this permutation action is primitive. But we cannot apply IsPrimitive (**Reference: IsPrimitive**) to the character pihom for getting this answer because primitivity of characters is defined in a different way, cf. IsPrimitiveCharacter (**Reference: IsPrimitiveCharacter**).

```
                        ───────── Example ─────────
  gap> IsPrimitive( g, comb, OnSets );
  true
```

We could also have computed the transitive permutation character of degree 276 using the GAP Character Table Library instead of the group G, since the character tables of G and all its maximal subgroups are available, together with the class fusions of the maximal subgroups into G.

```
                        ───────── Example ─────────
  gap> tbl:= CharacterTable( "M24" );
  CharacterTable( "M24" )
  gap> maxes:= Maxes( tbl );
  [ "M23", "M22.2", "2^4:a8", "M12.2", "2^6:3.s6", "L3(4).3.2_2",
    "2^6:(psl(3,2)xs3)", "L2(23)", "L3(2)" ]
  gap> s:= CharacterTable( maxes[2] );
  CharacterTable( "M22.2" )
  gap> TrivialCharacter( s )^tbl;
  Character( CharacterTable( "M24" ),
   [ 276, 36, 12, 15, 0, 4, 8, 0, 6, 3, 0, 3, 3, 2, 2, 1, 1, 0, 1, 1,
     0, 0, 0, 0, 0, 0 ] )
```

Note that the sequence of conjugacy classes in the library table of G does in general not agree with the succession computed for the group.

8.2   All Possible Permutation Characters of M_{11}

We compute all possible permutation characters of the Mathieu group M_{11}, using the three different strategies available in GAP. First we try the algorithm that enumerates all candidates via solving a system of inequalities, which is described in [BP98, Section 3.2].

```
                        ───────── Example ─────────
  gap> m11:= CharacterTable( "M11" );;
  gap> SetName( m11, "m11" );
  gap> perms:= PermChars( m11 );
  [ Character( m11, [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ] ), Character( m11,
    [ 11, 3, 2, 3, 1, 0, 1, 1, 0, 0 ] ), Character( m11,
    [ 12, 4, 3, 0, 2, 1, 0, 0, 1, 1 ] ), Character( m11,
    [ 22, 6, 4, 2, 2, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 55, 7, 1, 3, 0, 1, 1, 1, 0, 0 ] ), Character( m11,
    [ 66, 10, 3, 2, 1, 1, 0, 0, 0, 0 ] ), Character( m11,
    [ 110, 6, 2, 2, 0, 0, 2, 2, 0, 0 ] ), Character( m11,
    [ 110, 6, 2, 6, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 110, 14, 2, 2, 0, 2, 0, 0, 0, 0 ] ), Character( m11,
    [ 132, 12, 6, 0, 2, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 144, 0, 0, 0, 4, 0, 0, 0, 1, 1 ] ), Character( m11,
    [ 165, 13, 3, 1, 0, 1, 1, 1, 0, 0 ] ), Character( m11,
```

```
    [ 220, 4, 4, 0, 0, 4, 0, 0, 0, 0 ] ), Character( m11,
    [ 220, 12, 4, 4, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 220, 20, 4, 0, 0, 2, 0, 0, 0, 0 ] ), Character( m11,
    [ 330, 2, 6, 2, 0, 2, 0, 0, 0, 0 ] ), Character( m11,
    [ 330, 18, 6, 2, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 396, 12, 0, 4, 1, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 440, 8, 8, 0, 0, 2, 0, 0, 0, 0 ] ), Character( m11,
    [ 440, 24, 8, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 495, 15, 0, 3, 0, 0, 1, 1, 0, 0 ] ), Character( m11,
    [ 660, 4, 3, 4, 0, 1, 0, 0, 0, 0 ] ), Character( m11,
    [ 660, 12, 3, 0, 0, 3, 0, 0, 0, 0 ] ), Character( m11,
    [ 660, 12, 12, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 660, 28, 3, 0, 0, 1, 0, 0, 0, 0 ] ), Character( m11,
    [ 720, 0, 0, 0, 0, 0, 0, 0, 5, 5 ] ), Character( m11,
    [ 792, 24, 0, 0, 2, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 880, 0, 16, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 990, 6, 0, 2, 0, 0, 2, 2, 0, 0 ] ), Character( m11,
    [ 990, 6, 0, 6, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 990, 30, 0, 2, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 1320, 8, 6, 0, 0, 2, 0, 0, 0, 0 ] ), Character( m11,
    [ 1320, 24, 6, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 1584, 0, 0, 0, 4, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 1980, 12, 0, 4, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 1980, 36, 0, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 2640, 0, 12, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 3960, 24, 0, 0, 0, 0, 0, 0, 0, 0 ] ), Character( m11,
    [ 7920, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
gap> Length( perms );
39
```

Next we try the improved combinatorial approach that is sketched at the end of Section 3.2 in [BP98]. We get the same characters, except that they may be ordered in a different way; thus we compare the ordered lists.

```
―――――――――――――――――――――――― Example ――――――――――――――――――――――――
gap> degrees:= DivisorsInt( Size( m11 ) );;
gap> perms2:= [];;
gap> for d in degrees do
>        Append( perms2, PermChars( m11, d ) );
>    od;
gap> Set( perms ) = Set( perms2 );
true
```

Finally, we try the algorithm that is based on Gaussian elimination and that is described in [BP98, Section 3.3].

```
―――――――――――――――――――――――― Example ――――――――――――――――――――――――
gap> perms3:= [];;
gap> for d in degrees do
>        Append( perms3, PermChars( m11, rec( torso:= [ d ] ) ) );
>    od;
gap> Set( perms ) = Set( perms3 );
true
```

GAP provides two more functions to test properties of permutation characters. The first one yields no new information in our case, but the second excludes one possible permutation character; note that `TestPerm5` needs a *p*-modular Brauer table, and the GAP character table library contains all Brauer tables of M_{11}.

```
—————————————————————— Example ——————————————————————
  gap> newperms:= TestPerm4( m11, perms );;
  gap> newperms = perms;
  true
  gap> newperms:= TestPerm5( m11, perms, m11 mod 11 );;
  gap> newperms = perms;
  false
  gap> Difference( perms, newperms );
  [ Character( m11, [ 220, 4, 4, 0, 0, 4, 0, 0, 0, 0 ] ) ]
```

GAP knows the table of marks of M_{11}, from which the permutation characters can be extracted. It turns out that M_{11} has 39 conjugacy classes of subgroups but only 36 different permutation characters, so three candidates computed above are in fact not permutation characters.

```
—————————————————————— Example ——————————————————————
  gap> tom:= TableOfMarks( "M11" );
  TableOfMarks( "M11" )
  gap> trueperms:= PermCharsTom( m11, tom );;
  gap> Length( trueperms );  Length( Set( trueperms ) );
  39
  36
  gap> Difference( perms, trueperms );
  [ Character( m11, [ 220, 4, 4, 0, 0, 4, 0, 0, 0, 0 ] ),
    Character( m11, [ 660, 4, 3, 4, 0, 1, 0, 0, 0, 0 ] ),
    Character( m11, [ 660, 12, 3, 0, 0, 3, 0, 0, 0, 0 ] ) ]
```

8.3   The Action of $U_6(2)$ on the Cosets of M_{22}

We are interested in the permutation character of $U_6(2)$ (see [CCN+85, p. 115]) that corresponds to the action on the cosets of a M_{22} subgroup (see [CCN+85, p. 39]). The character tables of both the group and the point stabilizer are available in the GAP character table library, so we can compute class fusion and permutation character directly; note that if the class fusion is not stored on the table of the subgroup, in general one will not get a unique fusion but only a list of candidates for the fusion.

```
—————————————————————— Example ——————————————————————
  gap> u62:= CharacterTable( "U6(2)" );;
  gap> m22:= CharacterTable( "M22" );;
  gap> fus:= PossibleClassFusions( m22, u62 );
  [ [ 1, 3, 7, 10, 14, 15, 22, 24, 24, 26, 33, 34 ],
    [ 1, 3, 7, 10, 14, 15, 22, 24, 24, 26, 34, 33 ],
    [ 1, 3, 7, 11, 14, 15, 22, 24, 24, 27, 33, 34 ],
    [ 1, 3, 7, 11, 14, 15, 22, 24, 24, 27, 34, 33 ],
    [ 1, 3, 7, 12, 14, 15, 22, 24, 24, 28, 33, 34 ],
    [ 1, 3, 7, 12, 14, 15, 22, 24, 24, 28, 34, 33 ] ]
  gap> RepresentativesFusions( m22, fus, u62 );
  [ [ 1, 3, 7, 10, 14, 15, 22, 24, 24, 26, 33, 34 ] ]
```

We see that there are six possible class fusions that are equivalent under table automorphisms of $U_6(2)$ and $M22$.

```
─────────────────────────── Example ───────────────────────────
gap> cand:= Set( fus,
>  x -> Induced( m22, u62, [ TrivialCharacter( m22 ) ], x )[1] );
[ Character( CharacterTable( "U6(2)" ),
  [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, 0, 48, 0, 16, 6, 0, 0, 0, 0,
      0, 0, 6, 0, 2, 0, 0, 0, 4, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0 ] ), Character( CharacterTable( "U6(2)" ),
  [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, 48, 0, 0, 16, 6, 0, 0, 0, 0,
      0, 0, 6, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0 ] ), Character( CharacterTable( "U6(2)" ),
  [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 48, 0, 0, 0, 16, 6, 0, 0, 0, 0,
      0, 0, 6, 0, 2, 0, 4, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0 ] ) ]
gap> PermCharInfo( u62, cand ).ATLAS;
[ "1a+22a+252a+616a+1155c+1386a+8064a+9240c",
  "1a+22a+252a+616a+1155b+1386a+8064a+9240b",
  "1a+22a+252a+616a+1155a+1386a+8064a+9240a" ]
gap> aut:= AutomorphismsOfTable( u62 );;  Size( aut );
24
gap> elms:= Filtered( Elements( aut ), x -> Order( x ) = 3 );
[ (10,11,12)(26,27,28)(40,41,42), (10,12,11)(26,28,27)(40,42,41) ]
gap> Position( cand, Permuted( cand[1], elms[1] ) );
3
gap> Position( cand, Permuted( cand[3], elms[1] ) );
2
```

The six fusions induce three different characters, they are conjugate under the action of the unique subgroup of order 3 in the group of table automorphisms of $U_6(2)$. The table automorphisms of order 3 are induced by group automorphisms of $U_6(2)$ (see [CCN+85, p. 120]). As can be seen from the list of maximal subgroups of $U_6(2)$ in [CCN+85, p. 115], the three induced characters are in fact permutation characters which belong to the three classes of maximal subgroups of type M_{22} in $U_6(2)$, which are permuted by an outer automorphism of order 3. Now we want to compute the extension of the above permutation character to the group $U_6(2).2$, which corresponds to the action of this group on the cosets of a $M_{22}.2$ subgroup.

```
─────────────────────────── Example ───────────────────────────
gap> u622:= CharacterTable( "U6(2).2" );;
gap> m222:= CharacterTable( "M22.2" );;
gap> fus:= PossibleClassFusions( m222, u622 );
[ [ 1, 3, 7, 10, 13, 14, 20, 22, 22, 24, 29, 38, 39, 42, 41, 46, 50,
      53, 58, 59, 59 ] ]
gap> cand:= Induced( m222, u622, [ TrivialCharacter( m222 ) ], fus[1] );
[ Character( CharacterTable( "U6(2).2" ),
  [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 48, 0, 0, 16, 6, 0, 0, 0, 0, 0,
      6, 0, 2, 0, 4, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1080, 72,
      0, 48, 8, 0, 0, 0, 18, 0, 0, 0, 8, 0, 0, 2, 0, 0, 0, 0, 2, 2,
      0, 0, 0, 0, 0, 0 ] ) ]
gap> PermCharInfo( u622, cand ).ATLAS;
[ "1a+22a+252a+616a+1155a+1386a+8064a+9240a" ]
```

We see that for the embedding of $M_{22}.2$ into $U_6(2).2$, the class fusion is unique, so we get a unique extension of one of the above permutation characters. This implies that exactly one class of maximal subgroups of type M_{22} extends to $M_{22}.2$ in a given group $U_6(2).2$.

8.4 Degree $20\,736$ Permutation Characters of $U_6(2)$

Now we show an alternative way to compute the characters dealt with in the previous example. This works also if the character table of the point stabilizer is not available. In this situation we can compute all those characters that have certain properties of permutation characters. Of course this may take much longer than the above computations, which needed only a few seconds. (The following calculations may need several hours, depending on the computer used.)

```
———————————————————————— Example ————————————————————————
 gap> cand:= PermChars( u62, rec( torso := [ 20736 ] ) );
 [ Character( CharacterTable( "U6(2)" ),
     [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, 0, 48, 0, 16, 6, 0, 0, 0,
       0, 0, 0, 6, 0, 2, 0, 0, 0, 4, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0 ] ), Character( CharacterTable( "U6(2)" ),
     [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, 48, 0, 0, 16, 6, 0, 0, 0,
       0, 0, 0, 6, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0 ] ), Character( CharacterTable( "U6(2)" ),
     [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 48, 0, 0, 0, 16, 6, 0, 0, 0,
       0, 0, 0, 6, 0, 2, 0, 4, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0 ] ) ) ]
```

For the next step, that is, the computation of the extension of the permutation character to $U_6(2).2$, we may use the above information, since the values on the inner classes are prescribed. The question which of the three candidates for $U_6(2)$ extends to $U_6(2).2$ depends on the choice of the class fusion of $U_6(2)$ into $U_6(2).2$. With respect to the class fusion that is stored on the GAP library table, the third candidate extends, as can be seen from the fact that this one is invariant under the permutation of conjugacy classes of $U_6(2)$ that is induced by the action of the chosen supergroup $U_6(2).2$.

```
———————————————————————— Example ————————————————————————
 gap> u622:= CharacterTable( "U6(2).2" );;
 gap> inv:= InverseMap( GetFusionMap( u62, u622 ) );
 [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, [ 11, 12 ], 13, 14, 15, [ 16, 17 ],
   18, 19, 20, 21, 22, 23, 24, 25, 26, [ 27, 28 ], [ 29, 30 ], 31, 32,
   [ 33, 34 ], [ 35, 36 ], 37, [ 38, 39 ], 40, [ 41, 42 ], 43, 44,
   [ 45, 46 ] ]
 gap> ext:= List( cand, x -> CompositionMaps( x, inv ) );
 [ [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, [ 0, 48 ], 0, 16, 6, 0, 0,
       0, 0, 0, 6, 0, 2, 0, 0, [ 0, 4 ], 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0 ],
   [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, [ 0, 48 ], 0, 16, 6, 0, 0,
       0, 0, 0, 6, 0, 2, 0, 0, [ 0, 4 ], 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0 ],
   [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 48, 0, 0, 16, 6, 0, 0, 0, 0, 0,
       6, 0, 2, 0, 4, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 ] ]
 gap> cand:= PermChars( u622, rec( torso:= ext[3] ) );
 [ Character( CharacterTable( "U6(2).2" ),
     [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 48, 0, 0, 16, 6, 0, 0, 0, 0,
```

```
        0, 6, 0, 2, 0, 4, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1080,
        72, 0, 48, 8, 0, 0, 0, 18, 0, 0, 0, 8, 0, 0, 2, 0, 0, 0, 0, 2,
        2, 0, 0, 0, 0, 0, 0 ] ) ]
```

8.5   Degree $57\,572\,775$ Permutation Characters of $O_8^+(3)$

The group $O_8^+(3)$ (see [CCN+85, p. 140]) contains a subgroup of type $2^{3+6}.L_3(2)$, which extends to a maximal subgroup U in $O_8^+(3).3$. For the computation of the permutation character, we cannot use explicit induction since the table of U is not available in the GAP table library. Since $U \cap O_8^+(3)$ is contained in a $O_8^+(2)$ subgroup of $O_8^+(3)$, we can try to find the permutation character of $O_8^+(2)$ corresponding to the action on the cosets of $U \cap O_8^+(3)$, and then induce this character to $O_8^+(3)$. This kind of computations becomes more difficult with increasing degree, so we try to reduce the problem further. In fact, the $2^{3+6}.L_3(2)$ group is contained in a $2^6 : A_8$ subgroup of $O_8^+(2)$, in which the index is only 15; the unique possible permutation character of this degree can be read off immediately. Induction to $O_8^+(3)$ through the chain of subgroups is possible provided the class fusions are available. There are 24 possible fusions from $O_8^+(2)$ into $O_8^+(3)$, which are all equivalent w.r.t. table automorphisms of $O_8^+(3)$. If we later want to consider the extension of the permutation character in question to $O_8^+(3).3$ then we have to choose a fusion of an $O_8^+(2)$ subgroup that does *not* extend to $O_8^+(2).3$. But if for example our question is just whether the resulting permutation character is multiplicity-free then this can be decided already from the permutation character of $O_8^+(3)$.

```
————————————————————————— Example —————————————————————————
  gap> o8p3:= CharacterTable("O8+(3)");;
  gap> Size( o8p3 ) / (2^9*168);
  57572775
  gap> o8p2:= CharacterTable( "O8+(2)" );;
  gap> fus:= PossibleClassFusions( o8p2, o8p3 );;
  gap> Length( fus );
  24
  gap> rep:= RepresentativesFusions( o8p2, fus, o8p3 );
  [ [ 1, 5, 2, 3, 4, 5, 7, 8, 12, 16, 17, 19, 23, 20, 21, 22, 23, 24,
        25, 26, 37, 38, 42, 31, 32, 36, 49, 52, 51, 50, 43, 44, 45, 53,
        55, 56, 57, 71, 71, 71, 72, 73, 74, 78, 79, 83, 88, 89, 90, 94,
        100, 101, 105 ] ]
  gap> fus:= rep[1];;
  gap> Size( o8p2 ) / (2^9*168);
  2025
  gap> sub:= CharacterTable( "2^6:A8" );;
  gap> subfus:= GetFusionMap( sub, o8p2 );
  [ 1, 3, 2, 2, 4, 5, 6, 13, 3, 6, 12, 13, 14, 7, 21, 24, 11, 30, 29,
    31, 13, 17, 15, 16, 14, 17, 36, 37, 18, 41, 24, 44, 48, 28, 33, 32,
    34, 35, 35, 51, 51 ]
  gap> fus:= CompositionMaps( fus, subfus );
  [ 1, 2, 5, 5, 3, 4, 5, 23, 2, 5, 19, 23, 20, 7, 37, 31, 17, 50, 51,
    43, 23, 23, 21, 22, 20, 23, 56, 57, 24, 72, 31, 78, 89, 52, 45, 44,
    53, 55, 55, 100, 100 ]
  gap> Size( sub ) / (2^9*168);
  15
  gap> List( Irr( sub ), Degree );
  [ 1, 7, 14, 20, 21, 21, 21, 28, 35, 45, 45, 56, 64, 70, 28, 28, 35,
```

```
     35, 35, 35, 70, 70, 70, 70, 140, 140, 140, 140, 140, 210, 210, 252,
     252, 280, 280, 315, 315, 315, 315, 420, 448 ]
gap> cand:= PermChars( sub, 15 );
[ Character( CharacterTable( "2^6:A8" ),
   [ 15, 15, 15, 7, 7, 7, 7, 7, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3,
       3, 3, 3, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0 ] ) ]
gap> ind:= Induced( sub, o8p3, cand, fus );
[ Character( CharacterTable( "O8+(3)" ),
   [ 57572775, 59535, 59535, 59535, 3591, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 2187, 0, 27, 135, 135, 135, 243, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 27, 27, 27, 0, 0, 0, 0, 27,
       27, 27, 27, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0 ] ) ]
gap> o8p33:= CharacterTable( "O8+(3).3" );;
gap> inv:= InverseMap( GetFusionMap( o8p3, o8p33 ) );
[ 1, [ 2, 3, 4 ], 5, 6, [ 7, 8, 9 ], [ 10, 11, 12 ], 13,
   [ 14, 15, 16 ], 17, 18, 19, [ 20, 21, 22 ], 23, [ 24, 25, 26 ],
   [ 27, 28, 29 ], 30, [ 31, 32, 33 ], [ 34, 35, 36 ], [ 37, 38, 39 ],
   [ 40, 41, 42 ], [ 43, 44, 45 ], 46, [ 47, 48, 49 ], 50,
   [ 51, 52, 53 ], 54, 55, 56, 57, [ 58, 59, 60 ], [ 61, 62, 63 ], 64,
   [ 65, 66, 67 ], 68, [ 69, 70, 71 ], [ 72, 73, 74 ], [ 75, 76, 77 ],
   [ 78, 79, 80 ], [ 81, 82, 83 ], 84, 85, [ 86, 87, 88 ],
   [ 89, 90, 91 ], [ 92, 93, 94 ], 95, 96, [ 97, 98, 99 ],
   [ 100, 101, 102 ], [ 103, 104, 105 ], [ 106, 107, 108 ],
   [ 109, 110, 111 ], [ 112, 113, 114 ] ]
gap> ext:= CompositionMaps( ind[1], inv );
[ 57572775, 59535, 3591, 0, 0, 0, 0, 0, 2187, 0, 27, 135, 243, 0, 0,
   0, 0, 0, 0, 0, 27, 0, 0, 27, 27, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> perms:= PermChars( o8p33, rec( torso:= ext ) );
[ Character( CharacterTable( "O8+(3).3" ),
   [ 57572775, 59535, 3591, 0, 0, 0, 0, 0, 2187, 0, 27, 135, 243, 0,
       0, 0, 0, 0, 0, 27, 0, 0, 27, 27, 0, 8, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3159,
       3159, 243, 243, 39, 39, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 3,
       3, 3, 3, 0, 0, 0, 0, 0, 0, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,
       0, 0 ] ) ]
gap> PermCharInfo( o8p33, perms ).ATLAS;
[ "1a+780aabb+2457a+2808abc+9450aaabbcc+18200abcdddef+24192a+54600a^{5\
}b+70200aabb+87360ab+139776a^{5}+147420a^{4}b^{4}+163800ab+184275aabc+\
199017aa+218700a+245700a+291200aef+332800a^{4}b^{5}c^{5}+491400aaabcd+\
531441a^{5}b^{4}c^{4}+552825a^{4}+568620aabb+698880a^{4}b^{4}+716800aa\
abbccdddeeff+786240aabb+873600aa+998400aa+1257984a^{6}+1397760aa" ]
```

8.6   The Action of $O_7(3).2$ on the Cosets of $2^7.S_7$

We want to know whether the permutation character of $O_7(3).2$ (see [CCN+85, p. 108]) on the cosets of its maximal subgroup U of type $2^7.S_7$ is multiplicity-free. As in the previous examples, first we try to compute the permutation character of the simple group $O_7(3)$. It turns out that the direct

computation of all candidates from the degree is very time consuming. But we can use for example the additional information provided by the fact that U contains an A_7 subgroup. We compute the possible class fusions.

```
—————— Example ——————
  gap> o73:= CharacterTable( "O7(3)" );;
  gap> a7:= CharacterTable( "A7" );;
  gap> fus:= PossibleClassFusions( a7, o73 );
  [ [ 1, 3, 6, 10, 15, 16, 24, 33, 33 ],
    [ 1, 3, 7, 10, 15, 16, 22, 33, 33 ] ]
```

We cannot decide easily which fusion is the right one, but already the fact that no other fusions are possible gives us some information about impossible constituents of the permutation character we want to compute.

```
—————— Example ——————
  gap> ind:= List( fus,
  >        x -> Induced( a7, o73, [ TrivialCharacter( a7 ) ], x )[1] );;
  gap> mat:= MatScalarProducts( o73, Irr( o73 ), ind );;
  gap> sum:= Sum( mat );
  [ 2, 6, 2, 0, 8, 6, 2, 4, 4, 8, 3, 0, 4, 4, 9, 3, 5, 0, 0, 9, 0, 10,
    5, 6, 15, 1, 12, 1, 15, 7, 2, 4, 14, 16, 0, 12, 12, 7, 8, 8, 14,
    12, 12, 14, 6, 6, 20, 16, 12, 12, 12, 10, 10, 12, 12, 8, 12, 6 ]
  gap> const:= Filtered( [ 1 .. Length( sum ) ], x -> sum[x] <> 0 );
  [ 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 20, 22, 23, 24,
    25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42,
    43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58 ]
  gap> Length( const );
  52
  gap> const:= Irr( o73 ){ const };;
  gap> rat:= RationalizedMat( const );;
```

But much more can be deduced from the fact that certain zeros of the permutation character can be predicted.

```
—————— Example ——————
  gap> names:= ClassNames( o73 );
  [ "1a", "2a", "2b", "2c", "3a", "3b", "3c", "3d", "3e", "3f", "3g",
    "4a", "4b", "4c", "4d", "5a", "6a", "6b", "6c", "6d", "6e", "6f",
    "6g", "6h", "6i", "6j", "6k", "6l", "6m", "6n", "6o", "6p", "7a",
    "8a", "8b", "9a", "9b", "9c", "9d", "10a", "10b", "12a", "12b",
    "12c", "12d", "12e", "12f", "12g", "12h", "13a", "13b", "14a",
    "15a", "18a", "18b", "18c", "18d", "20a" ]
  gap> List( fus, x -> names{ x } );
  [ [ "1a", "2b", "3b", "3f", "4d", "5a", "6h", "7a", "7a" ],
    [ "1a", "2b", "3c", "3f", "4d", "5a", "6f", "7a", "7a" ] ]
  gap> torso:= [ 28431 ];;
  gap> zeros:= [ 5, 8, 9, 11, 17, 20, 23, 28, 29, 32, 36, 37, 38,
  >              43, 46, 47, 48, 53, 54, 55, 56, 57, 58 ];;
  gap> names{ zeros };
  [ "3a", "3d", "3e", "3g", "6a", "6d", "6g", "6l", "6m", "6p", "9a",
    "9b", "9c", "12b", "12e", "12f", "12g", "15a", "18a", "18b", "18c",
    "18d", "20a" ]
```

Every order 3 element of U lies in an A_7 subgroup of U, so among the classes of element order 3, at most the classes 3B, 3C, and 3F can have nonzero permutation character values. The excluded classes of element order 6 are the square roots of the excluded order 3 elements, likewise the given classes of element orders 9, 12, and 18 are excluded. The character value on 20A must be zero because U does not contain elements of this order. So we enter the additional information about these zeros.

```
────────────────────────── Example ──────────────────────────
  gap> for i in zeros do
  >        torso[i]:= 0;
  >    od;
  gap> torso;
  [ 28431,,,, 0,,, 0, 0,, 0,,,,,, 0,,, 0,,, 0,,,,, 0, 0,,, 0,,,, 0, 0,
    0,,,,, 0,,, 0, 0, 0,,,,, 0, 0, 0, 0, 0, 0 ]
  gap> perms:= PermChars( o73, rec( torso:= torso, chars:= rat ) );
  [ Character( CharacterTable( "O7(3)" ),
    [ 28431, 567, 567, 111, 0, 0, 243, 0, 0, 81, 0, 15, 3, 27, 15, 6,
        0, 0, 27, 0, 3, 27, 0, 0, 0, 3, 9, 0, 0, 3, 3, 0, 4, 1, 1, 0,
        0, 0, 0, 2, 2, 3, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0 ] ) ]
  gap> PermCharInfo( o73, perms ).ATLAS;
  [ "1a+78a+168a+182a+260ab+1092a+2457a+2730a+4095b+5460a+11648a" ]
```

We see that this character is already multiplicity free, so this holds also for its extension to $O_7(3).2$, and we need not compute this extension. (Of course we could compute it in the same way as in the examples above.)

8.7   The Action of $O_8^+(3).2_1$ on the Cosets of $2^7.A_8$

We are interested in the permutation character of $O_8^+(3).2_1$ that corresponds to the action on the cosets of a subgroup of type $2^7.A_8$. The intersection of the point stabilizer with the simple group $O_8^+(3)$ is of type $2^6.A_8$. First we compute the class fusion of these groups, modulo problems with ambiguities due to table automorphisms.

```
────────────────────────── Example ──────────────────────────
  gap> o8p3:= CharacterTable( "O8+(3)" );;
  gap> o8p2:= CharacterTable( "O8+(2)" );;
  gap> fus:= PossibleClassFusions( o8p2, o8p3 );;
  gap> NamesOfFusionSources( o8p2 );
  [ "A9", "2^8:O8+(2)", "(D10xD10).2^2", "(3x3^3:S3):2",
    "(3x3^(1+2)+:2A4).2", "2^(3+3+3).L3(2)", "NRS(O8+(2),2^(3+3+3)_a)",
    "NRS(O8+(2),2^(3+3+3)_b)", "O8+(2)N2", "O8+(2)M2", "O8+(2)M3",
    "O8+(2)M5", "O8+(2)M6", "O8+(2)M8", "O8+(2)M9", "(3xU4(2)):2",
    "O8+(2)M11", "O8+(2)M12", "2^(1+8)_+:(S3xS3xS3)", "3^4:2^3.S4(a)",
    "(A5xA5):2^2", "O8+(2)M16", "O8+(2)M17", "2^(1+8)+.O8+(2)", "7:6",
    "(A5xD10).2", "(D10xA5).2", "O8+(2)N5C", "2^6:A8", "2.O8+(2)",
    "2^2.O8+(2)", "S6(2)" ]
  gap> sub:= CharacterTable( "2^6:A8" );;
  gap> subfus:= GetFusionMap( sub, o8p2 );
  [ 1, 3, 2, 2, 4, 5, 6, 13, 3, 6, 12, 13, 14, 7, 21, 24, 11, 30, 29,
    31, 13, 17, 15, 16, 14, 17, 36, 37, 18, 41, 24, 44, 48, 28, 33, 32,
    34, 35, 35, 51, 51 ]
  gap> fus:= List( fus, x -> CompositionMaps( x, subfus ) );;
```

```
gap> fus:= Set( fus );;
gap> Length( fus );
24
```

The ambiguities due to Galois automorphisms disappear when we are looking for the permutation characters induced by the fusions.

```
———————————————— Example ————————————————
gap> ind:= List( fus, x -> Induced( sub, o8p3,
>                                   [ TrivialCharacter( sub ) ], x )[1] );;
gap> ind:= Set( ind );;
gap> Length( ind );
6
```

Now we try to extend the candidates to $O_8^+(3).2_1$; the choice of the fusion of $O_8^+(3)$ into $O_8^+(3).2_1$ determines which of the candidates may extend.

```
———————————————— Example ————————————————
gap> o8p32:= CharacterTable( "O8+(3).2_1" );;
gap> fus:= GetFusionMap( o8p3, o8p32 );;
gap> ext:= List( ind, x -> CompositionMaps( x, InverseMap( fus ) ) );;
gap> ext:= Filtered( ext, x -> ForAll( x, IsInt ) );
[ [ 3838185, 17577, 8505, 8505, 873, 0, 0, 0, 0, 6561, 0, 0, 729, 0,
      9, 105, 45, 45, 105, 30, 0, 0, 0, 0, 0, 0, 0, 0, 0, 189, 0, 0,
      0, 9, 9, 27, 27, 0, 0, 27, 9, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0,
      0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0,
      0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0 ],
  [ 3838185, 17577, 8505, 8505, 873, 0, 6561, 0, 0, 0, 0, 0, 729, 0,
      9, 105, 45, 45, 105, 30, 0, 0, 0, 0, 0, 0, 189, 0, 0, 0, 9, 0,
      0, 0, 9, 27, 27, 0, 0, 9, 27, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0,
      0, 2, 0, 0, 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
      0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ]
```

We compute the extensions of the first candidate; the other belongs to another class of subgroups, which is the image under an outer automorphism.

```
———————————————— Example ————————————————
gap> perms:= PermChars( o8p32, rec( torso:= ext[1] ) );
[ Character( CharacterTable( "O8+(3).2_1" ),
  [ 3838185, 17577, 8505, 8505, 873, 0, 0, 0, 0, 6561, 0, 0, 729, 0,
      9, 105, 45, 45, 105, 30, 0, 0, 0, 0, 0, 0, 0, 0, 0, 189, 0, 0,
      0, 9, 9, 27, 27, 0, 0, 27, 9, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0,
      0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0,
      0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 3159, 1575, 567, 63, 87,
      15, 0, 0, 45, 0, 81, 9, 27, 0, 0, 3, 3, 3, 3, 5, 5, 0, 0, 0, 4,
      0, 0, 27, 0, 9, 0, 0, 15, 0, 3, 0, 0, 2, 0, 0, 0, 0, 0, 3, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
gap> PermCharInfo( o8p32, perms ).ATLAS;
[ "1a+260abc+520ab+819a+2808b+9450aab+18200a+23400ac+29120b+36400aab+4\
6592abce+49140d+66339a+98280ab+163800a+189540d+232960d+332800ab+368550\
a+419328a+531441ab" ]
```

Now we repeat the calculations for $O_8^+(3).2_2$ instead of $O_8^+(3).2_1$.

```
_____ Example _____
gap> o8p32:= CharacterTable( "O8+(3).2_2" );;
gap> fus:= GetFusionMap( o8p3, o8p32 );;
gap> ext:= List( ind, x -> CompositionMaps( x, InverseMap( fus ) ) );;
gap> ext:= Filtered( ext, x -> ForAll( x, IsInt ) );;
gap> perms:= PermChars( o8p32, rec( torso:= ext[1] ) );
[ Character( CharacterTable( "O8+(3).2_2" ),
  [ 3838185, 17577, 8505, 873, 0, 0, 0, 6561, 0, 0, 0, 0, 729, 0, 9,
      105, 45, 105, 30, 0, 0, 0, 0, 0, 0, 189, 0, 0, 0, 9, 0, 9, 27,
      0, 0, 0, 27, 27, 9, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      2, 0, 0, 0, 0, 0, 9, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
      0, 6, 0, 0, 0, 0, 0, 0, 199017, 2025, 297, 441, 73, 9, 0,
      1215, 0, 0, 0, 0, 81, 0, 0, 0, 0, 27, 27, 0, 1, 9, 12, 0, 0,
      45, 0, 0, 1, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0,
      0, 0, 0, 0, 0 ] ) ]
gap> PermCharInfo( o8p32, perms ).ATLAS;
[ "1a+260aac+520ab+819a+2808a+9450aaa+18200accee+23400ac+29120a+36400a\
+46592aa+49140c+66339a+93184a+98280ab+163800a+184275ac+189540c+232960c\
+332800aa+419328a+531441aa" ]
```

We might be interested in the extension to $O_8^+(3).(2^2)_{122}$. It is clear that this cannot be multiplicity free because of the multiplicity 9450aaa in the character induced from $O_8^+(3).2_2$. We could put the extensions to the index two subgroups together, but it is simpler (and not expensive) to run the same program as above.

```
_____ Example _____
gap> o8p322:= CharacterTable( "O8+(3).(2^2)_{122}" );;
gap> fus:= GetFusionMap( o8p32, o8p322 );;
gap> ext:= List( perms, x -> CompositionMaps( x, InverseMap( fus ) ) );;
gap> ext:= Filtered( ext, x -> ForAll( x, IsInt ) );;
gap> perms:= PermChars( o8p322, rec( torso:= ext[1] ) );
[ Character( CharacterTable( "O8+(3).(2^2)_{122}" ),
  [ 3838185, 17577, 8505, 873, 0, 0, 0, 6561, 0, 0, 729, 0, 9, 105,
      45, 105, 30, 0, 0, 0, 0, 0, 0, 189, 0, 0, 9, 9, 27, 0, 0, 27,
      9, 0, 8, 1, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 9, 0, 0,
      0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 3159, 1575,
      567, 63, 87, 15, 0, 0, 45, 0, 81, 9, 27, 0, 0, 3, 3, 3, 5, 0,
      0, 4, 0, 0, 27, 0, 9, 0, 0, 15, 0, 3, 0, 0, 2, 0, 0, 0, 3, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 199017, 2025, 297, 441, 73, 9, 0,
      1215, 0, 0, 0, 0, 81, 0, 0, 0, 27, 27, 0, 1, 9, 12, 0, 0, 45,
      0, 0, 1, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0,
      0, 28431, 1647, 135, 63, 87, 39, 0, 0, 243, 27, 0, 0, 81, 63,
      0, 0, 0, 9, 0, 3, 3, 6, 2, 0, 0, 0, 9, 0, 0, 3, 3, 3, 0, 4, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0 ] ) ]
gap> PermCharInfo( o8p322, perms ).ATLAS;
[ "1a+260ace+819a+1040a+2808c+9450aac+18200a+23400ae+29120c+36400aac+4\
6592ac+49140g+66339a+93184a+163800b+189540g+196560a+232960g+332800ac+3\
68550a+419328a+531441ac" ]
```

8.8   The Action of $S_4(4).4$ on the Cosets of $5^2.[2^5]$

We want to know whether the permutation character corresponding to the action of $S_4(4).4$ (see [CCN+85, p. 44]) on the cosets of its maximal subgroup of type $5^2 : [2^5]$ is multiplicity free. The library names of subgroups for which the class fusions are stored are listed as value of the attribute `NamesOfFusionSources` (**Reference: NamesOfFusionSources**), and for groups whose isomorphism type is not determined by the name this is the recommended way to find out whether the table of the subgroup is contained in the GAP library and known to belong to this group. (It might be that a table with such a name is contained in the library but belongs to another group, and it may also be that the table of the group is contained in the library --with any name-- but it is not known that this group is isomorphic to a subgroup of $S_4(4).4$.)

```
————————————————————— Example —————————————————————
 gap> s444:= CharacterTable( "S4(4).4" );;
 gap> NamesOfFusionSources( s444 );
 [ "(L3(2)xS4(4):2).2", "S4(4)", "S4(4).2" ]
```

So we cannot simply fetch the table of the subgroup. As in the previous examples, we compute the possible permutation characters.

```
————————————————————— Example —————————————————————
 gap> perms:= PermChars( s444,
 >                    rec( torso:= [ Size( s444 ) / ( 5^2*2^5 ) ] ) );
 [ Character( CharacterTable( "S4(4).4" ),
   [ 4896, 384, 96, 0, 16, 32, 36, 16, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ),
   Character( CharacterTable( "S4(4).4" ),
   [ 4896, 192, 32, 0, 0, 8, 6, 1, 0, 2, 0, 0, 36, 0, 12, 0, 0, 0, 1,
       0, 6, 6, 2, 2, 0, 0, 0, 0, 1, 1 ] ),
   Character( CharacterTable( "S4(4).4" ),
   [ 4896, 240, 64, 0, 8, 8, 36, 16, 0, 0, 0, 0, 0, 12, 8, 0, 4, 4, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
```

So there are three candidates. None of them is multiplicity free, so we need not decide which of the candidates actually belongs to the group $5^2 : [2^5]$ we have in mind.

```
————————————————————— Example —————————————————————
 gap> PermCharInfo( s444, perms ).ATLAS;
 [ "1abcd+50abcd+153abcd+170a^{4}b^{4}+680aabb",
   "1a+50ac+153a+170aab+256a+680abb+816a+1020a",
   "1ac+50ac+68a+153abcd+170aabbb+204a+680abb+1020a" ]
```

(If we would be interested which candidate is the right one, we could for example look at the intersection with $S_4(4)$, and hope for a contradiction to the fact that the group must lie in a $(A_5 \times A_5) : 2$ subgroup.)

8.9   The Action of Co_1 on the Cosets of Involution Centralizers

We compute the permutation characters of the sporadic simple Conway group Co_1 (see [CCN+85, p. 180]) corresponding to the actions on the cosets of involution centralizers. Equivalently, we are interested in the action of Co_1 on conjugacy classes of involutions. These characters can be computed as follows. First we take the table of Co_1.

```
──────────────────────── Example ────────────────────────
  gap> t:= CharacterTable( "Co1" );
  CharacterTable( "Co1" )
```

The centralizer of each 2A element is a maximal subgroup of Co_1. This group is also contained in the table library. So we can compute the permutation character by explicit induction, and the decomposition in irreducibles is computed with the command `PermCharInfo` (**Reference: PermCharInfo**).

```
──────────────────────── Example ────────────────────────
  gap> s:= CharacterTable( Maxes( t )[5] );
  CharacterTable( "2^(1+8)+.O8+(2)" )
  gap> ind:= Induced( s, t, [ TrivialCharacter( s ) ] );;
  gap> PermCharInfo( t, ind ).ATLAS;
  [ "1a+299a+17250a+27300a+80730a+313950a+644644a+2816856a+5494125a+1243\
  2420a+24794000a" ]
```

The centralizer of a 2B element is not maximal. First we compute which maximal subgroup can contain it. The character tables of all maximal subgroups of Co_1 are contained in the GAP's table library, so we may take these tables and look at the group orders.

```
──────────────────────── Example ────────────────────────
  gap> centorder:= SizesCentralizers( t )[3];;
  gap> maxes:= List( Maxes( t ), CharacterTable );;
  gap> cand:= Filtered( maxes, x -> Size( x ) mod centorder = 0 );
  [ CharacterTable( "(A4xG2(4)):2" ) ]
  gap> u:= cand[1];;
  gap> index:= Size( u ) / centorder;
  3
```

So there is a unique class of maximal subgroups containing the centralizer of a 2B element, as a subgroup of index 3. We compute the unique permutation character of degree 3 of this group, and induce this character to G.

```
──────────────────────── Example ────────────────────────
  gap> subperm:= PermChars( u, rec( degree := index, bounds := false ) );
  [ Character( CharacterTable( "(A4xG2(4)):2" ),
    [ 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
        3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3,
        3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1 ] ) ]
  gap> subperm = PermChars( u, rec( torso := [ 3 ] ) );
  true
  gap> ind:= Induced( u, t, subperm );
  [ Character( CharacterTable( "Co1" ),
    [ 2065694400, 181440, 119408, 38016, 2779920, 0, 0, 378, 30240,
        864, 0, 720, 316, 80, 2520, 30, 0, 6480, 1508, 0, 0, 0, 0, 0,
        38, 18, 105, 0, 600, 120, 56, 24, 0, 12, 0, 0, 0, 120, 48, 18,
        0, 0, 6, 0, 360, 144, 108, 0, 0, 10, 0, 0, 0, 0, 0, 4, 2, 3, 9,
        0, 0, 15, 3, 0, 0, 4, 4, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0,
        12, 8, 0, 6, 0, 0, 3, 0, 1, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3,
        0 ] ) ]
```

```
gap> PermCharInfo( t, ind ).ATLAS;
[ "1a+1771a+8855a+27300aa+313950a+345345a+644644aa+871884aaa+1771000a+\
2055625a+4100096a+7628985a+9669660a+12432420aa+21528000aa+23244375a+24\
174150aa+24794000a+31574400aa+40370176a+60435375a+85250880aa+100725625\
a+106142400a+150732800a+184184000a+185912496a+207491625a+299710125a+30\
2176875a" ]
```

Finally, we try the same for the centralizer of a 2C element.

```
─────────── Example ───────────
gap> centorder:= SizesCentralizers( t )[4];;
gap> cand:= Filtered( maxes, x -> Size( x ) mod centorder = 0 );
[ CharacterTable( "Co2" ), CharacterTable( "2^11:M24" ) ]
```

The group order excludes all except two classes of maximal subgroups. But the 2C centralizer cannot lie in Co_2 because the involution centralizers in Co_2 are too small.

```
─────────── Example ───────────
gap> u:= cand[1];;
gap> GetFusionMap( u, t );
[ 1, 2, 2, 4, 7, 6, 9, 11, 11, 10, 11, 12, 14, 17, 16, 21, 23, 20,
  22, 22, 24, 28, 30, 33, 31, 32, 33, 33, 37, 42, 41, 43, 44, 48, 52,
  49, 53, 55, 53, 52, 54, 60, 60, 60, 64, 65, 65, 67, 66, 70, 73, 72,
  78, 79, 84, 85, 87, 92, 93, 93 ]
gap> centorder;
389283840
gap> SizesCentralizers( u )[4];
1474560
```

So we try the second candidate.

```
─────────── Example ───────────
gap> u:= cand[2];
CharacterTable( "2^11:M24" )
gap> index:= Size( u ) / centorder;
1288
gap> subperm:= PermChars( u, rec( torso := [ index ] ) );
[ Character( CharacterTable( "2^11:M24" ),
  [ 1288, 1288, 1288, 56, 56, 56, 56, 56, 56, 48, 48, 48, 48, 48, 10,
      10, 10, 10, 7, 7, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 4, 4,
      4, 4, 3, 3, 3, 2, 2, 2, 2, 2, 2, 3, 3, 3, 0, 0, 0, 0, 2, 2, 2,
      2, 3, 3, 3, 1, 1, 2, 2, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0 ] ) ]
gap> subperm = PermChars( u, rec( degree:= index, bounds := false ) );
true
gap> ind:= Induced( u, t, subperm );
[ Character( CharacterTable( "Co1" ),
  [ 10680579000, 1988280, 196560, 94744, 0, 17010, 0, 945, 7560,
      3432, 2280, 1728, 252, 308, 0, 225, 0, 0, 0, 270, 0, 306, 0,
      46, 45, 25, 0, 0, 120, 32, 12, 52, 36, 36, 0, 0, 0, 0, 0, 45,
      15, 0, 9, 3, 0, 0, 0, 0, 18, 0, 30, 0, 6, 18, 0, 3, 5, 0, 0, 0,
      0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 3, 0, 0, 0, 0, 1, 0, 0, 0, 0,
      6, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
```

```
gap> PermCharInfo( t, ind ).ATLAS;
[ "1a+17250aa+27300a+80730aa+644644aaa+871884a+1821600a+2055625aaa+281\
6856a+5494125a^{4}+12432420aa+16347825aa+23244375a+24174150aa+24667500\
aa+24794000aaa+31574400a+40370176a+55255200a+66602250a^{4}+83720000aa+\
85250880aaa+91547820aa+106142400a+150732800a+184184000aaa+185912496aaa\
+185955000aaa+207491625aaa+215547904aa+241741500aaa+247235625a+2578576\
00aa+259008750a+280280000a+302176875a+326956500a+387317700a+402902500a\
+464257024a+469945476b+502078500a+503513010a+504627200a+522161640a" ]
```

8.10   The Multiplicity Free Permutation Characters of $G_2(3)$

We compute the multiplicity free possible permutation characters of $G_2(3)$ (see [CCN+85, p. 60]). For each divisor d of the group order, we compute all those possible permutation characters of degree d of G for which each irreducible constituent occurs with multiplicity at most 1; this is done by prescribing the `maxmult` component of the second argument of `PermChars` (**Reference: PermChars**) to be the list with 1 at each position.

```
————————————————————— Example ———————————————————————
gap> t:= CharacterTable( "G2(3)" );
CharacterTable( "G2(3)" )
gap> t:= CharacterTable( "G2(3)" );;
gap> n:= Length( RationalizedMat( Irr( t ) ) );;
gap> maxmult:= List( [ 1 .. n ], i -> 1 );;
gap> perms:= [];;
gap> divs:= DivisorsInt( Size( t ) );;
gap> for d in divs do
>        Append( perms,
>                PermChars( t, rec( bounds  := false,
>                                   degree  := d,
>                                   maxmult := maxmult ) ) );
>    od;
gap> Length( perms );
42
gap> List( perms, Degree );
[ 1, 351, 351, 364, 364, 378, 378, 546, 546, 546, 546, 546, 702, 702,
  728, 728, 1092, 1092, 1092, 1092, 1092, 1092, 1092, 1092, 1456,
  1456, 1638, 1638, 2184, 2184, 2457, 2457, 2457, 2457, 3159, 3276,
  3276, 3276, 3276, 4368, 6552, 6552 ]
```

For finding out which of these candidates are really permutation characters, we could inspect them piece by piece, using the information in [CCN+85]. For example, the candidates of degrees 351, 364, and 378 are induced from the trivial characters of maximal subgroups of G, whereas the candidates of degree 546 are not permutation characters.

Since the table of marks of G is available in GAP, we can extract all permutation characters from the table of marks, and then filter out the multiplicity free ones.

```
————————————————————— Example ———————————————————————
gap> tom:= TableOfMarks( "G2(3)" );
TableOfMarks( "G2(3)" )
gap> tbl:= CharacterTable( "G2(3)" );
CharacterTable( "G2(3)" )
```

```
gap> permstom:= PermCharsTom( tbl, tom );;
gap> Length( permstom );
433
gap> multfree:= Intersection( perms, permstom );;
gap> Length( multfree );
15
gap> List( multfree, Degree );
[ 1, 351, 351, 364, 364, 378, 378, 702, 702, 728, 728, 1092, 1092,
  2184, 2184 ]
```

8.11 Degree $11\,200$ Permutation Characters of $O_8^+(2)$

We compute the primitive permutation characters of degree 11200 of $O_8^+(2)$ and $O_8^+(2).2$ (see [CCN$^+$85, p. 85]). The character table of the maximal subgroup of type $3^4 : 2^3.S_4$ in $O_8^+(2)$ is not available in the GAP table library. But the group extends to a wreath product of S_3 and S_4 in the group $O_8^+(2).2$, and the table of this wreath product can be constructed easily.

```
——————————————— Example ———————————————
gap> tbl2:= CharacterTable("O8+(2).2");;
gap> s3:= CharacterTable( "Symmetric", 3 );;
gap> s:= CharacterTableWreathSymmetric( s3, 4 );
CharacterTable( "Sym(3)wrS4" )
```

The permutation character pi of $O_8^+(2).2$ can thus be computed by explicit induction, and the character of $O_8^+(2)$ is obtained by restriction of pi.

```
——————————————— Example ———————————————
gap> fus:= PossibleClassFusions( s, tbl2 );
[ [ 1, 41, 6, 3, 48, 9, 42, 19, 51, 8, 5, 50, 24, 49, 7, 2, 44, 22,
      42, 12, 53, 17, 58, 21, 5, 47, 26, 50, 37, 52, 23, 60, 18, 4,
      46, 25, 14, 61, 20, 9, 53, 30, 51, 26, 64, 8, 52, 31, 13, 56,
      38 ] ]
gap> pi:= Induced( s, tbl2, [ TrivialCharacter( s ) ], fus[1] )[1];
Character( CharacterTable( "O8+(2).2" ),
 [ 11200, 256, 160, 160, 80, 40, 40, 76, 13, 0, 0, 8, 8, 4, 0, 0, 16,
   16, 4, 4, 4, 1, 1, 1, 1, 5, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 2, 2, 0,
   0, 1120, 96, 0, 16, 0, 16, 8, 10, 4, 6, 7, 12, 3, 0, 0, 2, 0, 4, 0,
   1, 1, 0, 0, 1, 0, 0, 0 ] )
gap> PermCharInfo( tbl2, pi ).ATLAS;
[ "1a+84a+168a+175a+300a+700c+972a+1400a+3200a+4200b" ]
gap> tbl:= CharacterTable( "O8+(2)" );
CharacterTable( "O8+(2)" )
gap> rest:= RestrictedClassFunction( pi, tbl );
Character( CharacterTable( "O8+(2)" ),
 [ 11200, 256, 160, 160, 160, 80, 40, 40, 40, 76, 13, 0, 0, 8, 8, 8,
   4, 0, 0, 0, 16, 16, 16, 4, 4, 4, 4, 1, 1, 1, 1, 1, 1, 5, 0, 0, 0,
   1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0 ] )
gap> PermCharInfo( tbl, rest ).ATLAS;
[ "1a+84abc+175a+300a+700bcd+972a+3200a+4200a" ]
```

8.12   A Proof of Nonexistence of a Certain Subgroup

We prove that the sporadic simple Mathieu group $G = M_{22}$ (see [CCN$^+$85, p. 39]) has no subgroup of index 56. In [Isa76], remark after Theorem 5.18, this is stated as an example of the case that a character may be a possible permutation character but not a permutation character. Let us consider the possible permutation character of degree 56 of G.

```
────────────────── Example ──────────────────
 gap> tbl:= CharacterTable( "M22" );
 CharacterTable( "M22" )
 gap> perms:= PermChars( tbl, rec( torso:= [ 56 ] ) );
 [ Character( CharacterTable( "M22" ),
   [ 56, 8, 2, 4, 0, 1, 2, 0, 0, 2, 1, 1 ] ) ]
 gap> pi:= perms[1];;
 gap> Norm( pi );
 2
 gap> Display( tbl, rec( chars:= perms ) );
 M22

     2  7  7  2  5  4  .  2  .  .  3  .  .
     3  2  1  2  .  .  .  1  .  .  .  .  .
     5  1  .  .  .  .  1  .  .  .  .  .  .
     7  1  .  .  .  .  .  .  1  1  .  .  .
    11  1  .  .  .  .  .  .  .  .  .  1  1

        1a 2a 3a 4a 4b 5a 6a 7a 7b 8a 11a 11b
     2P 1a 1a 3a 2a 2a 5a 3a 7a 7b 4a 11b 11a
     3P 1a 2a 1a 4a 4b 5a 2a 7b 7a 8a 11a 11b
     5P 1a 2a 3a 4a 4b 1a 6a 7b 7a 8a 11a 11b
     7P 1a 2a 3a 4a 4b 5a 6a 1a 1a 8a 11b 11a
    11P 1a 2a 3a 4a 4b 5a 6a 7a 7b 8a  1a  1a

 Y.1    56  8  2  4  .  1  2  .  .  2  1  1
```

Suppose that `pi` is a permutation character of G. Since G is 2-transitive on the 56 cosets of the point stabilizer S, this stabilizer is transitive on 55 points, and thus G has a subgroup U of index $56 \cdot 55 = 3080$. We compute the possible permutation character of this degree.

```
────────────────── Example ──────────────────
 gap> perms:= PermChars( tbl, rec( torso:= [ 56 * 55 ] ) );;
 gap> Length( perms );
 16
```

U is contained in S, so only those candidates must be considered that vanish on all classes where `pi` vanishes. Furthermore, the index of U in S is odd, so the Sylow 2 subgroups of U and S are isomorphic; S contains elements of order 8, hence also U does.

```
────────────────── Example ──────────────────
 gap> OrdersClassRepresentatives( tbl );
 [ 1, 2, 3, 4, 4, 5, 6, 7, 7, 8, 11, 11 ]
 gap> perms:= Filtered( perms, x -> x[5] = 0 and x[10] <> 0 );
 [ Character( CharacterTable( "M22" ),
   [ 3080, 56, 2, 12, 0, 0, 2, 0, 0, 2, 0, 0 ] ),
```

```
Character( CharacterTable( "M22" ),
[ 3080, 8, 2, 8, 0, 0, 2, 0, 0, 4, 0, 0 ] ),
Character( CharacterTable( "M22" ),
[ 3080, 24, 11, 4, 0, 0, 3, 0, 0, 2, 0, 0 ] ),
Character( CharacterTable( "M22" ),
[ 3080, 24, 20, 4, 0, 0, 0, 0, 0, 2, 0, 0 ] ) ]
```

For getting an overview of the distribution of the elements of U to the conjugacy classes of G, we use the output of PermCharInfo (**Reference: PermCharInfo**).

```
——————————————————————— Example ———————————————————————
gap> infoperms:= PermCharInfo( tbl, perms );;
gap> Display( tbl, infoperms.display );
M22

      2    7 7 2 5 2 3
      3    2 1 2 . 1 .
      5    1 . . . . .
      7    1 . . . . .
     11    1 . . . . .

         1a 2a 3a 4a 6a 8a
     2P  1a 1a 3a 2a 3a 4a
     3P  1a 2a 1a 4a 2a 8a
     5P  1a 2a 3a 4a 6a 8a
     7P  1a 2a 3a 4a 6a 8a
    11P  1a 2a 3a 4a 6a 8a

I.1    3080 56  2 12  2  2
I.2       1 21  8 54 24 36
I.3       1  3  4  9 12 18
I.4    3080  8  2  8  2  4
I.5       1  3  8 36 24 72
I.6       1  3  4  9 12 18
I.7    3080 24 11  4  3  2
I.8       1  9 44 18 36 36
I.9       1  3  4  9 12 18
I.10   3080 24 20  4  .  2
I.11      1  9 80 18  . 36
I.12      1  3  4  9 12 18
```

We have four candidates. For each the above list shows first the character values, then the cardinality of the intersection of U with the classes, and then lower bounds for the lengths of U-conjugacy classes of these elements. Only those classes of G are shown that contain elements of U for at least one of the characters.

If the first two candidates are permutation characters corresponding to U then U contains exactly 8 elements of order 3 and thus U has a normal Sylow 3 subgroup P. But the order of $N_G(P)$ is bounded by 72, which can be shown as follows. The only elements in G with centralizer order divisible by 9 are of order 1 or 3, so P is self-centralizing in G. The factor $N_G(P)/C_G(P)$ is isomorphic with a subgroup of $\mathrm{Aut}(G) \cong GL(2,3)$ which has order divisible by 16, hence the order of $N_G(P)$ divides 144. Now note that $[G : N_G(P)] \equiv 1 \pmod 3$ by Sylow's Theorem, and $|G|/144 = 3\,080 \equiv -1 \pmod 3$. Thus the first two candidates are not permutation characters.

If the last two candidates are permutation characters corresponding to U then U has self-normalizing Sylow subgroups. This is because the index of a Sylow 2 normalizer in G is odd and divides 9, and if it is smaller than 9 then U contains at most $3 \cdot 15 + 1$ elements of 2 power order; the index of a Sylow 3 normalizer in G is congruent to 1 modulo 3 and divides 16, and if it is smaller than 16 then U contains at most $4 \cdot 8$ elements of order 3.

But since U is solvable and not a p-group, not all its Sylow subgroups can be self-normalizing; note that U has a proper normal subgroup N containing a Sylow p subgroup P of U for a prime divisor p of $|U|$, and $U = N \cdot N_U(P)$ holds by the Frattini argument (see [Hup67, Satz I.7.8]).

8.13   A Permutation Character of the Lyons group

Let G be a maximal subgroup with structure $3^{2+4} : 2A_5.D_8$ in the sporadic simple Lyons group Ly. We want to compute the permutation character 1_G^{Ly}. (This construction has been explained in [BP98, Section 4.2], without showing explicit GAP code.)

In the representation of Ly as automorphism group of the rank 5 graph B with $9\,606\,125$ points (see [CCN$^+$85, p. 174]), G is the stabilizer of an edge. A group S with structure $3.McL.2$ is the point stabilizer. So the two point stabilizer $U = S \cap G$ is a subgroup of index 2 in G. The index of U in S is $15\,400$, and according to the list of maximal subgroups of $McL.2$ (see [CCN$^+$85, p. 100]), the group U is isomorphic to the preimage in $3.McL.2$ of a subgroup H of $McL.2$ with structure $3^{1+4}_+ : 4S_5$.

Using the improved combinatorial method described in [BP98, Section 3.2], all possible permutation characters of degree $15\,400$ for the group McL are computed. (The method of [BP98, Section 3.3] is slower but also needs only a few seconds.)

```
———————————————— Example ————————————————
  gap> ly:= CharacterTable( "Ly" );;
  gap> mcl:= CharacterTable( "McL" );;
  gap> mcl2:= CharacterTable( "McL.2" );;
  gap> 3mcl2:= CharacterTable( "3.McL.2" );;
  gap> perms:= PermChars( mcl, rec( degree:= 15400 ) );
  [ Character( CharacterTable( "McL" ),
    [ 15400, 56, 91, 10, 12, 25, 0, 11, 2, 0, 0, 2, 1, 1, 1, 0, 0, 3,
        0, 0, 1, 1, 1, 1 ] ), Character( CharacterTable( "McL" ),
    [ 15400, 280, 10, 37, 20, 0, 5, 10, 1, 0, 0, 2, 1, 1, 0, 0, 0, 2,
        0, 0, 0, 0, 0, 0 ] ) ]
```

We get two characters, corresponding to the two classes of maximal subgroups of index $15\,400$ in McL. The permutation character $\pi = 1_{H \cap McL}^{McL}$ is the one with nonzero value on the class 10A, since the subgroup of structure $2S_5$ in $H \cap McL$ contains elements of order 10.

```
———————————————— Example ————————————————
  gap> ord10:= Filtered( [ 1 .. NrConjugacyClasses( mcl ) ],
  >                      i -> OrdersClassRepresentatives( mcl )[i] = 10 );
  [ 15 ]
  gap> List( perms, pi -> pi[ ord10[1] ] );
  [ 1, 0 ]
  gap> pi:= perms[1];
  Character( CharacterTable( "McL" ),
   [ 15400, 56, 91, 10, 12, 25, 0, 11, 2, 0, 0, 2, 1, 1, 1, 0, 0, 3, 0,
     0, 1, 1, 1, 1 ] )
```

The character $1_H^{McL.2}$ is an extension of π, so we can use the method of [BP98, Section 3.3] to compute all possible permutation characters for the group *McL*.2 that have the values of π on the classes of *McL*. We find that the extension of π to a permutation character of *McL*.2 is unique. Regarded as a character of 3.*McL*.2, this character is equal to 1_U^S.

```
                          ─── Example ───
  gap> map:= InverseMap( GetFusionMap( mcl, mcl2 ) );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, [ 10, 11 ], 12, [ 13, 14 ], 15, 16, 17,
    18, [ 19, 20 ], [ 21, 22 ], [ 23, 24 ] ]
  gap> torso:= CompositionMaps( pi, map );
  [ 15400, 56, 91, 10, 12, 25, 0, 11, 2, 0, 2, 1, 1, 0, 0, 3, 0, 1, 1 ]
  gap> perms:= PermChars( mcl2, rec( torso:= torso ) );
  [ Character( CharacterTable( "McL.2" ),
    [ 15400, 56, 91, 10, 12, 25, 0, 11, 2, 0, 2, 1, 1, 0, 0, 3, 0, 1,
        1, 110, 26, 2, 4, 0, 0, 5, 2, 1, 1, 0, 0, 1, 1 ] ) ]
  gap> pi:= Inflated( perms[1], 3mcl2 );
  Character( CharacterTable( "3.McL.2" ),
   [ 15400, 15400, 56, 56, 91, 91, 10, 12, 12, 25, 25, 0, 0, 11, 11, 2,
    2, 0, 0, 0, 2, 2, 1, 1, 1, 0, 0, 0, 0, 3, 3, 0, 0, 0, 1, 1, 1, 1,
    1, 1, 110, 26, 2, 4, 0, 0, 5, 2, 1, 1, 0, 0, 1, 1 ] )
```

The fusion of conjugacy classes of *S* in *Ly* can be computed from the character tables of *S* and *Ly* given in [CCN$^+$85], it is unique up to Galois automorphisms of the table of *Ly*.

```
                          ─── Example ───
  gap> fus:= PossibleClassFusions( 3mcl2, ly );;  Length( fus );
  4
  gap> g:= AutomorphismsOfTable( ly );;
  gap> OrbitLengths( g, fus, OnTuples );
  [ 4 ]
```

Now we can induce 1_U^S to *Ly*, which yields $(1_U^S)^{Ly} = 1_U^{Ly}$.

```
                          ─── Example ───
  gap> pi:= Induced( 3mcl2, ly, [ pi ], fus[1] )[1];
  Character( CharacterTable( "Ly" ),
   [ 147934325000, 286440, 1416800, 1082, 784, 12500, 0, 672, 42, 24,
    0, 40, 0, 2, 20, 0, 0, 0, 64, 10, 0, 50, 2, 0, 0, 4, 0, 0, 0, 0, 4,
    0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] )
```

All elements of odd order in *G* are contained in *U*, for such an element *g* we have

$$1_G^{Ly}(g) = |C_{Ly}(g)|/|G| \cdot |G \cap Cl_{Ly}(g)| = |C_{Ly}(g)|/(2 \cdot |U|) \cdot |U \cap Cl_{Ly}(g)| = 1/2 \cdot 1_U^{Ly}(g) \,,$$

so we can prescribe the values of 1_G^{Ly} on all classes of odd element order. For elements *g* of even order we have the weaker condition $U \cap Cl_{Ly}(g) \subseteq G \cap Cl_{Ly}(g)$ and thus $1_G^{Ly}(g) \geq 1/2 \cdot 1_U^{Ly}(g)$, which gives lower bounds for the value of 1_G^{Ly} on the remaining classes.

```
                          ─── Example ───
  gap> orders:= OrdersClassRepresentatives( ly );
  [ 1, 2, 3, 3, 4, 5, 5, 6, 6, 6, 7, 8, 8, 9, 10, 10, 11, 11, 12, 12,
    14, 15, 15, 15, 18, 20, 21, 21, 22, 22, 24, 24, 24, 25, 28, 30, 30,
    31, 31, 31, 31, 31, 33, 33, 37, 37, 40, 40, 42, 42, 67, 67, 67 ]
```

```
gap> torso:= [];;
gap> for i in [ 1 .. Length( orders ) ] do
>      if orders[i] mod 2 = 1 then
>        torso[i]:= pi[i]/2;
>      fi;
>    od;
gap> torso;
[ 73967162500,, 708400, 541,, 6250, 0,,,, 0,,, 1,,, 0, 0,,,, 25, 1, 0,
  ,, 0, 0,,,,,,, 0,,,, 0, 0, 0, 0, 0, 0, 0, 0, 0,,,, 0, 0, 0 ]
```

Exactly one possible permutation character of *Ly* satisfies these conditions.

```
─────────────────────── Example ───────────────────────
gap> perms:= PermChars( ly, rec( torso:= torso ) );;
gap> Length( perms );
43
gap> perms:= Filtered( perms, cand -> ForAll( [ 1 .. Length( orders ) ],
>        i -> cand[i] >= pi[i] / 2 ) );
[ Character( CharacterTable( "Ly" ),
  [ 73967162500, 204820, 708400, 541, 392, 6250, 0, 1456, 61, 25, 0,
      22, 10, 1, 10, 0, 0, 0, 32, 5, 0, 25, 1, 0, 1, 2, 0, 0, 0, 0,
      4, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0,
      0, 0 ] ) ]
```

(The permutation character 1_G^{Ly} was used in the proof that the character χ_{37} of *Ly* (see [CCN+85, p. 175]) occurs with multiplicity at least 2 in each character of *Ly* that is induced from a proper subgroup of *Ly*.)

8.14 Identifying two subgroups of $\mathrm{Aut}(U_3(5))$ (October 2001)

According to the Atlas of Finite Groups [CCN+85, p. 34], the group $\mathrm{Aut}(U_3(5))$ has two classes of maximal subgroups of order $2^4 \cdot 3^3$, which have the structures $3^2{:}2S_4$ and $6^2{:}D_{12}$, respectively.

```
─────────────────────── Example ───────────────────────
gap> tbl:= CharacterTable( "U3(5).3.2" );
CharacterTable( "U3(5).3.2" )
gap> deg:= Size( tbl ) / ( 2^4*3^3 );
1750
gap> pi:= PermChars( tbl, rec( torso:= [ deg ] ) );
[ Character( CharacterTable( "U3(5).3.2" ),
  [ 1750, 70, 13, 2, 0, 0, 1, 0, 0, 0, 10, 7, 10, 4, 2, 0, 0, 0, 0,
      0, 0, 30, 10, 3, 0, 0, 1, 0, 0 ] ),
  Character( CharacterTable( "U3(5).3.2" ),
  [ 1750, 30, 4, 6, 0, 0, 0, 0, 0, 0, 40, 7, 0, 6, 0, 0, 0, 0, 0, 0,
      0, 20, 0, 2, 2, 0, 0, 0, 0 ] ) ]
```

Now the question is which character belongs to which subgroup. We see that the first character vanishes on the classes of element order 8 and the second does not, so only the first one can be the permutation character induced from $6^2{:}D_{12}$.

```
─────────────────────────── Example ───────────────────────────
 gap> ord8:= Filtered( [ 1 .. NrConjugacyClasses( tbl ) ],
 >                 i -> OrdersClassRepresentatives( tbl )[i] = 8 );
 [ 9, 25 ]
 gap> List( pi, x -> x{ ord8 } );
 [ [ 0, 0 ], [ 0, 2 ] ]
```

Thus the question is whether the second candidate is really a permutation character. Since none of the two candidates vanishes on any outer coset of $U_3(5)$ in Aut($U_3(5)$), the point stabilizers are extensions of groups of order $2^3 \cdot 3^2$ in $U_3(5)$. The restrictions of the candidates to $U_3(5)$ are different, so we can try to answer the question using information about this group.

```
─────────────────────────── Example ───────────────────────────
 gap> subtbl:= CharacterTable( "U3(5)" );
 CharacterTable( "U3(5)" )
 gap> rest:= RestrictedClassFunctions( pi, subtbl );
 [ Character( CharacterTable( "U3(5)" ),
   [ 1750, 70, 13, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 ] ),
   Character( CharacterTable( "U3(5)" ),
   [ 1750, 30, 4, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
```

The intersection of the $3^2\!:\!2S_4$ subgroup with $U_3(5)$ lies inside the maximal subgroup of type M_{10}, which does not contain elements of order 6. Only the second character has this property.

```
─────────────────────────── Example ───────────────────────────
 gap> ord6:= Filtered( [ 1 .. NrConjugacyClasses( subtbl ) ],
 >                 i -> OrdersClassRepresentatives( subtbl )[i] = 6 );
 [ 9 ]
 gap> List( rest, x -> x{ ord6 } );
 [ [ 1 ], [ 0 ] ]
```

In order to establish the two characters as permutation characters, we could also compute the permutation characters of the degree in question directly from the table of marks of $U_3(5)$, which is contained in the GAP library of tables of marks.

```
─────────────────────────── Example ───────────────────────────
 gap> tom:= TableOfMarks( "U3(5)" );
 TableOfMarks( "U3(5)" )
 gap> perms:= PermCharsTom( subtbl, tom );;
 gap> Set( Filtered( perms, x -> x[1] = deg ) ) = Set( rest );
 true
```

We were mainly interested in the multiplicities of irreducible characters in these characters. The action of Aut($U_3(5)$) on the cosets of $3^2\!:\!2S_4$ turns out to be multiplicity-free whereas that on the cosets of $6^2\!:\!D_{12}$ is not.

```
─────────────────────────── Example ───────────────────────────
 gap> PermCharInfo( tbl, pi ).ATLAS;
 [ "1a+21a+42a+84aac+105a+125a+126a+250a+252a+288bc",
   "1a+42a+84ac+105ab+125a+126a+250a+252b+288bc" ]
```

It should be noted that the restrictions of the multiplicity-free character to the subgroups $U_3(5).2$ and $U_3(5).3$ of Aut($U_3(5)$) are not multiplicity-free.

```
                            Example
  gap> subtbl2:= CharacterTable( "U3(5).2" );;
  gap> rest2:= RestrictedClassFunctions( pi, subtbl2 );;
  gap> PermCharInfo( subtbl2, rest2 ).ATLAS;
  [ "1a+21aab+28aa+56aa+84a+105a+125aab+126aab+288aa",
    "1a+21ab+28a+56a+84a+105ab+125aab+126a+252a+288aa" ]
  gap> subtbl3:= CharacterTable( "U3(5).3" );;
  gap> rest3:= RestrictedClassFunctions( pi, subtbl3 );;
  gap> PermCharInfo( subtbl3, rest3 ).ATLAS;
  [ "1a+21abc+84aab+105a+125abc+126abc+144bcef",
    "1a+21bc+84ab+105aa+125abc+126adg+144bcef" ]
```

8.15   A Permutation Character of $\mathrm{Aut}(O_8^+(2))$ (October 2001)

According to the Atlas of Finite Groups [CCN+85, p. 85], the group $G = \mathrm{Aut}(O_8^+(2))$ has a class of maximal subgroups of order $2^{13} \cdot 3^2$, thus the index of these subgroups in G is $3^4 \cdot 5^2 \cdot 7$. The intersection of these subgroups with $H = O_8^+(2)$ lie inside maximal subgroups of type $2^6{:}A_8$. We want to show that the permutation character of the action of G on the cosets of these subgroups is not multiplicity-free.

Since the table of marks for H is available in GAP, but not that for G, we first compute the H-permutation characters of the intersections with H of index $3^4 \cdot 5^2 \cdot 7 = 14\,175$ subgroups in G.

(Note that these intersections have order $2^{12} \cdot 3$ because subgroups of order $2^{12} \cdot 3^2$ are contained in $O_8^+(2).2$ and hence are not maximal in G.)

```
                            Example
  gap> t:= CharacterTable( "O8+(2).3.2" );;
  gap> s:= CharacterTable( "O8+(2)" );;
  gap> tom:= TableOfMarks( s );;
  gap> perms:= PermCharsTom( s, tom );;
  gap> deg:= 3^4*5^2*7;
  14175
  gap> perms:= Filtered( perms, x -> x[1] = deg );;
  gap> Length( perms );
  4
  gap> Length( Set( perms ) );
  1
```

We see that there are four classes of subgroups S in H that may belong to maximal subgroups of the desired index in G, and that the permutation characters are equal. They lead to such groups if they extend to G, so we compute the possible permutation characters of G that extend these characters.

```
                            Example
  gap> fus:= PossibleClassFusions( s, t );
  [ [ 1, 2, 3, 3, 3, 4, 5, 5, 5, 6, 7, 8, 9, 10, 10, 10, 11, 12, 12,
        12, 13, 13, 13, 14, 14, 14, 15, 16, 16, 16, 17, 17, 17, 18, 19,
        20, 21, 22, 22, 22, 23, 23, 23, 24, 24, 24, 25, 26, 26, 26, 27,
        27, 27 ] ]
  gap> fus:= fus[1];;
  gap> inv:= InverseMap( fus );;
  gap> comp:= CompositionMaps( perms[1], inv );
  [ 14175, 1215, 375, 79, 0, 0, 27, 27, 99, 15, 7, 0, 0, 0, 0, 9, 3, 1,
```

```
   0, 1, 1, 0, 0, 0, 0, 0, 0 ]
gap> ext:= PermChars( t, rec( torso:= comp ) );
[ Character( CharacterTable( "O8+(2).3.2" ),
  [ 14175, 1215, 375, 79, 0, 0, 27, 27, 99, 15, 7, 0, 0, 0, 0, 9, 3,
      1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 63, 9, 15, 7, 1, 0, 3, 3, 3, 1,
      0, 0, 1, 1, 945, 129, 45, 69, 21, 25, 13, 0, 0, 0, 9, 0, 3, 3,
      7, 1, 0, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0 ] ) ]
gap> PermCharInfo( t, ext[1] ).ATLAS;
[ "1a+50b+100a+252bb+300b+700b+972bb+1400a+1944a+3200b+4032b" ]
```

Thus we get one permutation character of G which is not multiplicity-free.

8.16 Four Primitive Permutation Characters of the Monster Group

In this section, we compute four primitive permutation characters 1_H^M of the sporadic simple Monster group M, using the following strategy.

Let E be an elementary abelian 2-subgroup of M, and $H = N_M(E)$. For an involution $z \in E$, let $G = C_M(z)$ and $U = G \cap H = C_H(z)$ and $V = C_H(E)$, a normal subgroup of H. According to the Atlas of Finite Groups [CCN+85, p. 234], G has the structure $2.B$ if z is in the class 2A of M, and G has the structure $2_+^{1+24}.Co_1$ if z is in the class 2B of M. In the latter case, let N denote the extraspecial normal subgroup of order 2^{25} in G. It will turn out that in our situation, U contains N.

We want to compute many values of 1_H^M from the knowledge of permutation characters 1_X^M, for suitable subgroups X with the property $V \leq X \leq U$, and then use the GAP function PermChars (**Reference: PermChars**) for computing all those possible permutation characters of M that take the known values; if there is a unique solution then this is the desired character 1_H^M.

(In the year 2023, the character tables of three of the four maximal subgroups H in question became available, and the character table of the last one became available in 2025. Since then one can compute the permutation characters directly by first computing the possible class fusions to M and then inducing the trivial character of H to M. We get the same results this way, see below.)
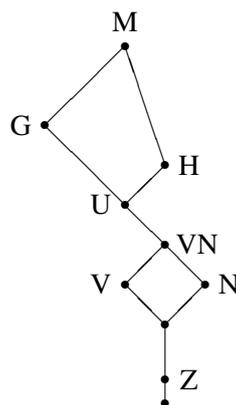
Why does this approach have a chance to be successful? Currently we do not have representations for the subgroups H in question, but the character tables of the involution centralizers G in M are available, and also either the character tables of X/V for the interesting subgroups X are known or we have enough information to compute the characters 1_X^G.

And how do we compute certain values of 1_H^M? Suppose that $\mathscr{C}$ is a union of classes of M and I is an index set such that $(1_H)_{\mathscr{C} \cap H} = (\sum_{i \in I} c_i 1_{X_i}^H)_{\mathscr{C} \cap H}$ holds for suitable rational numbers c_i.

The right hand side of this equality lives in H/V, provided that $\mathscr{C}$ "behaves well" w.r.t. factoring out the normal subgroup V of H, i. e., if there is a set of classes in H/V whose preimages in H form the set $H \cap \mathscr{C}$. For example, $\mathscr{C}$ may be the set of all those elements in M whose order is not divisible by a particular prime p that divides $|H|$ but not $|U|$.

Under these conditions, we have $(1_H^M)_{\mathscr{C}} = ((\sum_{i \in I} c_i 1_{X_i}^G)^M)_{\mathscr{C}}$, and we interpret the right hand side as follows: If X_i contains N then $1_{X_i}^G$ can be identified with $1_{X_i/N}^{G/N}$. If X_i contains at least Z then $1_{X_i}^G$ can be identified with $1_{X_i/Z}^{G/Z}$. As mentioned above, we have good chances to compute these characters. So the main task in each of the following sections is to find, for a suitable set $\mathscr{C}$ of classes, a linear combination of permutation characters of H/V whose restriction to $(\mathscr{C} \cap H)/V$ is constant and nonzero.

8.16.1 The Subgroup $2^2.2^{11}.2^{22}.(S_3 \times M_{24})$ (June 2009)

According to the Atlas of Finite Groups [CCN⁺85, p. 234], the Monster group M has a class of maximal subgroups H of the type $2^2.2^{11}.2^{22}.(S_3 \times M_{24})$. Currently the character table of H and the class fusion into M are not available in GAP. We are interested in the permutation character 1_H^G, and we will compute it without this information.

The subgroup H normalizes a Klein four group E whose involutions lie in the class 2B. We fix an involution z in E, and set $G = C_M(z)$, $U = C_H(z)$, and $V = C_H(E)$. Further, let N be the extraspecial normal subgroup of order 2^{25} in G.

So G has the structure $2_+^{1+24}.Co_1$, and U has index three in H. The order of NU/N is a multiple of $2^{2+11+22-25} \cdot 2 \cdot |M_{24}|$, and NU/N occurs as a subgroup of $G/N \cong Co_1$.

```
                        ─────── Example ───────
  gap> co1:= CharacterTable( "Co1" );;
  gap> order:= 2^(2+11+22-25) * 2 * Size( CharacterTable( "M24" ) );
  501397585920
  gap> maxes:= List( Maxes( co1 ), CharacterTable );;
  gap> filt:= Filtered( maxes, t -> Size( t ) mod order = 0 );
  [ CharacterTable( "2^11:M24" ) ]
  gap> List( filt, t -> Size( t ) / order );
  [ 1 ]
  gap> k:= filt[1];;
```

The list of maximal subgroups of Co_1 (see [CCN⁺85, p. 183]) tells us that NU/N is a maximal subgroup K of Co_1 and has the structure $2^{11} : M_{24}$. In particular, U contains N and thus $U/N \cong K$.

Let $\mathscr{C} = \{g \in M; 3 \nmid |g| \text{ or } 1_V^M(g^3) = 0\}$.

Then $(1_H)_{\mathscr{C} \cap H} = (1_U^H - 1/31_V^H)_{\mathscr{C} \cap H}$ holds, as we can see from computations with $H/V \cong S_3$, as follows.

```
                        ─────── Example ───────
  gap> f:= CharacterTable( "Symmetric", 3 );
  CharacterTable( "Sym(3)" )
  gap> OrdersClassRepresentatives( f );
  [ 1, 2, 3 ]
  gap> deg3:= PermChars( f, 3 );
  [ Character( CharacterTable( "Sym(3)" ), [ 3, 1, 0 ] ) ]
  gap> deg6:= PermChars( f, 6 );
```

```
  [ Character( CharacterTable( "Sym(3)" ), [ 6, 0, 0 ] ) ]
  gap> deg3[1] - 1/3 * deg6[1];
  ClassFunction( CharacterTable( "Sym(3)" ), [ 1, 1, 0 ] )
```

The character table of G is available in **GAP**, so we can compute the permutation character $\pi = 1_U^G$ by computing the primitive permutation character $1_K^{Co_1}$, identifying it with $1_{U/N}^{G/N}$, and then inflating this character to G.

```
──────────────── Example ────────────────
  gap> m:= CharacterTable( "M" );
  CharacterTable( "M" )
  gap> g:= CharacterTable( "MC2B" );
  CharacterTable( "2^1+24.Co1" )
  gap> pi:= RestrictedClassFunction( TrivialCharacter( k )^co1, g );;
```

Next we consider the permutation character $\phi = 1_V^G$. The group V does not contain N because K is perfect. But V contains Z because otherwise U would be a direct product of V and Z, which would imply that N would be a direct product of $V \cap N$ and Z. So we can regard ϕ as the inflation of $1_{V/Z}^{G/Z}$ from G/Z to G, i. e., we can perform the computations with the character table of the factor group G/Z.

```
──────────────── Example ────────────────
  gap> zclasses:= ClassPositionsOfCentre( g );;
  gap> gmodz:= g / zclasses;
  CharacterTable( "2^1+24.Co1/[ 1, 2 ]" )
  gap> invmap:= InverseMap( GetFusionMap( g, gmodz ) );;
  gap> pibar:= CompositionMaps( pi, invmap );;
```

Since $\phi(g) = [G:V] \cdot |g^G \cap V|/|g^G|$ holds for $g \in G$, and since $g^G \cap V \subseteq g^G \cap VN$, with equality if g has odd order, we get $\phi(g) = 2 \cdot \pi(g)$ if g has odd order, and $\phi(g) = 0$ if $\pi(g) = 0$.

We want to compute the possible permutation characters with these values.

```
──────────────── Example ────────────────
  gap> factorders:= OrdersClassRepresentatives( gmodz );;
  gap> phibar:= [];;
  gap> for i in [ 1 .. NrConjugacyClasses( gmodz ) ] do
  >      if factorders[i] mod 2 = 1 then
  >        phibar[i]:= 2 * pibar[i];
  >      elif pibar[i] = 0 then
  >        phibar[i]:= 0;
  >      fi;
  >    od;
  gap> cand:= PermChars( gmodz, rec( torso:= phibar ) );;
  gap> Length( cand );
  1
```

Now we know $\pi^M = 1_U^M$ and $\phi^M = 1_V^M$, so we can write down $(1_H^M)_{\mathscr{C}}$.

```
──────────────── Example ────────────────
  gap> phi:= RestrictedClassFunction( cand[1], g )^m;;
  gap> pi:= pi^m;;
  gap> cand:= ShallowCopy( pi - 1/3 * phi );;
```

```
gap> morders:= OrdersClassRepresentatives( m );;
gap> for i in [ 1 .. Length( morders ) ] do
>        if morders[i] mod 3 = 0 and phi[ PowerMap( m, 3 )[i] ] <> 0 then
>          Unbind( cand[i] );
>        fi;
>      od;
```

We claim that $1_H^M(g) \geq \pi^M(g) - 1/3\psi^M(g)$ for all $g \in M$. In order to see this, let H' denote the index two subgroup of H, and let $g \in M$. Since H is the disjoint union of V, $H' \setminus V$, and three H-conjugates of $U \setminus V$, we get

$$
\begin{aligned}
1_H^M(g) &= [M:H] \cdot |g^M \cap H|/|g^M| \\
&= [M:H] \cdot \left(|g^M \cap V| + 3|g^M \cap U \setminus V| + |g^M \cap H' \setminus V|\right)/|g^M| \\
&= [M:H] \cdot \left(3|g^M \cap U| - 2|g^M \cap V| + |g^M \cap H' \setminus V|\right)/|g^M| \\
&= 1_U^M(g) - 1/3 \cdot 1_V^G(g) + [M:H] \cdot |g^M \cap H' \setminus V|/|g^M|.
\end{aligned}
$$

Possible constituents of 1_H^M are those rational irreducible characters of M that are constituents of π^M.

```
——————————— Example ———————————
gap> constit:= Filtered( RationalizedMat( Irr( m ) ),
>                        chi -> ScalarProduct( m, chi, pi ) <> 0 );;
```

Now we compute the possible permutation characters that have the prescribed values, are compatible with the given lower bounds for values, and have only constituents in the given list.

```
——————————— Example ———————————
gap> cand:= PermChars( m,
>      rec( torso:= cand, chars:= constit,
>           lower:= ShallowCopy( pi - 1/3 * phi ),
>           normalsubgroup:= [ 1 .. NrConjugacyClasses( m ) ],
>           nonfaithful:= TrivialCharacter( m ) ) );
[ Character( CharacterTable( "M" ),
  [ 16009115629875684006343550944921875, 7774182899642733721875,
      1201685444413337875, 4436049512692980, 215448838605,
      131873639625, 760550656275, 110042727795, 943894035, 568854195,
      1851609375, 0, 4680311220, 405405, 78624756, 14467005, 178605,
      248265, 874650, 0, 76995, 591163, 224055, 34955, 29539, 20727,
      0, 0, 375375, 15775, 0, 0, 0, 495, 116532, 3645, 62316, 1017,
      11268, 357, 1701, 45, 117, 705, 0, 0, 4410, 1498, 0, 3780, 810,
      0, 0, 83, 135, 31, 0, 0, 0, 0, 0, 0, 0, 255, 195, 0, 215, 0, 0,
      210, 0, 42, 0, 35, 15, 1, 1, 160, 48, 9, 92, 25, 9, 9, 5, 1,
      21, 0, 0, 0, 0, 0, 98, 74, 42, 0, 0, 0, 120, 76, 10, 0, 0, 0,
      0, 0, 1, 1, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 3, 0,
      0, 0, 18, 0, 10, 0, 3, 3, 0, 1, 1, 1, 1, 0, 0, 2, 0, 0, 0, 0,
      0, 0, 2, 0, 0, 0, 0, 0, 6, 12, 0, 0, 2, 0, 0, 0, 2, 0, 0, 1, 1,
      0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0 ] ) ]
```

There is only one candidate, so we have found the permutation character.

The character table of H is available since 2023. We can compute the permutation character directly from this table.

```
─────────────────── Example ───────────────────
  gap> h:= CharacterTable( "2^(2+11+22).(M24xS3)" );;
  gap> fus:= PossibleClassFusions( h, m );;
  gap> cand = Set( fus, map -> InducedClassFunctionsByFusionMap( h, m,
  >                             [ TrivialCharacter( h ) ], map )[1] );
  true
```

8.16.2 The Subgroup $2^3.2^6.2^{12}.2^{18}.(L_3(2) \times 3.S_6)$ (September 2009)

According to the Atlas of Finite Groups [CCN+85, p. 234], the Monster group M has a class of maximal subgroups H of the type $2^3.2^6.2^{12}.2^{18}.(L_3(2) \times 3.S_6)$. Currently the character table of H and the class fusion into M are not available in GAP. We are interested in the permutation character 1_H^G, and we will compute it without this information.

The subgroup H normalizes an elementary abelian group E of order eight whose involutions lie in the class 2B. We fix an involution z in E, and set $G = C_M(z)$, $U = C_H(z)$, and $V = C_H(E)$. Further, let N be the extraspecial normal subgroup of order 2^{25} in G.

So G has the structure $2_+^{1+24}.Co_1$, and U has index seven in H. The order of NU/N is a multiple of $2^{3+6+12+18-25} \cdot |L_3(2)| \cdot |3.S_6|/7$, and NU/N occurs as a subgroup of $G/N \cong Co_1$.

```
─────────────────── Example ───────────────────
  gap> co1:= CharacterTable( "Co1" );;
  gap> order:= 2^(3+6+12+18-25) * 168 * 3 * Factorial( 6 ) / 7;
  849346560
  gap> maxes:= List( Maxes( co1 ), CharacterTable );;
  gap> filt:= Filtered( maxes, t -> Size( t ) mod order = 0 );
  [ CharacterTable( "2^(1+8)+.O8+(2)" ),
    CharacterTable( "2^(4+12).(S3x3S6)" ) ]
  gap> List( filt, t -> Size( t ) / order );
  [ 105, 1 ]
  gap> o8p2:= CharacterTable( "O8+(2)" );;
  gap> PermChars( o8p2, rec( torso:= [ 105 ] ) );
  [ ]
  gap> k:= filt[2];;
```

The list of maximal subgroups of Co_1 (see [CCN+85, p. 183]) tells us that NU/N is a maximal subgroup K of Co_1 and has the structure $2^{4+12}.(S_3 \times 3.S_6)$. (Note that the group $O_8^+(2)$ has no proper subgroup of index 105.) In particular, U contains N and thus $U/N \cong K$.

Let $\mathscr{C}$ be the set of elements in M whose order is not divisible by 7. Then $(1_H)_{\mathscr{C} \cap H} = (1_U^H - 1/31 1_{VN}^H + 1/21 1_V^H)_{\mathscr{C} \cap H}$ holds, as we can see from computations with $H/V \cong L_3(2)$, as follows.

So S4, V4, 1 suffice! -->

```
─────────────────── Example ───────────────────
  gap> f:= CharacterTable( "L3(2)" );
  CharacterTable( "L3(2)" )
  gap> OrdersClassRepresentatives( f );
  [ 1, 2, 3, 4, 7, 7 ]
  gap> deg7:= PermChars( f, 7 );
  [ Character( CharacterTable( "L3(2)" ), [ 7, 3, 1, 1, 0, 0 ] ) ]
  gap> deg42:= PermChars( f, 42 );
  [ Character( CharacterTable( "L3(2)" ), [ 42, 2, 0, 2, 0, 0 ] ),
    Character( CharacterTable( "L3(2)" ), [ 42, 6, 0, 0, 0, 0 ] ) ]
```

```
gap> deg168:= PermChars( f, 168 );
[ Character( CharacterTable( "L3(2)" ), [ 168, 0, 0, 0, 0, 0 ] ) ]
gap> deg7[1] - 1/3 * deg42[2] + 1/21 * deg168[1];
ClassFunction( CharacterTable( "L3(2)" ), [ 1, 1, 1, 1, 0, 0 ] )
```

(Note that VN/V is a Klein four group, and there is only one transitive permutation character of $L_3(2)$ that is induced from such subgroups.)

The character table of G is available in GAP, so we can compute the permutation character $\pi = 1_U^G$ by computing the primitive permutation character $1_K^{Co_1}$, identifying it with $1_{U/N}^{G/N}$, and then inflating this character to G.

```
─────────────────── Example ───────────────────
gap> m:= CharacterTable( "M" );
CharacterTable( "M" )
gap> g:= CharacterTable( "MC2B" );
CharacterTable( "2^1+24.Co1" )
gap> pi:= RestrictedClassFunction( TrivialCharacter( k )^co1, g );;
```

The permutation character $\psi = 1_{VN}^G$ can be computed as the inflation of $1_{VN/N}^{G/N} = (1_{VN/N}^{U/N})^{G/N}$, where $1_{VN/N}^{U/N}$ is a character of K that can be identified with the regular permutation character of $U/VN \cong S_3$.

```
─────────────────── Example ───────────────────
gap> nsg:= ClassPositionsOfNormalSubgroups( k );;
gap> nsgsizes:= List( nsg, x -> Sum( SizesConjugacyClasses( k ){ x } ) );;
gap> nn:= nsg[ Position( nsgsizes, Size( k ) / 6 ) ];;
gap> psi:= 0 * [ 1 .. NrConjugacyClasses( k ) ];;
gap> for i in nn do
>        psi[i]:= 6;
>    od;
gap> psi:= InducedClassFunction( k, psi, co1 );;
gap> psi:= RestrictedClassFunction( psi, g );;
```

Next we consider the permutation character $\phi = 1_V^G$. The group V does not contain N because K does not have a factor group of the type S_4. But V contains Z because U/V is centerless. So we can regard ϕ as the inflation of $1_{V/Z}^{G/Z}$ from G/Z to G, i. e., we can perform the computations with the character table of the factor group G/Z.

```
─────────────────── Example ───────────────────
gap> zclasses:= ClassPositionsOfCentre( g );;
gap> gmodz:= g / zclasses;
CharacterTable( "2^1+24.Co1/[ 1, 2 ]" )
gap> invmap:= InverseMap( GetFusionMap( g, gmodz ) );;
gap> psibar:= CompositionMaps( psi, invmap );;
```

Since $\phi(g) = [G:V] \cdot |g^G \cap V|/|g^G|$ holds for $g \in G$, and since $g^G \cap V \subseteq g^G \cap VN$, with equality if g has odd order, we get $\phi(g) = 4 \cdot \psi(g)$ if g has odd order, and $\phi(g) = 0$ if $\psi(g) = 0$.

We want to compute the possible permutation characters with these values. This is easier if we "go down" from VN to V in two steps.

```
————————————————— Example ——————————————————
gap> factorders:= OrdersClassRepresentatives( gmodz );;
gap> phibar:= [];;
gap> upperphibar:= [];;
gap> for i in [ 1 .. NrConjugacyClasses( gmodz ) ] do
>        if factorders[i] mod 2 = 1 then
>          phibar[i]:= 2 * psibar[i];
>        elif psibar[i] = 0 then
>          phibar[i]:= 0;
>        fi;
>        upperphibar[i]:= 2 * psibar[i];
>     od;
gap> cand:= PermChars( gmodz, rec( torso:= phibar,
>             upper:= upperphibar,
>             normalsubgroup:= [ 1 .. NrConjugacyClasses( gmodz ) ],
>             nonfaithful:= TrivialCharacter( gmodz ) ) );;
gap> Length( cand );
3
```

One of the candidates computed in this first step is excluded by the fact that it is induced from a subgroup that contains N/Z.

```
————————————————— Example ——————————————————
gap> nn:= First( ClassPositionsOfNormalSubgroups( gmodz ),
>             x -> Sum( SizesConjugacyClasses( gmodz ){x} ) = 2^24 );
[ 1 .. 4 ]
gap> cont:= PermCharInfo( gmodz, cand ).contained;;
gap> cand:= cand{ Filtered( [ 1 .. Length( cand ) ],
>                   i -> Sum( cont[i]{ nn } ) < 2^24 ) };;
gap> Length( cand );
2
```

Now we run the second step. After excluding the candidates that cannot be induced from subgroups whose intersection with N/Z has index four in N/Z, we get four solutions.

```
————————————————— Example ——————————————————
gap> poss:= [];;
gap> for v in cand do
>        phibar:= [];
>        upperphibar:= [];
>        for i in [ 1 .. NrConjugacyClasses( gmodz ) ] do
>          if factorders[i] mod 2 = 1 then
>            phibar[i]:= 2 * v[i];
>          elif v[i] = 0 then
>            phibar[i]:= 0;
>          fi;
>          upperphibar[i]:= 2 * v[i];
>        od;
>        Append( poss, PermChars( gmodz, rec( torso:= phibar,
>                      upper:= upperphibar,
>                      normalsubgroup:= [ 1 .. NrConjugacyClasses( gmodz ) ],
>                      nonfaithful:= TrivialCharacter( gmodz ) ) ) );
>     od;
```

```
gap> Length( poss );
6
gap> cont:= PermCharInfo( gmodz, poss ).contained;;
gap> poss:= poss{ Filtered( [ 1 .. Length( poss ) ],
>                            i -> Sum( cont[i]{ nn } ) < 2^23 ) };;
gap> Length( poss );
4
gap> phicand:= RestrictedClassFunctions( poss, g );;
```

Since we have several candidates for 1_V^G, we form the linear combinations for all these candidates.

```
———————————————— Example ————————————————
gap> phicand:= RestrictedClassFunctions( poss, g );;
gap> phicand:= InducedClassFunctions( phicand, m );;
gap> psi:= psi^m;;
gap> pi:= pi^m;;
gap> cand:= List( phicand,
>           phi -> ShallowCopy( pi - 1/3 * psi + 1/21 * phi ) );;
gap> morders:= OrdersClassRepresentatives( m );;
gap> for x in cand do
>       for i in [ 1 .. Length( morders ) ] do
>         if morders[i] mod 7 = 0 then
>           Unbind( x[i] );
>         fi;
>       od;
>     od;
```

Exactly one of the candidates has only integral values.

```
———————————————— Example ————————————————
gap> cand:= Filtered( cand, x -> ForAll( x, IsInt ) );
[ [ 4050306254358548053604918389065234375, 148844831270071996434375,
    2815847622206994375, 14567365753025085, 3447181417680,
    659368198125, 3520153823175, 548464353255, 5706077895,
    3056566695, 264515625, 0, 19572895485, 6486480, 186109245,
    61410960, 758160, 688365,,, 172503, 1264351, 376155, 137935,
    99127, 52731, 0, 0, 119625, 3625, 0, 0, 0, 0, 402813, 29160,
    185301, 2781, 21069, 1932, 4212, 360, 576, 1125, 0, 0,,,, 2160,
    810, 0, 0, 111, 179, 43, 0, 0, 0, 0, 0, 0, 0, 185, 105, 0, 65,
    0, 0,,,,, 0, 0, 0, 0, 337, 105, 36, 157, 37, 18, 18, 16, 4, 21,
    0, 0, 0, 0, 0,,,,, 0, 0, 60, 40, 10, 0, 0, 0, 0, 0, 1, 1, 0, 0,
    0,,, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 1, 0, 0, 0,,,,, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,,,, 0, 0, 0, 6, 8, 0, 0, 2,
    0, 0, 0, 0, 0, 0, 0, 0,,, 0, 0, 0, 0, 0,,,, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0,, 0 ] ]
```

Possible constituents of 1_H^M are those rational irreducible characters of M that are constituents of π^M.

```
———————————————— Example ————————————————
gap> constit:= Filtered( RationalizedMat( Irr( m ) ),
>                        chi -> ScalarProduct( m, chi, pi ) <> 0 );;
```

Now we compute the possible permutation characters that have the prescribed values and have only constituents in the given list.

```
_____ Example _____
gap> cand:= PermChars( m, rec( torso:= cand[1], chars:= constit ) );
[ Character( CharacterTable( "M" ),
  [ 40503062543585480536049183890652343375, 14884483127007199643435375,
      2815847622206994375, 14567365753025085, 3447181417680,
      659368198125, 3520153823175, 548464353255, 5706077895,
      3056566695, 264515625, 0, 19572895485, 6486480, 186109245,
      61410960, 758160, 688365, 58310, 0, 172503, 1264351, 376155,
      137935, 99127, 52731, 0, 0, 119625, 3625, 0, 0, 0, 0, 402813,
      29160, 185301, 2781, 21069, 1932, 4212, 360, 576, 1125, 0, 0,
      1302, 294, 0, 2160, 810, 0, 0, 111, 179, 43, 0, 0, 0, 0, 0, 0,
      0, 185, 105, 0, 65, 0, 0, 224, 0, 14, 0, 0, 0, 0, 0, 337, 105,
      36, 157, 37, 18, 18, 16, 4, 21, 0, 0, 0, 0, 0, 70, 38, 14, 0,
      0, 0, 60, 40, 10, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 10, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 5, 1, 0, 0, 0, 24, 0, 6, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 6, 8, 0, 0, 2,
      0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 4, 0, 2, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0 ] ) ]
```

There is only one candidate, so we have found the permutation character.

The character table of H is available since 2023. We can compute the permutation character directly from this table. (The class fusion from H to M is unique up to table automorphisms, but its computation is a bit tricky, thus we do not compute this fusion here.)

```
_____ Example _____
gap> h:= CharacterTable( "2^[39].(L3(2)x3.S6)" );;
gap> cand[1] = TrivialCharacter( h )^m;
true
```

8.16.3 The Subgroup $2^5.2^{10}.2^{20}.(S_3 \times L_5(2))$ (October 2009)

According to the Atlas of Finite Groups [CCN+85, p. 234], the Monster group M has a class of maximal subgroups H of the type $2^5.2^{10}.2^{20}.(S_3 \times L_5(2))$. Currently the character table of H and the class fusion into M are not available in GAP. We are interested in the permutation character 1_H^G, and we will compute it without this information.

The subgroup H normalizes an elementary abelian group E of order 32 whose involutions lie in the class 2B. We fix an involution z in E, and set $G = C_M(z)$, $U = C_H(z)$, and $V = C_H(E)$. Further, let N be the extraspecial normal subgroup of order 2^{25} in G.

So G has the structure $2_+^{1+24}.Co_1$, and U has index 31 in H. The order of NU/N is a multiple of $2^{5+10+20-25} \cdot |L_5(2)| \cdot |S_3|/31$, and NU/N occurs as a subgroup of $G/N \cong Co_1$.

```
_____ Example _____
gap> co1:= CharacterTable( "Co1" );;
gap> order:= 2^35*Size( CharacterTable( "L5(2)" ) )*6 / 2^25 / 31;
1981808640
gap> maxes:= List( Maxes( co1 ), CharacterTable );;
gap> filt:= Filtered( maxes, t -> Size( t ) mod order = 0 );
[ CharacterTable( "2^11:M24" ), CharacterTable( "2^(1+8)+.O8+(2)" ),
  CharacterTable( "2^(2+12):(A8xS3)" ) ]
```

```
gap> List( filt, t -> Size( t ) / order );
[ 253, 45, 1 ]
gap> m24:= CharacterTable( "M24" );;
gap> cand:= PermChars( m24, rec( torso:=[ 253 ] ) );
[ Character( CharacterTable( "M24" ),
  [ 253, 29, 13, 10, 1, 5, 5, 1, 3, 2, 1, 1, 1, 1, 3, 0, 2, 1, 1, 1,
      0, 0, 1, 1, 0, 0 ] ) ]
gap> TestPerm5( m24, cand, m24 mod 11 );
[  ]
gap> PermChars( CharacterTable( "O8+(2)" ), rec( torso:=[ 45 ] ) );
[  ]
gap> k:= filt[3];;
```

The list of maximal subgroups of Co_1 (see [CCN+85, p. 183]) tells us that NU/N is a maximal subgroup K of Co_1 and has the structure $2^{2+12}.(A_8 \times S_3)$. (Note that the group M_{24} has no proper subgroup of index 253, which is shown above using the 11-modular Brauer table of M_{24}. Furthermore, the group $O_8^+(2)$ has no subgroup of index 45.) In particular, U contains N and thus $U/N \cong K$.

Let $\mathscr{C}$ be the set of elements in M whose order is not divisible by 31 or 21. We want to find an index set I and subgroups X_i, for $i \in I$, with the property that $V \leq X_i \leq U$ and

$$(1_H)_{\mathscr{C} \cap H} = \left(\sum_{i \in I} c_i 1_{X_i}^H \right)_{\mathscr{C} \cap H}$$

holds for suitable rational integers c_i. Let W be the full preimage of the elementary normal subgroup of order 16 in $U/V \cong 2^4.A_8$ under the natural epimorphism from U to U/V, and set $I_1 = \{i \in I; W \leq X_i\}$ and $I_2 = I \setminus I_1$.

Using the known table of marks of U/V, we will find a solution such that $[W : (W \cap X_i)] = 2$ for all $i \in I_2$. First we compute the permutation characters $1_S^{U/V}$ for all subgroups S of U/V that contain W/V, and induce them to H/V.

```
─────── Example ───────
gap> subtbl:= CharacterTable( "2^4:A8" );;
gap> subtom:= TableOfMarks( subtbl );;
gap> perms:= PermCharsTom( subtbl, subtom );;
gap> nsg:= ClassPositionsOfNormalSubgroups( subtbl );
[ [ 1 ], [ 1, 2 ], [ 1 .. 25 ] ]
gap> above:= Filtered( perms, x -> x[1] = x[2] );;
gap> tbl:= CharacterTable( "L5(2)" );;
gap> above:= Set( Induced( subtbl, tbl, above ) );;
```

Next we compute the permutation characters $1_S^{U/V}$ for all subgroups S of U/V whose intersection with W/V has index two in W/V. Afterwards we exclude certain subgroups that would slow down later computations, and induce also these characters to H/V.

```
─────── Example ───────
gap> index2:= Filtered( perms,
>       x -> Sum( PermCharInfo( subtbl, [x] ).contained[1]{ [1,2] } ) = 8 );;
gap> index2:= Filtered( index2, x -> not x[1] in [ 630, 840, 1260, 1680 ] );;
gap> index2:= Set( Induced( subtbl, tbl, index2 ) );;
```

Now we induce the permutation characters to H/V, and compute the coefficients of a linear combination as desired.

```
                        ──── Example ────
  gap> orders:= OrdersClassRepresentatives( tbl );;
  gap> goodclasses:= Filtered( [ 1 .. NrConjugacyClasses( tbl ) ],
  >                             i -> not orders[i] in [ 21, 31 ] );
  [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 ]
  gap> matrix:= List( Concatenation( above, index2 ), x -> x{ goodclasses } );;
  gap> sol:= SolutionMat( matrix,
  >               ListWithIdenticalEntries( Length( goodclasses ), 1 ) );
  [ 692/651, 57/217, -78/217, -26/217, 0, 74/651, 11/217, 0, 3/217,
    151/651, 0, 22/651, 0, 0, 0, -11/217, 0, 0, 0, 0, 0, 0, 0, 0,
    -115/651, 0, -3/31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -34/93,
    -11/651, 0, 2/21, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1/31, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
  gap> nonzero:= Filtered( [ 1 .. Length( sol ) ], i -> sol[i] <> 0 );
  [ 1, 2, 3, 4, 6, 7, 9, 10, 12, 16, 25, 27, 106, 107, 109, 120 ]
  gap> sol:= sol{ nonzero };;
```

Now we transfer this linear combination to the character tables which are given in our situation.

Those constituents that are induced from subgroups of H above W can be identified uniquely via their degrees and their values distribution; we compute these characters in the character table of U/W obtained as a factor table of the character table of U/N, lift them back to U/N, induce them to G/N, inflate them to G, and then induce them fo M.

```
                        ──── Example ────
  gap> a8degrees:= List( above{ Filtered( nonzero,
  >                                 x -> x <= Length( above ) ) },
  >                    x -> x[1] ) / 31;
  [ 1, 8, 15, 28, 56, 56, 70, 105, 120, 168, 336, 336 ]
  gap> a8tbl:= subtbl / [ 1, 2 ];;
  gap> invtoa8:= InverseMap( GetFusionMap( subtbl, a8tbl ) );;
  gap> nsg:= ClassPositionsOfNormalSubgroups( k );;
  gap> nn:= First( nsg, x -> Sum( SizesConjugacyClasses( k ){ x } ) = 6*2^14 );;
  gap> a8tbl_other:= k / nn;;
  gap> g:= CharacterTable( "MC2B" );
  CharacterTable( "2^1+24.Co1" )
  gap> constit:= [];;
  gap> for i in [ 1 .. Length( a8degrees ) ] do
  >       cand:= PermChars( a8tbl_other, rec( torso:= [ a8degrees[i] ] ) );
  >       filt:= Filtered( perms, x -> x^tbl = above[ nonzero[i] ] );
  >       filt:= List( filt, x -> CompositionMaps( x, invtoa8 ) );
  >       cand:= Filtered( cand,
  >               x -> ForAny( filt, y -> Collected( x ) = Collected(y) ) );
  >       Add( constit, List( Induced( Restricted( Induced(
```

```
>             Restricted( cand, k ), co1 ), g ), m ), ValuesOfClassFunction ) );
>      od;
gap> List( constit, Length );
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

Dealing with the remaining constituents is more involved. For a permutation character $1_{X/V}^{U/V}$, we compute $1_{WX/V}^{U/V}$, a character whose degree is half as large and which can be regarded as a character of U/W. This character can be treated like the ones above: We lift it to U/N, induce it to G/N, and inflate it to $G/Z(G)$; let this character be $1_Y^{G/Z(G)}$, for some subgroup Y. Then we compute the possible permutation characters of $G/Z(G)$ that can be induced from a subgroup of index two inside Y, inflate these characters to G and then induce them to M.

```
─────────────────────── Example ───────────────────────
gap> downdegrees:= List( index2{ Filtered( nonzero,
>                                 x -> x > Length( above ) )
>                               - Length( above ) },
>                        x -> x[1] ) / 31;
[ 30, 210, 210, 1920 ]
gap> f:= g / ClassPositionsOfCentre( g );;
gap> forders:= OrdersClassRepresentatives( f );;
gap> inv:= InverseMap( GetFusionMap( g, f ) );;
gap> for j in [ 1 .. Length( downdegrees ) ] do
>      chars:= [];
>      cand:= PermChars( a8tbl_other, rec( torso:= [ downdegrees[j]/2 ] ) );
>      filt:= Filtered( perms, x -> x^tbl = index2[ nonzero[
>                 j + Length( a8degrees ) ] - Length( above ) ] );
>      filt:= Induced( subtbl, a8tbl, filt,
>                 GetFusionMap( subtbl, a8tbl ));
>      cand:= Filtered( cand, x -> ForAny( filt,
>               y -> Collected( x ) = Collected( y ) ) );
>      cand:= Restricted( Induced( Restricted( cand, k ), co1 ), g );
>      for chi in cand do
>        cchi:= CompositionMaps( chi, inv );
>        upper:= [];
>        pphi:= [];
>        for i in [ 1 .. NrConjugacyClasses( f ) ] do
>          if forders[i] mod 2 = 1 then
>            pphi[i]:= 2 * cchi[i];
>          elif cchi[i] = 0 then
>            pphi[i]:= 0;
>          fi;
>          upper[i]:= 2* cchi[i];
>        od;
>        Append( chars, PermChars( f, rec( torso:= ShallowCopy( pphi ),
>            upper:= upper,
>            normalsubgroup:= [ 1 .. 4 ],
>            nonfaithful:= cchi ) ) );
>      od;
>      Add( constit, List( Induced( Restricted( chars, g ), m ),
>                       ValuesOfClassFunction ) );
>    od;
```

```
gap> List( constit, Length );
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 10, 10, 2 ]
```

Now we form the possible linear combinations.

```
_____ Example _____
gap> cand:= List( Cartesian( constit ), l -> sol * l );;
gap> m:= CharacterTable( "M" );
CharacterTable( "M" )
gap> morders:= OrdersClassRepresentatives( m );;
gap> for x in cand do
>      for i in [ 1 .. Length( morders ) ] do
>        if morders[i] mod 31 = 0 or morders[i] mod 21 = 0 then
>          Unbind( x[i] );
>        fi;
>      od;
>    od;
```

Exactly one of the candidates has only integral values.

```
_____ Example _____
gap> cand:= Filtered( cand, x -> ForAll( x, IsInt ) );
[ [ 3919651213895369084133791989417968750, 2391448729295137699687500,
      4741631380424688750, 95004559258859250, 6463465158150,
      3343634862750, 9541617648750, 1473391032750, 14813923950,
      13132815150, 0, 8203125, 98278859250, 12162150, 915563250, 93887910,
      1159110, 5873310, 8746500, 0, 795150, 5819550, 3363750, 1043710,
      623310, 368550, 0, 0, 0, 0, 281250, 5250, 11250, 0, 1883250, 167670,
      889650, 24030, 94770, 11550, 8910, 2070, 3510, 6270, 0, 0, 44100, 14980,
      0, 0, 0, 300, 1500, 910, 1510, 310, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1250,
      0, 50, 50,,,,,, 0, 0, 0, 0, 1410, 450, 270, 610, 270, 90, 90, 70, 30, 150,
      0, 0, 0, 0, 0, 980, 740, 420, 0, 0, 300, 0, 0, 0, 60, 60, 60,,, 10, 10,
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 10, 0,,,,,, 0,
      0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 20, 0, 0, 0, 0, 0,
      0, 0, 20, 20, 0, 20,,, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,,,, 0,
      0, 0, 0, 0, 0,,, 0, 0, 0, 0, 0, 0,, 0, 0, 0 ] ]
```

Now we compute the possible permutation characters that have the prescribed values.

```
_____ Example _____
gap> cand:= PermChars( m, rec( torso:= cand[1] ) );
[ Character( CharacterTable( "M" ),
  [ 3919651213895369084133791989417968750, 2391448729295137699687500,
      4741631380424688750, 95004559258859250, 6463465158150,
      3343634862750, 9541617648750, 1473391032750, 14813923950,
      13132815150, 0, 8203125, 98278859250, 12162150, 915563250, 93887910,
      1159110, 5873310, 8746500, 0, 795150, 5819550, 3363750, 1043710,
      623310, 368550, 0, 0, 0, 0, 281250, 5250, 11250, 0, 1883250, 167670,
      889650, 24030, 94770, 11550, 8910, 2070, 3510, 6270, 0, 0, 44100, 14980,
      0, 0, 0, 300, 1500, 910, 1510, 310, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1250,
      0, 50, 50, 210, 0, 42, 0, 0, 0, 0, 0, 1410, 450, 270, 610, 270, 90, 90,
      70, 30, 150, 0, 0, 0, 0, 0, 980, 740, 420, 0, 0, 300, 0, 0, 0, 60, 60,
      60, 30, 30, 10, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
         1, 1, 0, 18, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
         0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 3, 3, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
```

There is only one candidate, so we have found the permutation character.

The character table of *H* is available since 2025. We can compute the permutation character directly from this table.

```
                              ─── Example ───
  gap> h:= CharacterTable( "2^(5+10+20).(S3xL5(2))" );;
  gap> fus:= PossibleClassFusions( h, m );;
  gap> cand = Set( fus, map -> InducedClassFunctionsByFusionMap( h, m,
  >                            [ TrivialCharacter( h ) ], map )[1] );
  true
```

8.16.4 The Subgroup $2^{10+16}.O_{10}^+(2)$ (November 2009)

According to the Atlas of Finite Groups [CCN+85, p. 234], the Monster group *M* has a class of maximal subgroups *H* of the type $2^{10+16}.O_{10}^+(2)$. Currently the character table of *H* and the class fusion into *M* are not available in GAP. We are interested in the permutation character 1_H^M, and we will compute it without this information.

The subgroup *H* normalizes an elementary abelian group *E* of order 2^{10} which contains 496 involutions in the class 2A and 527 involutions in the class 2B. Let *V* denote the normal subgroup of order 2^{26} in *H*, and set $\bar{H} = H/N$. Since the smallest two indices of maximal subgroups of $\bar{H}$ are 496 and 527, respectively, *H* acts transitively on both the 2A and the 2B involutions in *E*, and the centralizers of these involutions contain *V*.

```
                              ─── Example ───
  gap> Hbar:= CharacterTable( "O10+(2)" );;
  gap> U_Abar:= CharacterTable( "O10+(2)M1" );
  CharacterTable( "S8(2)" )
  gap> Index( Hbar, U_Abar );
  496
  gap> U_Bbar:= CharacterTable( "O10+(2)M2" );
  CharacterTable( "2^8:O8+(2)" )
  gap> Index( Hbar, U_Bbar );
  527
```

We fix a 2A involution z_A in *E*, and set $G_A = C_M(z_A)$ and $U_A = C_H(z_A)$. So G_A has the structure 2.*B* and U_A has the structure $2^{10+16}.S_8(2)$. From the list of maximal subgroups of *B* we see that the image of G_A under the natural epimorphism from G_A to *B* is a maximal subgroup of *B* and has the structure $2^{9+16}.S_8(2)$.

```
                              ─── Example ───
  gap> b:= CharacterTable( "B" );
  CharacterTable( "B" )
  gap> Horder:= 2^26 * Size( Hbar );
  15770110055923770163200
  gap> order:= Horder / ( 2 * 496 );
  15897288870199296600
  gap> maxes:= List( Maxes( b ), CharacterTable );;
```

```
gap> filt:= Filtered( maxes, t -> Size( t ) mod order = 0 );
[ CharacterTable( "2^(9+16).S8(2)" ) ]
gap> List( filt, t -> Size( t ) / order );
[ 1 ]
gap> u1:= filt[1];
CharacterTable( "2^(9+16).S8(2)" )
```

Analogously, we fix a 2B involution z_B in E, and set $G_B = C_M(z_B)$ and $U_B = C_H(z_B)$, Further, let N be the extraspecial normal subgroup of order 2^{25} in G_B. So G_B has the structure $2^{1+24}_+.Co_1$, and U_B has index 527 in G_B. From the list of maximal subgroups of Co_1 we see that the image of U_B under the natural epimorphism from G_B to Co_1 is a maximal subgroup of Co_1 and has the structure $2^{1+8}_+.O_8^+(2)$.

```
——————————————————————————— Example ———————————————————————————
gap> co1:= CharacterTable( "Co1" );;
gap> order:= Horder / ( 2^25 * 527 );
89181388800
gap> maxes:= List( Maxes( co1 ), CharacterTable );;
gap> filt:= Filtered( maxes, t -> Size( t ) mod order = 0 );
[ CharacterTable( "2^(1+8)+.O8+(2)" ) ]
gap> List( filt, t -> Size( t ) / order );
[ 1 ]
gap> u2:= filt[1];
CharacterTable( "2^(1+8)+.O8+(2)" )
```

First we compute the permutation characters $\pi_A = 1_{U_A}^M$ and $\pi_B = 1_{U_B}^M$.

```
——————————————————————————— Example ———————————————————————————
gap> m:= CharacterTable( "M" );
CharacterTable( "M" )
gap> 2b:= CharacterTable( "MC2A" );
CharacterTable( "2.B" )
gap> mm:= CharacterTable( "MC2B" );
CharacterTable( "2^1+24.Co1" )
gap> pi_A:= RestrictedClassFunction( TrivialCharacter( u1 )^b, 2b )^m;;
gap> pi_B:= RestrictedClassFunction( TrivialCharacter( u2 )^co1, mm )^m;;
```

The degree of 1_H^M is of course known.

```
——————————————————————————— Example ———————————————————————————
gap> torso:= [ Size( m ) / Horder ];
[ 512372707698741056749515292734375 ]
```

Next we compute some zero values of 1_H^M, using the following conditions.

- For $g \in M$, if $|g|$ does not divide $|H|$ or if $|g|$ is not the product of an element order in H/V and a 2-power. (In fact we could use that the exponent of V is 4, but this would not improve the result.)

- Let $U \le H \le G$, and let p be a prime that does not divide $[H : U]$. Then U contains a Sylow p subgroup of H, so each element of order p in H is conjugate in H to an element in U. For $g \in G$, $g = g_p h$, where the order of g_p is a power of p such that $1_U^G(g_p) = 0$ holds, we have $1_H^G(g) = 0$. We apply this to $U \in \{U_A, U_B\}$.

```
                          ──────── Example ────────
  gap> morders:= OrdersClassRepresentatives( m );;
  gap> 2parts:= Union( [ 1 ], Filtered( Set( morders ),
  >                            x -> IsPrimePowerInt( x ) and IsEvenInt( x ) ) );
  [ 1, 2, 4, 8, 16, 32 ]
  gap> factorders:= Set( OrdersClassRepresentatives( Hbar ) );;
  gap> primes_A:= Filtered( PrimeDivisors( Horder ), p -> 496 mod p <> 0 );
  [ 3, 5, 7, 17 ]
  gap> primes_B:= Filtered( PrimeDivisors( Horder ), p -> 527 mod p <> 0 );
  [ 2, 3, 5, 7 ]
  gap> primes:= Union( primes_A, primes_B );;
  gap> n:= NrConjugacyClasses( m );;
  gap> for i in [ 1 .. n ] do
  >    if Horder mod morders[i] <> 0 then
  >      torso[i]:= 0;
  >    elif ForAll( factorders, x -> not morders[i] / x in 2parts ) then
  >      torso[i]:= 0;
  >    else
  >      for p in primes do
  >        if morders[i] mod p = 0 then
  >          pprime:= morders[i];
  >          while pprime mod p = 0 do pprime:= pprime / p; od;
  >          pos:= PowerMap( m, pprime )[i];
  >          if p in primes_A and pi_A[ pos ] = 0 then
  >            torso[i]:= 0;
  >          elif p in primes_B and pi_B[ pos ] = 0 then
  >            torso[i]:= 0;
  >          fi;
  >        fi;
  >      od;
  >    fi;
  > od;
  gap> torso;
  [ 51237270769874105674951529273437 5,,,,, 0,,,,,,,,,,,,,, 0,, 0,,,,,,,,,,,
    ,,,, 0,,,, 0,,,,,, 0, 0, 0,,, 0,,,, 0,,,,,,,,,,, 0,,,,,,,, 0, 0, 0,
    0, 0, 0, 0,,,,, 0,,,,, 0, 0, 0, 0, 0, 0,,,, 0, 0,,,,, 0,,,,,,, 0, 0,
    , 0, 0,,,,,, 0, 0, 0, 0, 0,,,,, 0,, 0, 0, 0, 0, 0,, 0, 0, 0, 0, 0, 0,
    , 0,, 0, 0, 0, 0,, 0, 0, 0, 0, 0,,,,,,, 0,,, 0, 0,, 0, 0, 0, 0, 0,
    0, 0, 0, 0,, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0 ]
```

Now we want to compute as many nonzero values of 1_H^M as possible, using the same approach as in the previous sections. For that, we first compute several permutation characters 1_X^M, for subgroups X with the property $V < X < U_A$ or $V < X < U_B$. Then we find several subsets $\mathscr{C}$ of M, each being a union of conjugacy classes of M such that $(1_H)_{\mathscr{C} \cap H}$ is a linear combination of the characters 1_X^H, restricted to $\mathscr{C} \cap H$. This yields the values of 1_H^M on the classes in $\mathscr{C}$.

The actual computations are performed with the characters $1_{X/V}^{H/V}$. So we build two parallel lists cand and candbar of permutation characters of M and of H/V, respectively. For that, we write two small GAP functions:

- In the function AddSubgroupOfS82, we choose a subgroup Y of $S_8(2) \cong U_A/V$, compute $1_Y^{U_A/V}$,

inflate it to a character of U_A, induce this character to B, inflate the result to G_A, and finally induce this character to M.

- In the function `AddSubgroupOfO8p2`, we choose a subgroup Y of the factor group $F \cong O_8^+(2)$ of U_B/N, compute 1_Y^F, inflate it to a character of U_B/N, induce this to a character of $G_B/N \cong Co_1$, inflate this to a character of G_B, and finally induce this character to M.

One difficulty in this case is that choosing a subgroup X/V of H/V involves fixing the class fusion into H/V, but it is not clear which is a compatible class fusion of the corresponding subgroup X into M; therefore, each entry of `cand` is in fact not the permutation character of M in question but a list of possibilities.

```
——————————————————— Example ———————————————————
gap> cand:= [ [ pi_A ], [ pi_B ] ];;
gap> candbar:= [ TrivialCharacter( U_Abar )^Hbar,
>                TrivialCharacter( U_Bbar )^Hbar ];;
gap> AddSubgroupOfS82:= function( subname )
>   local psis82;
>
>   psis82:= TrivialCharacter( CharacterTable( subname ) )^U_Abar;
>   Add( cand, [ Restricted( Restricted( psis82, u1 )^b, 2b )^m ] );
>   Add( candbar, psis82 ^ Hbar );
> end;;
gap> tt1:= CharacterTable( "O8+(2)" );
CharacterTable( "O8+(2)" )
gap> AddSubgroupOfO8p2:= function( subname )
>   local psi, list, char;
>
>   psi:= TrivialCharacter( CharacterTable( subname ) )^tt1;
>   list:= [];
>   for char in Orbit( AutomorphismsOfTable( tt1 ), psi, Permuted ) do
>     AddSet( list, Restricted( Restricted( char, u2 ) ^ co1, mm ) ^ m );
>   od;
>   Add( cand, list );
>   Add( candbar, Restricted( psi, U_Bbar ) ^ Hbar );
> end;;
```

Now we choose the subgroups that will turn out to be sufficient for our computations.

```
——————————————————— Example ———————————————————
gap> AddSubgroupOfS82( "O8+(2).2" );
gap> AddSubgroupOfO8p2( "S6(2)" );
gap> AddSubgroupOfS82( "O8-(2).2" );
gap> AddSubgroupOfS82( "A10.2" );
gap> AddSubgroupOfS82( "S4(4).2" );
gap> AddSubgroupOfS82( "L2(17)" );
gap> AddSubgroupOfO8p2( "A9" );
gap> AddSubgroupOfO8p2( "2^6:A8" );
gap> AddSubgroupOfO8p2( "(3xU4(2)):2" );
gap> AddSubgroupOfO8p2( "(A5xA5):2^2" );
gap> AddSubgroupOfS82( "S8(2)M4" );
```

In the case of $A_5 < S_8(2)$, the function `AddSubgroupOfS82` does not work because there are several class fusions of A_5 into $S_8(2)$. We choose one fusion; the fact that it really describes an em-

bedding of an A_5 type subgroup of $S_8(2)$ can be checked using the function `NrPolyhedralSubgroups` (**Reference: NrPolyhedralSubgroups**).

```
——————————— Example ———————————
  gap> a5:= CharacterTable( "A5" );;
  gap> fus:= PossibleClassFusions( a5, U_Abar )[1];;
  gap> NrPolyhedralSubgroups( U_Abar, fus[2], fus[3], fus[4] );
  rec( number := 548352, type := "A5" )
  gap> psis82:= Induced( a5, U_Abar, [ TrivialCharacter( a5 ) ], fus )[1];;
  gap> Add( cand, [ Restricted( Restricted( psis82, u1 )^b, 2b )^m ] );
  gap> Add( candbar, psis82 ^ Hbar );
  gap> List( cand, Length );
  [ 1, 1, 1, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1 ]
```

The following function takes a condition on conjugacy classes in terms of their element orders, which gives a set $\mathscr{C}$ of elements in M. It forms the corresponding set of elements in H/V and tries to express the restriction of $1_{H/V}$ as a linear combination of the characters $1_X^{H/V}$ that are stored in the list `candbar`. If this works and if the corresponding linear combination of the candidates in `cand` is unique, the newly found values of 1_H^M are entered into the list `torso`.

```
——————————— Example ———————————
  gap> Hbarorders:= OrdersClassRepresentatives( Hbar );;
  gap> TryCondition:= function( cond )
  >    local pos, sol, lincomb, oldknown, i;
  >
  >    pos:= Filtered( [ 1 .. Length( Hbarorders ) ],
  >              i -> cond( Hbarorders[i] ) );
  >    sol:= SolutionMat( candbar{[1..Length(candbar)]}{ pos },
  >              ListWithIdenticalEntries( Length( pos ), 1 ) );
  >    if sol = fail then
  >      return "no solution";
  >    fi;
  >
  >    pos:= Filtered( [ 1 .. Length( morders) ], i -> cond( morders[i] ) );
  >    lincomb:= Filtered( Set( Cartesian( cand ), x -> sol * x ),
  >                x -> ForAll( pos, i -> IsPosInt( x[i] ) or x[i] = 0 ) );
  >    if Length( lincomb ) <> 1 then
  >      return "solution is not unique";
  >    fi;
  >
  >    lincomb:= lincomb[1];;
  >    oldknown:= Number( torso );
  >    for i in pos do
  >      if IsBound( torso[i] ) then
  >        if torso[i] <> lincomb[i] then
  >          Error( "contradiction of new and known value at position ", i );
  >        fi;
  >      elif not IsInt( lincomb[i] ) or lincomb[i] < 0 then
  >        Error( "new value at position ", i, " is not a nonneg. integer" );
  >      fi;
  >      torso[i]:= lincomb[i];
  >    od;
  >    return Concatenation( "now ", String( Number( torso ) ), " values (",
```

```
>                  String( Number( torso ) - oldknown ), " new)" );
> end;;
```

This procedure makes sense only if the elements of H that satisfy the condition are contained in the full preimage of the classes of H/V that satisfy the condition. Note that this is in fact the case for the conditions used below. This is clear for condition concerning only *odd* element orders, because V is a 2-group. Also the set of all elements of the orders 9, 18, and 36 is such a "closed" set, since M has no elements of order 72. Finally, the set of all elements of the orders 1, 2, and 4 in H is admissible because it is contained in the preimage of the set of all elements of these orders in H/V.

```
───────────────────────── Example ─────────────────────────
gap> TryCondition( x -> x mod 7 = 0 and x mod 3 <> 0 );
"now 99 values (7 new)"
gap> TryCondition( x -> x mod 17 = 0 and x mod 3 <> 0 );
"now 102 values (3 new)"
gap> TryCondition( x -> x mod 5 = 0 and x mod 3 <> 0 );
"now 119 values (17 new)"
gap> TryCondition( x -> 4 mod x = 0 );
"now 125 values (6 new)"
gap> TryCondition( x -> 9 mod x = 0 );
"now 129 values (4 new)"
gap> TryCondition( x -> x in [ 9, 18, 36 ] );
"now 138 values (9 new)"
```

Possible constituents of 1_H^M are those rational irreducible characters of M that are constituents of π^M.

```
───────────────────────── Example ─────────────────────────
gap> constit:= Filtered( RationalizedMat( Irr( m ) ),
>            x -> ScalarProduct( m, x, pi_A ) <> 0
>                 and ScalarProduct( m, x, pi_B ) <> 0 );;
```

For the missing values, we can provide at least lower bounds, using that $U \leq H \leq G$ implies $1_H^G(g) \geq 1_U^G(g)/[H:U] = [G:H] \cdot 1_U^G(g)/1_U^G(1)$.

```
───────────────────────── Example ─────────────────────────
gap> lower:= [];;
gap> Hindex:= Size( m ) / Horder;
51237270769874105674951529273437
gap> for i in [ 1 .. NrConjugacyClasses( m ) ] do
>    lower[i]:= Maximum( pi_A[i] / ( pi_A[1] / Hindex ),
>                        pi_B[i] / ( pi_B[1] / Hindex ) );
>    if not IsInt( lower[i] ) then
>      lower[i]:= Int( lower[i] + 1 );
>    fi;
> od;
```

Now we compute the possible permutation characters that have the prescribed values, are compatible with the given lower bounds for values, and have only constituents in the given list.

```
───────────────────────── Example ─────────────────────────
gap> cand:= PermChars( m, rec( torso:= torso, chars:= constit,
>     lower:= lower, normalsubgroup:= [ 1 .. NrConjugacyClasses( m ) ],
```

```
>          nonfaithful:= TrivialCharacter( m ) ) );
[ Character( CharacterTable( "M" ),
  [ 5123727069874105674951529273437, 405589064025344574375,
      29628786742129575, 658201521662685, 215448838605, 0,
      121971774375, 28098354375, 335229607, 108472455, 164587500,
      4921875, 2487507165, 2567565, 26157789, 6593805, 398925, 0,
      437325, 0, 44983, 234399, 90675, 21391, 41111, 12915, 6561,
      6561, 177100, 7660, 6875, 315, 275, 0, 113373, 17901, 57213, 0,
      4957, 1197, 909, 301, 397, 0, 0, 0, 3885, 525, 0, 2835, 90, 45,
      0, 103, 67, 43, 28, 81, 189, 9, 9, 9, 0, 540, 300, 175, 20, 15,
      7, 420, 0, 0, 0, 0, 0, 0, 0, 165, 61, 37, 37, 0, 9, 9, 13, 5,
      0, 0, 0, 0, 0, 0, 77, 45, 13, 0, 0, 45, 115, 19, 10, 0, 5, 5,
      9, 9, 1, 1, 0, 0, 4, 0, 0, 9, 9, 3, 1, 0, 0, 0, 0, 0, 0, 4, 1,
      1, 0, 24, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 1, 0, 4, 0, 0, 0,
      0, 1, 0, 0, 0, 0, 0, 3, 3, 1, 1, 2, 0, 3, 3, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0 ] ) ]
```

There is only one candidate, so we have found the permutation character.

The character table of *H* is available since 2023. We can compute the permutation character directly from this table.

```
——————————————— Example ———————————————
gap> h:= CharacterTable( "2^(10+16).O10+(2)" );;
gap> fus:= PossibleClassFusions( h, m );;
gap> cand = Set( fus, map -> InducedClassFunctionsByFusionMap( h, m,
>                                 [ TrivialCharacter( h ) ], map )[1] );
true
```

8.17   A permutation character of the Baby Monster (June 2012)

We compute the character of the Baby Monster that is induced from the trivial character of a Sylow 2-subgroup. (Gabriel Navarro had asked me how GAP can compute this character.) We start with the computation of those transitive permutation characters of the symmetric group on five points that have degree 15. Note that the function PermChars (**Reference: PermChars**) computes in general only candidates, but here we are sure that the result consists of permutation characters because it is unique.

```
——————————————— Example ———————————————
gap> t:= CharacterTable( "S5" );
CharacterTable( "A5.2" )
gap> pi:= PermChars( t, rec( torso:= [ 15 ] ) );
[ Character( CharacterTable( "A5.2" ), [ 15, 3, 0, 0, 3, 1, 0 ] ) ]
```

Next, we regard this character as a character of the group $2^5 : S_5$ that occurs as a maximal subgroup of index 231 in $M_{22} : 2$.

```
——————————————— Example ———————————————
gap> m222:= CharacterTable( "M22.2" );
CharacterTable( "M22.2" )
gap> mx:= List( Maxes( m222 ), CharacterTable );;
gap> mx:= Filtered( mx, x -> Size( m222 ) / Size( x ) = 231 );
[ CharacterTable( "M22.2M4" ) ]
```

```
gap> pi:= pi[1]{ GetFusionMap( mx[1], t ) };
[ 15, 15, 3, 3, 3, 0, 0, 3, 3, 1, 1, 0, 15, 15, 3, 3, 3, 0, 0, 3, 3,
  1, 1, 0 ]
```

We induce this character to $M_{22} : 2$. (Note that this is the character that is induced from the trivial character of a Sylow 2-subgroup of $M_{22} : 2$.)

```
───────────────── Example ─────────────────
gap> pi:= InducedClassFunction( mx[1], pi, m222 );
ClassFunction( CharacterTable( "M22.2" ),
 [ 3465, 105, 0, 9, 5, 0, 0, 0, 0, 1, 0, 189, 45, 9, 13, 0, 1, 0, 0,
   0, 0 ] )
```

Next, we regard this character as a character of the group $2^{10} : M_{22} : 2$ that occurs as a maximal subgroup of index 46575 in Co_2.

```
───────────────── Example ─────────────────
gap> co2:= CharacterTable( "Co2" );
CharacterTable( "Co2" )
gap> mx:= List( Maxes( co2 ), CharacterTable );;
gap> mx:= Filtered( mx, x -> Size( co2 ) / Size( x ) = 46575 );
[ CharacterTable( "2^10:m22:2" ) ]
gap> pi:= pi{ GetFusionMap( mx[1], m222 ) };
[ 3465, 3465, 3465, 3465, 105, 105, 105, 105, 105, 105, 105, 105, 0,
  0, 0, 0, 0, 9, 9, 9, 9, 9, 9, 5, 5, 5, 5, 5, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 1, 1, 1, 0, 189, 189, 189, 189, 189, 189, 45, 45, 45, 45,
  9, 9, 9, 9, 13, 13, 13, 13, 13, 13, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0,
  0, 0, 0, 0, 0, 0, 0 ]
```

We induce this character to Co_2.

```
───────────────── Example ─────────────────
gap> pi:= InducedClassFunction( mx[1], pi, co2 );
ClassFunction( CharacterTable( "Co2" ),
 [ 161382375, 626535, 162855, 27495, 0, 0, 6615, 3975, 2727, 855,
   567, 975, 115, 0, 0, 0, 0, 0, 0, 0, 0, 0, 63, 51, 19, 27, 35, 7, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0 ] )
```

Next, we regard this character as a character of the group $2^{1+22}.Co_2$ that occurs as a maximal subgroup of index 11707448673375 in the Baby Monster.

```
───────────────── Example ─────────────────
gap> b:= CharacterTable( "B" );
CharacterTable( "B" )
gap> mx:= List( Maxes( b ), CharacterTable );;
gap> mx:= Filtered( mx, x -> Size( b ) / Size( x ) = 11707448673375 );
[ CharacterTable( "2^(1+22).Co2" ) ]
gap> pi:= pi{ GetFusionMap( mx[1], co2 ) };;
gap> pi[1];
161382375
```

We induce this character to the Baby Monster.

```
─────────────── Example ───────────────
gap> pi:= InducedClassFunction( mx[1], pi, b );
ClassFunction( CharacterTable( "B" ),
 [ 1889375872099856765625, 2609385408855225, 62316674429625,
   207818526825, 268788490425, 0, 0, 13052741625, 7537207545,
   128298681, 270580905, 46366425, 74315385, 35633385, 3937689,
   201825, 1233225, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 713097,
   241425, 320625, 88521, 275265, 57705, 19305, 20089, 9441, 6489,
   2577, 1825, 5345, 753, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   273, 417, 105, 97, 185, 33, 9, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] )
```

8.18 A permutation character of $2.B$ (October 2017)

We compute the character of the double cover $2.B$ of the Baby Monster that is induced from the trivial character of a subgroup U of the structure $2^{1+22}.McL$.

This subgroup occurs as the intersection of two conjugates of $2.B$ inside the Monster group M. More precisely, we consider $2.B$ as the centralizer of an involution a in M, and we are interested in the permutation action of M on the cosets of $2.B$ (or, equivalently, on the conjugacy class in M of this involution). The restriction of this action to $2.B$ has nine orbits. One of them has point stabilizer U.

Background information can be found in [GJMS89]. The decomposition into the nine orbits appears in Definition (3.4.9) on p 587, and our orbit is characterized in Table VII (on p. 582) by the facts that its points c have order 4 and the squares of ac lie in the class 2B of M. This implies that a and c do not commute, hence a does not lie in U.

From this description, we know that U is a subgroup of a maximal subgroup of the type $2^{2+22}.Co_2$ in $2.B$, and the group $\langle U, a \rangle$ has the type $2^{2+22}.McL$.

Thus we can proceed in two steps. First we induce the trivial character of $\langle U, a \rangle$ to $2.B$. Then we use the variant of the GAP function `PermChars` (**Reference: PermChars**) that allows us to prescribe the permutation character of the closure with a normal subgroup, which is $\langle a \rangle$ in our case.

The first step can be performed by inducing the trivial character of McL to Co_2, . . .

```
─────────────── Example ───────────────
gap> mcl:= CharacterTable( "McL" );
CharacterTable( "McL" )
gap> co2:= CharacterTable( "Co2" );
CharacterTable( "Co2" )
gap> ind:= Induced( mcl, co2, [ TrivialCharacter( mcl ) ] )[1];
Character( CharacterTable( "Co2" ),
 [ 47104, 0, 1024, 0, 16, 160, 0, 0, 0, 0, 64, 0, 0, 4, 24, 16, 0, 0,
   0, 16, 0, 8, 0, 0, 0, 0, 0, 8, 4, 4, 0, 0, 2, 0, 0, 0, 0, 0, 0, 4,
   0, 0, 2, 2, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1 ] )
```

. . . regarding this character as a character of $2^{1+22}.Co_2$, . . .

```
─────────────── Example ───────────────
gap> m:= CharacterTable( "BM2" );
CharacterTable( "2^(1+22).Co2" )
```

```
gap> infl:= ind{ GetFusionMap( m, co2 ) };
[ 47104, 47104, 47104, 47104, 47104, 47104, 47104, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 1024, 1024, 1024, 1024, 1024, 1024, 1024, 1024,
  1024, 1024, 1024, 1024, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 16, 16, 16, 16, 160, 160, 160, 160, 160, 160, 160, 160,
  160, 160, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 64, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 4, 4, 4, 24, 24, 24, 24, 24, 24, 24, 24, 16, 16, 16,
  16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16,
  16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 8, 8, 8,
  8, 8, 8, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 4, 4,
  4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2,
  2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 4, 4,
  4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  1, 1, 1, 1 ]
```

… inducing this character to *B*, …

```
_____ Example _____
gap> b:= CharacterTable( "B" );
CharacterTable( "B" )
gap> ind:= Induced( m, b, [ infl ] )[1];
ClassFunction( CharacterTable( "B" ),
 [ 551467662310656000, 186911262720, 272993634304, 0, 634521600,
  194594400, 69984, 8495104, 17465344, 129024, 276480, 2073600,
  16384, 798720, 46080, 0, 5120, 138600, 1000, 110880, 252000,
  112480, 432, 12960, 0, 1312, 8352, 864, 432, 0, 2520, 0, 2880,
  2880, 3072, 2880, 0, 0, 256, 64, 1152, 576, 640, 192, 96, 0, 108,
  2520, 744, 0, 104, 120, 40, 30, 160, 16, 1120, 1024, 0, 0, 96, 288,
  64, 144, 0, 96, 0, 108, 16, 48, 0, 32, 12, 0, 0, 0, 168, 0, 104,
  48, 0, 4, 0, 0, 0, 0, 32, 16, 8, 8, 0, 24, 12, 4, 0, 0, 0, 0, 24,
  4, 24, 24, 0, 0, 0, 0, 4, 0, 0, 6, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 8, 0, 16, 8, 4, 0, 0, 0, 0, 0, 4, 2,
  2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] )
```

… and regarding the result as a character of 2.*B*.

```
_____ Example _____
gap> 2b:= CharacterTable( "2.B" );
CharacterTable( "2.B" )
gap> infl:= ind{ GetFusionMap( 2b, b ) };
[ 551467662310656000, 551467662310656000, 186911262720, 272993634304,
  272993634304, 0, 634521600, 194594400, 194594400, 69984, 69984,
```

```
   8495104, 17465344, 129024, 276480, 2073600, 2073600, 16384, 798720,
   46080, 0, 5120, 138600, 138600, 1000, 1000, 110880, 252000, 112480,
   112480, 432, 12960, 0, 1312, 1312, 8352, 864, 864, 432, 0, 2520,
   2520, 0, 2880, 2880, 3072, 2880, 0, 0, 256, 64, 1152, 576, 576,
   640, 192, 96, 0, 0, 108, 108, 2520, 744, 744, 0, 104, 104, 120, 40,
   40, 30, 30, 160, 16, 1120, 1024, 0, 0, 0, 96, 288, 64, 144, 144, 0,
   96, 0, 108, 108, 16, 48, 0, 32, 12, 12, 0, 0, 0, 0, 168, 0, 104,
   104, 48, 0, 0, 4, 4, 0, 0, 0, 0, 32, 16, 8, 8, 8, 0, 0, 24, 12, 4,
   4, 0, 0, 0, 0, 0, 0, 24, 4, 24, 24, 0, 0, 0, 0, 0, 4, 0, 0, 0, 6,
   6, 6, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0,
   0, 0, 0, 0, 0, 0, 8, 0, 16, 8, 4, 4, 0, 0, 0, 0, 0, 0, 4, 4, 2, 2,
   2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
   0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

Now we have the character ψ that represents the "nonfaithful half" of the desired permutation character. We have to "fill it up" with faithful characters of 2.*B* of total degree $\psi(1)$ such that the sum with ψ can be a permutation character of 2.*B*.

The GAP function PermChars (**Reference: PermChars**) is designed for this situation. We specify the normal subgroup $N = \langle a \rangle$ by listing the positions of its conjugacy classes in the character table of 2.*B*, we enter the known permutation character $1_{UN}^{2.B}$, and of course we specify the degree of the possible permutation characters.

```
                               Example
  gap> centre:= ClassPositionsOfCentre( 2b );
  [ 1, 2 ]
  gap> pi:= PermChars( 2b, rec( torso:= [ 2 * infl[1], 0 ],
  >                                 normalsubgroup:= centre,
  >                                 nonfaithful:= infl ) );
  [ Character( CharacterTable( "2.B" ),
    [ 1102935324621312000, 0, 186911262720, 541790208000, 4197060608,
        0, 634521600, 389188800, 0, 139968, 0, 8495104, 17465344,
        129024, 276480, 4026240, 120960, 16384, 798720, 46080, 0, 5120,
        277200, 0, 2000, 0, 110880, 252000, 190080, 34880, 432, 12960,
        0, 2592, 32, 8352, 1728, 0, 432, 0, 5040, 0, 0, 2880, 2880,
        3072, 2880, 0, 0, 256, 64, 1152, 1008, 144, 640, 192, 96, 0, 0,
        216, 0, 2520, 960, 528, 0, 200, 8, 120, 80, 0, 60, 0, 160, 16,
        1120, 1024, 0, 0, 0, 96, 288, 64, 216, 72, 0, 96, 0, 216, 0,
        16, 48, 0, 32, 24, 0, 0, 0, 0, 0, 168, 0, 160, 48, 48, 0, 0, 8,
        0, 0, 0, 0, 0, 32, 16, 8, 12, 4, 0, 0, 24, 12, 0, 8, 0, 0, 0,
        0, 0, 0, 24, 4, 24, 24, 0, 0, 0, 0, 0, 4, 0, 0, 0, 6, 6, 8, 4,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0,
        0, 0, 0, 0, 8, 0, 16, 8, 8, 0, 0, 0, 0, 0, 0, 0, 8, 0, 2, 2, 2,
        2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ) ]
  gap> MatScalarProducts( 2b, Irr( 2b ), pi );
  [ [ 1, 1, 2, 1, 2, 0, 2, 3, 2, 0, 0, 1, 4, 1, 2, 0, 3, 2, 0, 2, 0, 0,
        2, 2, 0, 0, 2, 3, 1, 5, 0, 4, 3, 2, 0, 0, 3, 2, 0, 6, 4, 0, 1,
        1, 0, 0, 0, 0, 3, 0, 1, 0, 0, 5, 0, 5, 2, 0, 0, 2, 0, 0, 4, 1,
        0, 2, 0, 4, 2, 4, 4, 3, 0, 2, 4, 2, 4, 0, 3, 0, 3, 2, 5, 0, 1,
        0, 3, 1, 0, 1, 1, 2, 5, 3, 1, 1, 4, 5, 1, 1, 0, 3, 0, 0, 3, 2,
```

```
1, 1, 2, 1, 1, 4, 0, 3, 2, 3, 1, 3, 0, 1, 3, 0, 2, 2, 1, 3, 3,
0, 0, 2, 0, 0, 0, 0, 3, 0, 3, 3, 3, 1, 0, 3, 0, 4, 0, 1, 0, 0,
2, 0, 0, 2, 0, 0, 2, 1, 1, 0, 0, 0, 0, 1, 2, 1, 1, 1, 0, 1, 1,
1, 1, 1, 1, 0, 2, 1, 1, 3, 3, 0, 0, 0, 1, 1, 1, 1, 2, 3, 2, 0,
0, 2, 2, 4, 3, 5, 2, 4, 0, 0, 0, 0, 5, 2, 0, 0, 0, 1, 1, 0, 0,
0, 0, 0, 0, 7, 0, 0, 1, 7, 7, 0, 0, 0, 1, 6, 4, 5, 0, 0, 3, 0,
0, 0, 0, 0, 4, 1, 1, 3, 8, 3, 2, 2, 5, 0, 1 ] ]
```

We are lucky: There is a unique solution, and its computation is quite fast.

8.19 Generation of sporadic simple groups by π- and π'-subgroups (December 2021)

This section shows the computations that are needed in order to show the following statements from [BG].

At the time when this paper was written, a special treatment of the Monster group was necessary because the classification of its maximal subgroups was not yet complete, and also not all character tables of the known maximal subgroups were available. The now unnecessary computations for the Monster group are shown in Section 8.19.1.

Proposition 2.2: Let S be a sporadic simple group and let P be a Sylow 2-subgroup of S. If $1 \neq x \in S$, then $S = \langle P, x^g \rangle$ for some $g \in S$.

Theorem 7.1: Let S be a sporadic simple group and let $p \leq q$ be primes each dividing $|S|$. Then S can be generated by a Sylow p-subgroup and a Sylow q-subgroup.

A stronger version of Theorem 7.1: Let S be a sporadic simple group, p be a prime dividing $|S|$, and P be a Sylow p-subgroup of S. If $1 \neq x \in S$, then $S = \langle P, x^g \rangle$ for some $g \in S$.

First we show [BG, Proposition 2.2]. Let S be a sporadic simple group, fix a Sylow 2-subgroup P of S, and let x be a nonidentity element in S. We use known information about maximal subgroups of S to show that x^S is not a subset of the union of those maximal subgroups in S that contain P.

Let M be a maximal subgroup of S with the property $P \leq M$. The number of S-conjugates of M that contain P is equal to $|N_S(P)|/|N_M(P)| \leq [N_S(P) : P]$, thus these subgroups can contain at most $[N_S(P) : P]|x^S \cap M|$ elements from the class x^S.

Thus the number of elements in x^S that generate a proper subgroup of S together with P is bounded from above by $[N_S(P) : P]\sum_M |x^S \cap M|$, where the sum is taken over representatives M of conjugacy classes of maximal subgroups of odd index in S.

Let 1_M^S denote the permutation character of the action of S on the cosets of M. We have $|x^S \cap M| = |x^S|1_M^S(x)/1_M^S(1)$. Hence we are done when we show that

$$[N_S(P) : P]\sum_M 1_M^S(x)/1_M^S(1) < 1$$

holds.

The numbers $[N_S(P) : P]$ can be read off from [Wil98, Table I]. Here we use the fact that the character table of the Sylow 2-normalizer of S is available except if S is one of Co_1, J_4, F_{3+}, B, or M, and that the Sylow 2-subgroup if self-normalizing in these cases.

```
────────────────────────── Example ──────────────────────────
 gap> names:= AllCharacterTableNames( IsSporadicSimple, true,
 >             IsDuplicateTable, false : OrderedBy:= Size );
 [ "M11", "M12", "J1", "M22", "J2", "M23", "HS", "J3", "M24", "McL",
```

```
    "He", "Ru", "Suz", "ON", "Co3", "Co2", "Fi22", "HN", "Ly", "Th",
    "Fi23", "Co1", "J4", "F3+", "B", "M" ]
gap> normindices:= rec( Co1:= 1, J4:= 1, F3\+:= 1, B:= 1, M:= 1 );;
gap> for name in names do
>       n:= CharacterTable( Concatenation( name, "N2" ) );
>       if n = fail then
>         Print( name, "\n" );
>       else
>         2part:= 2^Length( Positions( Factors( Size( n ) ), 2 ) );
>         normindices.( name ):= Size( n ) / 2part;
>       fi;
>     od;
Co1
J4
F3+
B
M
```

For all sporadic simple groups S, the primitive permutation characters 1_M^S can be computed from the data about maximal subgroups contained in GAP's library of character tables.

```
────────────────── Example ──────────────────
gap> maxbound:= [];;
gap> for name in names do
>       t:= CharacterTable( name );
>       mx:= List( Maxes( t ), CharacterTable );
>       odd:= Filtered( mx, s -> ( Size( t ) / Size( s ) ) mod 2 <> 0 );
>       primperm:= List( odd, s -> TrivialCharacter( s )^t );
>       sum:= normindices.( name ) * Sum( primperm, pi -> pi / pi[1] );
>       Add( maxbound,
>             [ name, Maximum( sum{ [ 2 .. Length( sum ) ] } ) ] );
>     od;
gap> SortBy( maxbound, x -> - x[2] );
gap> maxbound[1];
[ "J2", 3/5 ]
```

We see that the left hand side of the above inequality is always less than or equal to $3/5$, in particular it is less than 1.

Next we show [BG, Theorem 7.1] and its stronger version stated above.

As a first step, we generalize the approach from the above computations in order to check for which prime divisors p of $|S|$ and for which nontrivial conjugacy classes x^S of S the group S is generated by a Sylow p-subgroup P together with a conjugate of x.

The upper bound $[N_S(P) : P]$ for $|N_S(P)|/|N_M(P)|$, for a maximal subgroup M of S that contains P, is not good enough in some of the cases considered here. Instead of it, we compute the upper bound $u(S, M, p)$ which is defined as follows; we assume that we know $|N_S(P)|$.

- If P is cyclic then we can compute $|N_M(P)|$ from the character table of M, and set $u(S, M, p) = |N_S(P)|/|N_M(P)|$.

- Otherwise, if P is normal in M, we set $u(S, M, p) = |N_S(P)|/|M|$.

- Otherwise, if we know a subgroup U of M such that P is a proper normal subgroup of U, we set $u(S,M,p) = |N_S(P)|/|U|$.

- Otherwise, we set $u(S,M,p) = |N_S(P)|/|P|$.

—————————— Example ——————————

```
gap> upper_bound:= function( tblS, tblM, p )
>     local ppart, ppartposS, ppartposM, n, N_S, f, subname, u;
>
>     ppart:= Product( Filtered( Factors( Size( tblS ) ), x -> x = p ), 1 );
>     ppartposS:= Positions( OrdersClassRepresentatives( tblS ), ppart );
>     if 0 < Length( ppartposS ) then
>       # P is cyclic.
>       if tblM = fail then
>         return ( SizesCentralizers( tblS )[ ppartposS[1] ] * Phi( ppart )
>                   / Length( ppartposS ) ) / ppart;
>       else
>         ppartposM:= Positions( OrdersClassRepresentatives( tblM ), ppart );
>         return ( SizesCentralizers( tblS )[ ppartposS[1] ] * Phi( ppart )
>                   / Length( ppartposS ) ) /
>               ( SizesCentralizers( tblM )[ ppartposM[1] ] * Phi( ppart )
>                   / Length( ppartposM ) );
>       fi;
>     fi;
>
>     # Compute |N_S(P)|.
>     n:= CharacterTable( Concatenation( Identifier( tblS ), "N",
>                           String( p ) ) );
>     if n <> fail then
>       N_S:= Size( n );
>     elif p = 2 then
>       N_S:= ppart * normindices.( Identifier( tblS ) );
>     elif Identifier( tblS ) = "M" and p = 3 then
>       # The Sylow 3-normalizer is contained in 3^(3+2+6+6):(L3(3)xSD16)
>       N_S:= ppart * 2^6;
>     elif Identifier( tblS ) = "F3+" and p = 3 then
>       N_S:= ppart * 8;
>     else
>       Error( "cannot compute |N_S(P)|" );
>     fi;
>
>     if tblM = fail then
>       return N_S / ppart;
>     elif Sum( SizesConjugacyClasses( tblM ){
>                 ClassPositionsOfPCore( tblM, p ) } ) = ppart then
>       # P is normal in M.
>       return N_S / Size( tblM );
>     fi;
>
>     # Inspect known character tables of subgroups of M.
>     f:= N_S / ppart;
>     for subname in NamesOfFusionSources( tblM ) do
>       u:= CharacterTable( subname );
>       if ClassPositionsOfKernel( GetFusionMap( u, tblM ) ) = [ 1 ] and
```

```
>             Sum( SizesConjugacyClasses( u ){
>                     ClassPositionsOfPCore( u, p ) } ) = ppart then
>           f:= Minimum( f, N_S / Size( u ) );
>       fi;
>     od;
>
>     return f;
>  end;;
```

We run over the sporadic simple groups and collect in the list `badcases_strong` those "bad" prime divisors p of $|S|$ and conjugacy class representatives x of nonidentity elements in S for which

$$\sum_M u(S,M,p) 1^S_M(x) / 1^S_M(1) \geq 1$$

holds, where the sum is taken over representatives M of conjugacy classes of maximal subgroups of S whose index in S is coprime to p. In these cases, we have to find other arguments.

For the proof of [BG, Theorem 7.1], we can discard all those entries from the list of "bad" p and x where x is not a q-element, for some prime q, or where another nonidentity q-element exists that does not occur in the list. This is done by collecting a second list `badcases_thm` of the remaining "bad" cases.

For the proof of the stronger version, we will later explicitly compute group elements from the classes in question that generate S together with a Sylow p-subgroup. (The only technical complication is that the class fusion of maximal subgroups of the type $(2^2 \times F_4(2)) : 2$ of the Baby Monster is currently not known, thus we cannot simply induce the trivial character in this case. However, the permutation character is uniquely determined by the two character tables.)

```
───────────────────────────── Example ─────────────────────────────
gap> badcases_thm:= [];;
gap> badcases_strong:= [];;
gap> good:= "dummy";;   #  Avoid a message about an unbound variable ...
gap> for name in Filtered( names, x -> x <> "M" ) do
>       t:= CharacterTable( name );
>       orders:= OrdersClassRepresentatives( t );
>       n:= NrConjugacyClasses( t );
>       mx:= List( Maxes( t ), CharacterTable );
>       for p in PrimeDivisors( Size( t ) ) do
>         good:= Filtered( mx, s -> ( Size( t ) / Size( s ) ) mod p <> 0 );
>         primperm:= [];
>         for s in good do
>           if GetFusionMap( s, t ) <> fail then
>             Add( primperm, TrivialCharacter( s )^t );
>           else
>             ind:= Set( PossibleClassFusions( s, t ),
>                     map -> InducedClassFunctionsByFusionMap( s, t,
>                             [ TrivialCharacter( s ) ], map )[1] );
>             if Length( ind ) <> 1 then
>               Error( "permutation character not uniquely determined" );
>             fi;
>             Add( primperm, ind[1] );
>           fi;
>         od;
```

```
>            sum:= Sum( [ 1 .. Length( good ) ],
>                       i -> upper_bound( t, good[i], p )
>                            * primperm[i] / primperm[i][1] );
>            badpos:= Filtered( [ 2 .. Length( sum ) ], i -> sum[i] >= 1 );
>            if badpos <> [] then
>              Add( badcases_strong, [ name, p, ShallowCopy( badpos ) ] );
>              for i in ShallowCopy( badpos ) do
>                q:= SmallestRootInt( orders[i] );
>                if IsPrimeInt( q ) then
>                  if ForAny( [ 2 .. n ],
>                             j -> SmallestRootInt( orders[j] ) = q
>                                  and not j in badpos ) then
>                    RemoveSet( badpos, i );
>                  fi;
>                fi;
>              od;
>              if not IsEmpty( badpos ) then
>                Add( badcases_thm, [ name, p, badpos ] );
>              fi;
>            fi;
>          od;
>      od;
gap> badcases_thm;
[ [ "M23", 3, [ 3 ] ], [ "HS", 3, [ 4, 11 ] ] ]
gap> badcases_strong;
[ [ "M11", 5, [ 2 ] ], [ "M12", 5, [ 3, 4 ] ], [ "M22", 3, [ 2 ] ],
  [ "M22", 5, [ 2 ] ], [ "J2", 3, [ 2 ] ], [ "M23", 3, [ 2, 3 ] ],
  [ "M23", 5, [ 2 ] ], [ "M23", 7, [ 2 ] ],
  [ "HS", 3, [ 2, 3, 4, 5, 6, 7, 9, 11 ] ], [ "HS", 5, [ 2, 3, 5 ] ],
  [ "M24", 5, [ 2, 4 ] ], [ "M24", 7, [ 2, 4 ] ], [ "He", 5, [ 2 ] ],
  [ "Co2", 3, [ 2, 3 ] ], [ "Fi22", 5, [ 2 ] ], [ "Fi22", 7, [ 2 ] ],
  [ "Fi23", 5, [ 2, 3, 5 ] ], [ "Fi23", 7, [ 2 ] ], [ "B", 7, [ 2 ] ]
 ]
```

Most of these open cases can be ruled out by constructing the group S and a Sylow p-subgroup P in question and then finding explicit elements x such that S is generated by P and x. For that, we use the data from the Atlas of Group Representations [WWT+].

The following function tries to find random elements from all conjugacy classes of nonidentity elements that have the desired property. It returns `fail` if no straight line program is available for computing class representatives, and returns P and the list of class representatives that generate together with P if such elements were found. Thus the function will not return if the generation property does not hold.

```
─────────────────────── Example ───────────────────────
gap> prove_generation:= function( name, p )
>    local S, prg, P, reps, good, x, g, U;
>
>    prg:= AtlasProgram( name, "classes" );
>    if prg = fail then
>      return fail;
>    fi;
>
>    S:= AtlasGroup( name );
```

```
>     P:= SylowSubgroup( S, p );
>     reps:= ResultOfStraightLineProgram( prg.program, GeneratorsOfGroup( S ) );
>     good:= [];
>     for x in Filtered( reps, x -> Order( x ) <> 1 ) do
>       repeat
>         g:= Random( S );
>         U:= ClosureGroup( P, x^g );
>       until Size( U ) = Size( S );
>       Add( good, x^g );
>     od;
>
>     return [ P, good ];
> end;;
gap> for entry in badcases_strong do
>       res:= prove_generation( entry[1], entry[2] );
>       if res = fail then
>         Print( "no classes script for ", entry, "\n" );
>       fi;
>     od;
no classes script for [ "He", 5, [ 2 ] ]
no classes script for [ "Fi22", 5, [ 2 ] ]
no classes script for [ "Fi22", 7, [ 2 ] ]
no classes script for [ "Fi23", 5, [ 2, 3, 5 ] ]
no classes script for [ "Fi23", 7, [ 2 ] ]
no classes script for [ "B", 7, [ 2 ] ]
```

In the remaining cases, we show only the generation property for the class representatives in the list. These are involutions from the class 2A, and for the group Fi_{23} and $p = 5$ additionally elements from the classes 2B and 3A.

A 2A element in the group *He* can be found as the fifth power of any element of order 10.

―――――――――――― Example ――――――――――――
```
gap> S:= AtlasGroup( "He" );;
gap> repeat
>      x:= Random( S );
>    until Order( x ) = 10;
gap> x:= x^5;;
gap> P5:= SylowSubgroup( S, 5 );;
gap> repeat
>      g:= Random( S );
>      U:= ClosureGroup( P5, x^g );
>    until Size( U ) = Size( S );
```

A 2A element in the group Fi_{22} can be found as the 15-th power of any element of order 30.

―――――――――――― Example ――――――――――――
```
gap> S:= AtlasGroup( "Fi22" );;
gap> repeat
>      x:= Random( S );
>    until Order( x ) = 30;
gap> x:= x^15;;
gap> P5:= SylowSubgroup( S, 5 );;
gap> repeat
```

```
>       g:= Random( S );
>       U:= ClosureGroup( P5, x^g );
>    until Size( U ) = Size( S );
gap> P7:= SylowSubgroup( S, 7 );;
gap> repeat
>       g:= Random( S );
>       U:= ClosureGroup( P7, x^g );
>    until Size( U ) = Size( S );
```

A 2A element in the group Fi_{23} can be found as the 21-st power of any element of order 42.

─── Example ───
```
gap> S:= AtlasGroup( "Fi23" );;
gap> repeat
>       x:= Random( S );
>    until Order( x ) = 42;
gap> x:= x^21;;
gap> P5:= SylowSubgroup( S, 5 );;
gap> repeat
>       g:= Random( S );
>       U:= ClosureGroup( P5, x^g );
>    until Size( U ) = Size( S );
gap> P7:= SylowSubgroup( S, 7 );;
gap> repeat
>       g:= Random( S );
>       U:= ClosureGroup( P7, x^g );
>    until Size( U ) = Size( S );
```

A 2B element in the group Fi_{23} can be found as the 30-th power of any element of order 60.

─── Example ───
```
gap> repeat
>       x:= Random( S );
>    until Order( x ) = 60;
gap> x:= x^30;;
gap> repeat
>       g:= Random( S );
>       U:= ClosureGroup( P5, x^g );
>    until Size( U ) = Size( S );
```

A 3A element in the group Fi_{23} can be found as the 20-th power of any element of order 60.

─── Example ───
```
gap> repeat
>       x:= Random( S );
>    until Order( x ) = 60;
gap> x:= x^20;;
gap> repeat
>       g:= Random( S );
>       U:= ClosureGroup( P5, x^g );
>    until Size( U ) = Size( S );
```

In the open case for the Baby Monster, we have to show that the group is generated by a 2A element and an element of order 7. This can be done character-theoretically, for example as follows. There are such elements *x* and *y* whose product *xy* has order 47, and the only proper subgroups of the Baby Monster that contain elements of order 47 are contained in maximal subgroups of the type 47 : 23. Thus *x* and *y* generate the Baby Monster.

```
———————————————————— Example ————————————————————
gap> t:= CharacterTable( "B" );;
gap> 7pos:= Positions( OrdersClassRepresentatives( t ), 7 );
[ 31 ]
gap> 47pos:= Positions( OrdersClassRepresentatives( t ), 47 );
[ 172, 173 ]
gap> ClassMultiplicationCoefficient( t, 2, 7pos[1], 47pos[1] );
7332
gap> Filtered( Maxes( t ),
>         x -> Size( CharacterTable( x ) ) mod 47 = 0 );
[ "47:23" ]
```

8.19.1 Special Arguments for the Monster Group that are no longer needed

The following computations were needed to prove the statements in question before the complete list of conjugacy classes of maximal subgroups of the Monster group was known, due to [DLP25], and before the character tables of all maximal subgroups of the Monster were available.

First we show [BG, Proposition 2.2].

The Monster group is known to contain exactly five classes of maximal subgroups of odd index, of the structures $2^{1+24}.Co_1$ (the normalizer of a 2B element in the Monster), $2^{10+16}.O_{10}^+(2)$, $2^{2+11+22}.(M_{24} \times S_3)$, $2^{5+10+20}.(S_3 \times L_5(2))$, $[2^{39}].(L_3(2) \times 3S_6)$. The corresponding permutation characters are known, see Section 8.16. First we read the information about the known primitive permutation characters of the Monster into the GAP session, and extract the primitive permutation characters of odd degree.

```
———————————————————— Example ————————————————————
gap> dir:= DirectoriesPackageLibrary( "ctbllib", "data" );;
gap> filename:= Filename( dir, "prim_perm_M.json" );;
gap> Monster_prim_data:= EvalString( StringFile( filename ) )[2];;
gap> Length( Monster_prim_data );
46
gap> t:= CharacterTable( "M" );;
gap> monstermaxindices:= [];;
gap> monstermaxtables:= [];;
gap> for entry in Monster_prim_data do
>     if Length( entry ) = 1 then
>       s:= CharacterTable( entry[1] );
>       Add( monstermaxtables, s );
>       Add( monstermaxindices, Size( t ) / Size( s ) );
>     else
>       Add( monstermaxtables, fail );
>       Add( monstermaxindices, entry[2][1] );
>     fi;
>   od;
gap> odd_prim:= [];;
gap> for i in [ 1 .. Length( Monster_prim_data ) ] do
```

```
>        if monstermaxindices[i] mod 2 <> 0 then
>          if monstermaxtables[i] <> fail then
>            Add( odd_prim, TrivialCharacter( monstermaxtables[i] )^t );
>          else
>            Add( odd_prim, Monster_prim_data[i][2] );
>          fi;
>        fi;
>    od;
gap> Length( odd_prim );
5
```

Now we can use the same approach as for the other sporadic simple groups.

```
─────────────────────── Example ───────────────────────
gap> sum:= normindices.M * Sum( odd_prim, pi -> pi / pi[1] );;
gap> max:= Maximum( sum{ [ 2 .. Length( sum ) ] } );
12784979/103007903752128375
gap> max < 10^-9;
true
```

Next we show [BG, Theorem 7.1] and its stronger version stated above.

From [NW13] and [Wil] we know 44 classes of maximal subgroups, and that each possible additional maximal subgroup is almost simple and has socle $L_2(13)$, $U_3(4)$, $U_3(8)$, or $Sz(8)$. This implies that we know all those maximal subgroups that contain a Sylow p-subgroup of S except in the case $p = 19$, where maximal subgroups with socle $U_3(8)$ may arise.

Thus let us first consider that at least one of p, r is different from 19. In this situation, we use the same approach as for the other sporadic simple groups. The only complication is that not all permutation characters 1_M^S, for the relevant maximal subgroups M of S, are known; however, if this happens then the character table of M is known, and we can compute the possible permutation characters, and take the common upper bounds for these characters. In each case, we get that the claimed property holds.

```
─────────────────────── Example ───────────────────────
gap> t:= CharacterTable( "M" );;
gap> orders:= OrdersClassRepresentatives( t );;
gap> for p in Difference( PrimeDivisors( Size( t ) ), [ 19 ] ) do
>    goodpos:= Filtered( [ 1 .. Length( Monster_prim_data ) ],
>                         i -> monstermaxindices[i] mod p <> 0 );
>    sum:= ListWithIdenticalEntries( NrConjugacyClasses( t ), 0 );
>    for i in goodpos do
>      if Length( Monster_prim_data[i] ) = 2 then
>        # We know the permutation character but not the subgroup table.
>        sum:= sum + upper_bound( t, fail, p )
>                  * Monster_prim_data[i][2] / monstermaxindices[i];
>      else
>        s:= monstermaxtables[i];
>        if GetFusionMap( s, t ) <> fail then
>          # We can compute the permutation character.
>          sum:= sum + upper_bound( t, s, p )
>                    * TrivialCharacter( s )^t / monstermaxindices[i];
>        else
>          # We get only candidates for the permutation character.
```

```
>          cand:= Set( PossibleClassFusions( s, t ),
>                  map -> InducedClassFunctionsByFusionMap( s, t,
>                        [ TrivialCharacter( s ) ], map )[1] );
>          # For each class, take the maximum of the possible values.
>          sum:= sum + upper_bound( t, s, p )
>                  * List( TransposedMat( cand ), Maximum )
>                  / monstermaxindices[i];
>        fi;
>      fi;
>    od;
>    badpos:= Filtered( [ 2 .. Length( sum ) ], i -> sum[i] >= 1 );
>    if badpos <> [] then
>      Error( "check open cases in ", badpos, "\n" );
>    fi;
> od;
```

Finally, let $p = r = 19$. The group S has exactly one class of elements of order 19. Let x be such an element. From the character table of S, we compute that there exist conjugates y of x such that xy has order 71. Since $\langle x, y \rangle = \langle x, xy \rangle$ holds and no maximal subgroup of S has order divisible by $19 \cdot 71$, we have $\langle x, y \rangle = S$.

```
———————————————— Example ————————————————
gap> pos19:= Positions( OrdersClassRepresentatives( t ), 19 );
[ 63 ]
gap> pos71:= Positions( OrdersClassRepresentatives( t ), 71 );
[ 169, 170 ]
gap> ClassMultiplicationCoefficient( t, pos19[1], pos19[1], pos71[1] );
6217431523705660204178063536023874334415016198936
gap> ForAny( monstermaxindices,
>            x -> ( Size( t ) / x ) mod ( 19 * 71 ) = 0 );
false
gap> ForAny( [ "L2(13)", "U3(4)", "U3(8)", "Sz(8)" ],
>            x -> Size( CharacterTable( x ) ) mod 71 = 0 );
false
```

Chapter 9

Ambiguous Class Fusions in the **GAP** Character Table Library

Date: January 11th, 2004

This is a collection of examples showing how class fusions between character tables can be determined using the **GAP** system [GAP24]. In each of these examples, the fusion is *ambiguous* in the sense that the character tables do not determine it up to table automorphisms. Our strategy is to compute first all possibilities with the **GAP** function `PossibleClassFusions` (**Reference: PossibleClassFusions**), and then to use either other character tables or information about the groups for excluding some of these candidates until only one (orbit under table automorphisms) remains.

The purpose of this writeup is twofold. On the one hand, the computations are documented this way. On the other hand, the **GAP** code shown for the examples can be used as test input for automatic checking of the data and the functions used; therefore, each example ends with a comparison of the result with the fusion that is actually stored in the **GAP** Character Table Library [Bre25].

The examples use the **GAP** Character Table Library, so we first load this package.

```
 ───────────────────────────── Example ─────────────────────────────
  gap> LoadPackage( "ctbllib", false );
  true
```

9.1   Some **GAP** Utilities

The function `SetOfComposedClassFusions` takes two list of class fusions, where the first list consists of fusions between the character tables of the groups *H* and *G*, say, and the second list consists of class fusions between the character tables of the groups *U* and *H*, say; the return value is the set of compositions of each map in the first list with each map in the second list (via `CompositionMaps` (**Reference: CompositionMaps**)).

Note that the returned list may be a proper subset of the set of all possible class fusions between *U* and *G*, which can be computed with `PossibleClassFusions` (**Reference: PossibleClassFusions**).

```
 ───────────────────────────── Example ─────────────────────────────
  gap> SetOfComposedClassFusions:= function( hfusg, ufush )
  >     local result, map1, map2;
  >     result:= [];;
  >     for map2 in hfusg do
  >       for map1 in ufush do
```

```
>          AddSet( result, CompositionMaps( map2, map1 ) );
>        od;
>     od;
>     return result;
> end;;
```

9.2   Fusions Determined by Factorization through Intermediate Subgroups

This situation clearly occurs only for nonmaximal subgroups. Interesting examples are Sylow normalizers.

9.2.1 $Co_3N5 \to Co_3$ (September 2002)

Let H be the Sylow 5 normalizer in the sporadic simple group Co_3. The class fusion of H into Co_3 is not uniquely determined by the character tables of the two groups.

```
——————————————— Example ———————————————
  gap> co3:= CharacterTable( "Co3" );
  CharacterTable( "Co3" )
  gap> h:= CharacterTable( "Co3N5" );
  CharacterTable( "5^(1+2):(24:2)" )
  gap> hfusco3:= PossibleClassFusions( h, co3 );;
  gap> Length( RepresentativesFusions( h, hfusco3, co3 ) );
  2
```

As H is not maximal in Co_3, we look at those maximal subgroups of Co_3 whose order is divisible by that of H.

```
——————————————— Example ———————————————
  gap> mx:= Maxes( co3 );
  [ "McL.2", "HS", "U4(3).(2^2)_{133}", "M23", "3^5:(2xm11)",
    "2.S6(2)", "U3(5).3.2", "3^1+4:4s6", "2^4.a8", "L3(4).D12",
    "2xm12", "2^2.[2^7*3^2].S3", "s3xpsl(2,8).3", "a4xs5" ]
  gap> maxes:= List( mx, CharacterTable );;
  gap> filt:= Filtered( maxes, x -> Size( x ) mod Size( h ) = 0 );
  [ CharacterTable( "McL.2" ), CharacterTable( "HS" ),
    CharacterTable( "U3(5).3.2" ) ]
```

According to the Atlas (see [CCN+85, pp. 34 and 100]), H occurs as the Sylow 5 normalizer in $U_3(5).3.2$ and in $McL.2$; however, H is not a subgroup of HS, since otherwise H would be contained in subgroups of type $U_3(5).2$ (see [CCN+85, p. 80]), but the only possible subgroups in these groups are too small (see [CCN+85, p. 34]).

We compute the possible class fusions from H into $McL.2$ and from $McL.2$ to Co_3, and then form the compositions of these maps.

```
——————————————— Example ———————————————
  gap> max:= filt[1];;
  gap> hfusmax:= PossibleClassFusions( h, max );;
  gap> maxfusco3:= PossibleClassFusions( max, co3 );;
  gap> comp:= SetOfComposedClassFusions( maxfusco3, hfusmax );;
```

```
gap> Length( comp );
2
gap> reps:= RepresentativesFusions( h, comp, co3 );
[ [ 1, 2, 3, 4, 8, 8, 7, 9, 10, 11, 17, 17, 19, 19, 22, 23, 27, 27,
      30, 33, 34, 40, 40, 40, 40, 42 ] ]
```

So factoring through a maximal subgroup of type *McL*.2 determines the fusion from *H* to *Co*$_3$ uniquely up to table automorphisms.

Alternatively, we can use the group $U_3(5).3.2$ as intermediate subgroup, which leads to the same result.

```
——————————————— Example ———————————————
gap> max:= filt[3];;
gap> hfusmax:= PossibleClassFusions( h, max );;
gap> maxfusco3:= PossibleClassFusions( max, co3 );;
gap> comp:= SetOfComposedClassFusions( maxfusco3, hfusmax );;
gap> reps2:= RepresentativesFusions( h, comp, co3 );;
gap> reps2 = reps;
true
```

Finally, we compare the result with the map that is stored on the library table of *H*.

```
——————————————— Example ———————————————
gap> GetFusionMap( h, co3 ) in reps;
true
```

9.2.2 $31 : 15 \to B$ (March 2003)

The Sylow 31 normalizer *H* in the sporadic simple group *B* has the structure 31 : 15.

```
——————————————— Example ———————————————
gap> b:= CharacterTable( "B" );;
gap> h:= CharacterTable( "31:15" );;
gap> hfusb:= PossibleClassFusions( h, b );;
gap> Length( RepresentativesFusions( h, hfusb, b ) );
2
```

We determine the correct fusion using the fact that *H* is contained in a (maximal) subgroup of type *Th* in *B*.

```
——————————————— Example ———————————————
gap> th:= CharacterTable( "Th" );;
gap> hfusth:= PossibleClassFusions( h, th );;
gap> thfusb:= PossibleClassFusions( th, b );;
gap> comp:= SetOfComposedClassFusions( thfusb, hfusth );;
gap> Length( comp );
2
gap> reps:= RepresentativesFusions( h, comp, b );
[ [ 1, 145, 146, 82, 82, 19, 82, 7, 19, 82, 82, 19, 7, 82, 19, 82, 82
      ] ]
gap> GetFusionMap( h, b ) in reps;
true
```

9.2.3   *SuzN3 → Suz* **(September 2002)**

The class fusion from the Sylow 3 normalizer into the sporadic simple group *Suz* is not uniquely determined by the character tables of these groups.

```
─────────────────────────────── Example ───────────────────────────────
  gap> h:= CharacterTable( "SuzN3" );
  CharacterTable( "3^5:(3^2:SD16)" )
  gap> suz:= CharacterTable( "Suz" );
  CharacterTable( "Suz" )
  gap> hfussuz:= PossibleClassFusions( h, suz );;
  gap> Length( RepresentativesFusions( h, hfussuz, suz ) );
  2
```

Since *H* is not maximal in *Suz*, we try to factorize the fusion through a suitable maximal subgroup.

```
─────────────────────────────── Example ───────────────────────────────
  gap> maxes:= List( Maxes( suz ), CharacterTable );;
  gap> filt:= Filtered( maxes, x -> Size( x ) mod Size( h ) = 0 );
  [ CharacterTable( "3_2.U4(3).2_3'" ), CharacterTable( "3^5:M11" ),
    CharacterTable( "3^2+4:2(2^2xa4)2" ) ]
```

The group $3_2.U_4(3).2'_3$ does not admit a fusion from *H*.

```
─────────────────────────────── Example ───────────────────────────────
  gap> PossibleClassFusions( h, filt[1] );
  [ ]
```

Definitely $3^5 : M_{11}$ contains a group isomorphic with *H*, because the Sylow 3 normalizer in M_{11} has the structure $3^2 : SD_{16}$; using $3^{2+4} : 2(2^2 \times A_4)2$ would lead to the same result as we get below. We compute the compositions of possible class fusions.

```
─────────────────────────────── Example ───────────────────────────────
  gap> max:= filt[2];;
  gap> hfusmax:= PossibleClassFusions( h, max );;
  gap> maxfussuz:= PossibleClassFusions( max, suz );;
  gap> comp:= SetOfComposedClassFusions( maxfussuz, hfusmax );;
  gap> repr:= RepresentativesFusions( h, comp, suz );
  [ [ 1, 2, 2, 4, 5, 4, 5, 5, 5, 5, 5, 6, 9, 9, 14, 15, 13, 16, 16, 14,
        15, 13, 13, 13, 16, 15, 14, 16, 16, 16, 21, 21, 23, 22, 29, 29,
        29, 38, 39 ] ]
```

So the factorization determines the fusion map up to table automorphisms. We check that this map is equal to the stored one.

```
─────────────────────────────── Example ───────────────────────────────
  gap> GetFusionMap( h, suz ) in repr;
  true
```

9.2.4 $F_{3+}N5 \to F_{3+}$ (**March 2002**)

The class fusion from the table of the Sylow 5 normalizer H in the sporadic simple group F_{3+} into F_{3+} is ambiguous.

```
─────────────────────── Example ───────────────────────
 gap> f3p:= CharacterTable( "F3+" );;
 gap> h:= CharacterTable( "F3+N5" );;
 gap> hfusf3p:= PossibleClassFusions( h, f3p );;
 gap> Length( RepresentativesFusions( h, hfusf3p, f3p ) );
 2
```

H is not maximal in F_{3+}, so we look for tables of maximal subgroups that can contain H.

```
─────────────────────── Example ───────────────────────
 gap> maxes:= List( Maxes( f3p ), CharacterTable );;
 gap> filt:= Filtered( maxes, x -> Size( x ) mod Size( h ) = 0 );
 [ CharacterTable( "Fi23" ), CharacterTable( "2.Fi22.2" ),
   CharacterTable( "(3xO8+(3):3):2" ), CharacterTable( "O10-(2)" ),
   CharacterTable( "(A4xO8+(2).3).2" ), CharacterTable( "He.2" ),
   CharacterTable( "F3+M14" ), CharacterTable( "(A5xA9):2" ) ]
 gap> possfus:= List( filt, x -> PossibleClassFusions( h, x ) );
 [ [ ], [ ], [ ], [ ],
   [ [ 1, 69, 110, 12, 80, 121, 4, 72, 113, 11, 11, 79, 79, 120, 120,
           3, 71, 11, 79, 23, 91, 112, 120, 132, 29, 32, 97, 100, 37,
           37, 105, 105, 139, 140, 145, 146, 155, 155, 156, 156, 44,
           44, 167, 167, 48, 48, 171, 171, 57, 57, 180, 180, 66, 66,
           189, 189 ],
       [ 1, 69, 110, 12, 80, 121, 4, 72, 113, 11, 11, 79, 79, 120,
           120, 3, 71, 11, 79, 23, 91, 112, 120, 132, 29, 32, 97, 100,
           37, 37, 105, 105, 140, 139, 146, 145, 156, 156, 155, 155,
           44, 44, 167, 167, 48, 48, 171, 171, 57, 57, 180, 180, 66,
           66, 189, 189 ] ], [ ], [ ], [ ] ]
```

We see that from the eight possible classes of maximal subgroups in F_{3+} that might contain H, only the group of type $(A_4 \times O_8^+(2).3).2$ admits a class fusion from H. Hence we can compute the compositions of the possible fusions from H into this group with the possible fusions from this group into F_{3+}.

```
─────────────────────── Example ───────────────────────
 gap> max:= filt[5];
 CharacterTable( "(A4xO8+(2).3).2" )
 gap> hfusmax:= possfus[5];;
 gap> maxfusf3p:= PossibleClassFusions( max, f3p );;
 gap> comp:= SetOfComposedClassFusions( maxfusf3p, hfusmax );;
 gap> Length( comp );
 2
 gap> repr:= RepresentativesFusions( h, comp, f3p );
 [ [ 1, 2, 4, 12, 35, 54, 3, 3, 16, 9, 9, 11, 11, 40, 40, 2, 3, 9, 11,
       35, 36, 13, 40, 90, 7, 22, 19, 20, 43, 43, 50, 50, 8, 8, 23,
       23, 46, 46, 47, 47, 10, 10, 9, 9, 10, 10, 11, 11, 26, 26, 28,
       28, 67, 67, 68, 68 ] ]
```

Finally, we check whether the map stored in the table library is correct.

```
                             ─────── Example ───────
  gap> GetFusionMap( h, f3p ) in repr;
  true
```

Note that we did *not* determine the class fusion from the maximal subgroup $(A_4 \times O_8^+(2).3).2$ into F_{3+} up to table automorphisms (see Section 9.3.2 for this problem), since also the ambiguous result was enough for computing the fusion from H into F_{3+}.

9.3 Fusions Determined Using Commutative Diagrams Involving Smaller Subgroups

In each of the following examples, the class fusion of a (not necessarily maximal) subgroup M of a group G into G is determined by considering a proper subgroup U of M whose class fusion into G can be computed, perhaps using another subgroup S of G that also contains U.
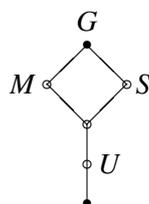
9.3.1 *BN7 → B* (**March 2002**)

Let H be a Sylow 7 normalizer in the sporadic simple group B. The class fusion of H into B is not uniquely determined by the character tables of the two groups.

```
                             ─────── Example ───────
  gap> b:= CharacterTable( "B" );
  CharacterTable( "B" )
  gap> h:= CharacterTable( "BN7" );
  CharacterTable( "BN7" )
  gap> hfusb:= PossibleClassFusions( h, b );;
  gap> Length( RepresentativesFusions( h, hfusb, b ) );
  2
```

Let us consider a maximal subgroup of the type Th in B (cf. [CCN+85, p. 217]). By [CCN+85, p. 177], the Sylow 7 normalizers in Th are maximal subgroups of Th and have the structure $7^2 : (3 \times 2S_4)$. Let U be such a subgroup.

Note that the only maximal subgroups of Th whose order is divisible by the order of a Sylow 7 subgroup of B have the types $^3D_4(2).3$ and $7^2 : (3 \times 2S_4)$, and the Sylow 7 normalizers in the former groups have the structure $7^2 : (3 \times 2A_4)$, cf. [CCN+85, p. 89].

```
                             ─────── Example ───────
  gap> Number( Factors( Size( b ) ), x -> x = 7 );
  2
  gap> th:= CharacterTable( "Th" );
  CharacterTable( "Th" )
```

```
gap> Filtered( Maxes( th ), x -> Size( CharacterTable( x ) ) mod 7^2 = 0 );
[ "3D4(2).3", "7^2:(3x2S4)" ]
```

The class fusion of *U* into *B* via *Th* is uniquely determined by the character tables of these groups.

```
                         ——— Example ———
gap> thn7:= CharacterTable( "ThN7" );
CharacterTable( "7^2:(3x2S4)" )
gap> comp:= SetOfComposedClassFusions( PossibleClassFusions( th, b ),
>                 PossibleClassFusions( thn7, th ) );
[ [ 1, 31, 7, 7, 5, 28, 28, 17, 72, 72, 6, 6, 7, 28, 27, 27, 109,
      109, 17, 45, 45, 72, 72, 127, 127, 127, 127 ] ]
```

The condition that the class fusion of *U* into *B* factors through *H* determines the class fusion of *H* into *B* up to table automorphisms.

```
                         ——— Example ———
gap> thn7fush:= PossibleClassFusions( thn7, h );;
gap> filt:= Filtered( hfusb, x ->
>                 ForAny( thn7fush, y -> CompositionMaps( x, y ) in comp ) );;
gap> Length( RepresentativesFusions( h, filt, b ) );
1
```

Finally, we compare the result with the map that is stored on the library table of *H*.

```
                         ——— Example ———
gap> GetFusionMap( h, b ) in filt;
true
```

9.3.2 $(A_4 \times O_8^+(2).3).2 \rightarrow Fi_{24}'$ (**November 2002**)

The class fusion of the maximal subgroup $M \cong (A_4 \times O_8^+(2).3).2$ of $G = Fi_{24}'$ is ambiguous.

```
                         ——— Example ———
gap> m:= CharacterTable( "(A4xO8+(2).3).2" );;
gap> t:= CharacterTable( "F3+" );;
gap> mfust:= PossibleClassFusions( m, t );;
gap> repr:= RepresentativesFusions( m, mfust, t );;
gap> Length( repr );
2
```

We first observe that the elements of order three in the normal subgroup of type A_4 in *M* lie in the class 3A of Fi_{24}'.

```
                         ——— Example ———
gap> a4inm:= Filtered( ClassPositionsOfNormalSubgroups( m ),
>                      n -> Sum( SizesConjugacyClasses( m ){ n } ) = 12 );
[ [ 1, 69, 110 ] ]
gap> OrdersClassRepresentatives( m ){ a4inm[1] };
[ 1, 2, 3 ]
gap> List( repr, map -> map[110] );
[ 4, 4 ]
gap> OrdersClassRepresentatives( t ){ [ 1 .. 4 ] };
[ 1, 2, 2, 3 ]
```

Let us take one such element g, say. Its normalizer S in G has the structure $(3 \times O_8^+(3).3).2$; this group is maximal in G, and its character table is available in GAP.

```
───── Example ─────
gap> s:= CharacterTable( "F3+N3A" );
CharacterTable( "(3xO8+(3):3):2" )
```

The intersection $N_M(g) = S \cap M$ contains a subgroup U of the type $3 \times O_8^+(2).3$, and in the following we compute the class fusions of U into S and M, and then utilize the fact that only those class fusions from M into G are possible whose composition with the class fusion from U into M equals a composition of class fusions from U into S and from S into G.

```
───── Example ─────
gap> u:= CharacterTable( "Cyclic", 3 ) * CharacterTable( "O8+(2).3" );
CharacterTable( "C3xO8+(2).3" )
gap> ufuss:= PossibleClassFusions( u, s );;
gap> ufusm:= PossibleClassFusions( u, m );;
gap> sfust:= PossibleClassFusions( s, t );;
gap> comp:= SetOfComposedClassFusions( sfust, ufuss );;
gap> Length( comp );
6
gap> filt:= Filtered( mfust,
>      x -> ForAny( ufusm, map -> CompositionMaps( x, map ) in comp ) );;
gap> repr:= RepresentativesFusions( m, filt, t );;
gap> Length( repr );
1
gap> GetFusionMap( m, t ) in repr;
true
```

So the class fusion from M into G is determined up to table automorphisms by the commutative diagram.

9.3.3 $A_6 \times L_2(8).3 \to Fi'_{24}$ (**November 2002**)

The class fusion of the maximal subgroup $M \cong A_6 \times L_2(8).3$ of $G = Fi'_{24}$ is ambiguous.

```
───── Example ─────
gap> m:= CharacterTable( "A6xL2(8):3" );;
gap> t:= CharacterTable( "F3+" );;
gap> mfust:= PossibleClassFusions( m, t );;
gap> Length( RepresentativesFusions( m, mfust, t ) );
2
```

We will use the fact that the direct factor of the type A_6 in M contains elements in the class 3A of G. This fact can be shown as follows.

```
───── Example ─────
gap> dppos:= ClassPositionsOfDirectProductDecompositions( m );
[ [ [ 1, 12 .. 67 ], [ 1 .. 11 ] ] ]
gap> List( dppos[1], l -> Sum( SizesConjugacyClasses( t ){ l } ) );
[ 17733424133316996808705, 4545066196775803392 ]
gap> List( dppos[1], l -> Sum( SizesConjugacyClasses( m ){ l } ) );
[ 360, 1512 ]
```

```
gap> 3Apos:= Position( OrdersClassRepresentatives( t ), 3 );
4
gap> 3Ainm:= List( mfust, map -> Position( map, 3Apos ) );
[ 23, 23, 23, 23, 34, 34, 34, 34 ]
gap> ForAll( 3Ainm, x -> x in dppos[1][1] );
true
```

Since the normalizer of an element of order three in A_6 has the form $3^2 : 2$, such a 3A element in M contains a subgroup U of the structure $3^2 : 2 \times L_2(8).3$ which is contained in the 3A normalizer S in G, which has the structure $(3 \times O_8^+(3).3).2$.

(Note that all classes in the $3^2 : 2$ type group are rational, and its character table is available in the GAP Character Table Library with the identifier "3^2:2".)

```
─────────────────────────── Example ───────────────────────────
gap> u:= CharacterTable( "3^2:2" ) * CharacterTable( "L2(8).3" );
CharacterTable( "3^2:2xL2(8).3" )
gap> s:= CharacterTable( "F3+N3A" );
CharacterTable( "(3xO8+(3):3):2" )
gap> ufuss:= PossibleClassFusions( u, s );;
gap> comp:= SetOfComposedClassFusions( sfust, ufuss );;
gap> ufusm:= PossibleClassFusions( u, m );;
gap> filt:= Filtered( mfust,
>                map -> ForAny( ufusm,
>                          map2 -> CompositionMaps( map, map2 ) in comp ) );;
gap> repr:= RepresentativesFusions( m, filt, t );;
gap> Length( repr );
1
gap> GetFusionMap( m, t ) in repr;
true
```

9.3.4 $(3^2 : D_8 \times U_4(3).2^2).2 \to B$ **(June 2007)**

Let G be a maximal subgroup of the type $(3^2 : D_8 \times U_4(3).2^2).2$ in the sporadic simple group B, cf. [CCN$^+$85, p. 217]. Computing the class fusion of G into B just from the character tables of the two groups takes extremely long. So we use additional information.

According to [CCN$^+$85, p. 217], G is the normalizer in B of an elementary abelian group $\langle x, y \rangle$ of order 9, with x, y in the class 3A of B, and $N = N_B(\langle x \rangle)$ has the structure $S_3 \times Fi_{22}.2$. The intersection $G \cap N$ has the structure $S_3 \times S_3 \times U_4(3).2^2$, which is the direct product of S_3 and the normalizer in $Fi_{22}.2$ of a 3A element of $Fi_{22}.2$, see [CCN$^+$85, p. 163]. Thus we may use that the class fusions from $G \cap N$ into B through G or N coincide.

The class fusion from N into B is uniquely determined by the character tables.

```
─────────────────────────── Example ───────────────────────────
gap> b:= CharacterTable( "B" );;
gap> n:= CharacterTable( "BN3A" );
CharacterTable( "S3xFi22.2" )
gap> nfusb:= PossibleClassFusions( n, b );;
gap> Length( nfusb );
1
gap> nfusb:= nfusb[1];;
```

The computation of the class fusion from $G \cap N$ into N is sped up by computing first the class fusion modulo the direct factor S_3, and then lifting these fusion maps.

```
─────────── Example ───────────
gap> fi222:= CharacterTable( "Fi22.2" );;
gap> fi222n3a:= CharacterTable( "S3xU4(3).(2^2)_{122}" );;
gap> s3:= CharacterTable( "S3" );;
gap> inter:= s3 * fi222n3a;;
gap> intermods3fusnmods3:= PossibleClassFusions( fi222n3a, fi222 );;
gap> Length( intermods3fusnmods3 );
2
gap> Length( RepresentativesFusions( fi222n3a, intermods3fusnmods3, fi222 ) );
1
```

We get two equivalent possibilities, and need to consider only one of them. For lifting it to a map between $G \cap N$ and N, the safe way is to use the fusion map between the two factors for computing an approximation. (Additionally, we could interpret the known maps as fusions between two subgroups, and use this for improving the approximation, but in this case the speedup is not worth the effort.)

```
─────────── Example ───────────
gap> interfusn:= CompositionMaps( InverseMap( GetFusionMap( n, fi222 ) ),
>        CompositionMaps( intermods3fusnmods3[1],
>            GetFusionMap( inter, fi222n3a ) ) );;
gap> interfusn:= PossibleClassFusions( inter, n,
>        rec( fusionmap:= interfusn, quick:= true ) );;
gap> Length( interfusn );
1
```

The lift is unique. Since we lift a class fusion to direct products, we could also "extend" the fusion directly. But note that this would assume the ordering of classes in character tables of direct products. This alternative would work as follows.

```
─────────── Example ───────────
gap> nccl:= NrConjugacyClasses( fi222 );;
gap> interfusn[1] = Concatenation( List( [ 0 .. 2 ],
>                        i -> intermods3fusnmods3[1] + i * nccl ) );
true
```

Next we compute the class fusions from $G \cap N$ to G. We get two equivalent solutions.

```
─────────── Example ───────────
gap> tblg:= CharacterTable( "BM14" );
CharacterTable( "(3^2:D8xU4(3).2^2).2" )
gap> interfusg:= PossibleClassFusions( inter, tblg );;
gap> Length( interfusg );
2
gap> Length( RepresentativesFusions( inter, interfusg, tblg ) );
1
```

The approximation of the class fusion from G to B is computed by composing the known maps. Because we have chosen one of the two possible maps from $G \cap N$ to N, here we consider the two possibilities. From these approximations, we compute the possible class fusions.

```
——————————————— Example ———————————————
gap> interfusb:= CompositionMaps( nfusb, interfusn[1] );;
gap> approx:= List( interfusg,
>       map -> CompositionMaps( interfusb, InverseMap( map ) ) );;
gap> gfusb:= Set( Concatenation( List( approx,
>                 map -> PossibleClassFusions( tblg, b,
>                             rec( fusionmap:= map ) ) ) ) );;
gap> Length( gfusb );
4
gap> Length( RepresentativesFusions( tblg, gfusb, b ) );
1
```

Finally, we compare the result with the class fusion that is stored on the library table.

```
——————————————— Example ———————————————
gap> GetFusionMap( tblg, b ) in gfusb;
true
```

9.3.5 $7^{1+4} : (3 \times 2.S_7) \to M$ (May 2009)

The class fusion of the maximal subgroup U of type $7^{1+4} : (3 \times 2.S_7)$ of the Monster group M into M is ambiguous.

```
——————————————— Example ———————————————
gap> tblu:= CharacterTable( "7^(1+4):(3x2.S7)" );;
gap> m:= CharacterTable( "M" );;
gap> ufusm:= PossibleClassFusions( tblu, m );;
gap> Length( RepresentativesFusions( tblu, ufusm, m ) );
2
```

The subgroup U contains a Sylow 7–subgroup of M, and the only maximal subgroups of M with this property are the class of U and another class of subgroups, of the type $7^{2+1+2} : GL_2(7)$. Moreover, it turns out that the Sylow 7 normalizers in the subgroups in both classes have the same order, hence they are the Sylow 7 normalizers in M.

For that, we use representations from the Atlas of Group Representations [WWT+], and access these representations via the GAP package AtlasRep ([WPN+22]).

```
——————————————— Example ———————————————
gap> LoadPackage( "atlasrep", false );
true
gap> g1:= AtlasGroup( "7^(2+1+2):GL2(7)" );;
gap> s1:= SylowSubgroup( g1, 7 );;
gap> n1:= Normalizer( g1, s1 );;
gap> g2:= AtlasGroup( "7^(1+4):(3x2.S7)" );;
gap> s2:= SylowSubgroup( g2, 7 );;
gap> n2:= Normalizer( g2, s2 );;
gap> Size( n1 ) = Size( n2 );
true
gap> ( Size( m ) / Size( s1 ) ) mod 7 <> 0;
true
```

So let N be a Sylow 7 normalizer in U, and choose a subgroup S of the type $7^{2+1+2} : GL_2(7)$ that contains N.

We compute the character table of N. Computing the possible class fusions of N into M directly yields two possibilities, but the class fusion of N into M via S is uniquely determined by the character tables.

```
——————— Example ———————
gap> tbln:= CharacterTable( Image( IsomorphismPcGroup( n1 ) ) );;
gap> tbls:= CharacterTable( "7^(2+1+2):GL2(7)" );;
gap> nfusm:= PossibleClassFusions( tbln, m );;
gap> Length( RepresentativesFusions( tbln, nfusm, m ) );
2
gap> nfuss:= PossibleClassFusions( tbln, tbls );;
gap> sfusm:= PossibleClassFusions( tbls, m );;
gap> nfusm:= SetOfComposedClassFusions( sfusm, nfuss );;
gap> Length( nfusm );
1
```

Now we use the condition that the class fusions from N into M factors through U. This determines the class fusion of U into M up to table automorphisms.

```
——————— Example ———————
gap> nfusu:= PossibleClassFusions( tbln, tblu );;
gap> ufusm:= Filtered( ufusm, map2 -> ForAny( nfusu,
>       map1 -> CompositionMaps( map2, map1 ) in nfusm ) );;
gap> Length( RepresentativesFusions( tblu, ufusm, m ) );
1
```

Let C be the centralizer in U of the normal subgroup of order 7; note that C is the 7B centralizer on M. We can use the information about the class fusion of U into M for determining the class fusion of C into M. The class fusion of C into M is not determined by the character tables, but the class fusion of C into U is determined up to table automorphisms, so the same holds for the class fusion of C into M.

```
——————— Example ———————
gap> tblc:= CharacterTable( "MC7B" );
CharacterTable( "7^1+4.2A7" )
gap> cfusm:= PossibleClassFusions( tblc, m );;
gap> Length( RepresentativesFusions( tblc, cfusm, m ) );
2
gap> cfusu:= PossibleClassFusions( tblc, tblu );;
gap> cfusm:= SetOfComposedClassFusions( ufusm, cfusu );;
gap> Length( RepresentativesFusions( tblc, cfusm, m ) );
1
```

9.3.6 $3^7.O_7(3) : 2 \to Fi_{24}$ (November 2010)

The class fusion of the maximal subgroup $M \cong 3^7.O_7(3) : 2$ of $G = Fi_{24} = F_{3+}.2$ is ambiguous.

```
——————— Example ———————
gap> m:= CharacterTable( "3^7.O7(3):2" );;
gap> t:= CharacterTable( "F3+.2" );;
gap> mfust:= PossibleClassFusions( m, t );;
gap> Length( RepresentativesFusions( m, mfust, t ) );
2
```

We will use the fact that the elementary abelian normal subgroup of order 3^7 in M contains an element x, say, in the class 3A of G. This fact can be shown as follows.

```
──────── Example ────────
gap> nsg:= ClassPositionsOfNormalSubgroups( m );
[ [ 1 ], [ 1 .. 4 ], [ 1 .. 158 ], [ 1 .. 291 ] ]
gap> Sum( SizesConjugacyClasses( m ){ nsg[2] } );
2187
gap> 3^7;
2187
gap> rest:= Set( mfust, map -> map{ nsg[2] } );
[ [ 1, 4, 5, 6 ] ]
gap> List( rest, l -> ClassNames( t, "Atlas" ){ l } );
[ [ "1A", "3A", "3B", "3C" ] ]
```

The normalizer S of $\langle x \rangle$ in G has the form $S_3 \times O_8^+(3):S_3$, and the order of $U = S \cap M = N_M(\langle x \rangle)$ is 53059069440, so U has index 3360 in S.

```
──────── Example ────────
gap> s:= CharacterTable( "F3+.2N3A" );
CharacterTable( "S3xO8+(3):S3" )
gap> PowerMap( m, 2 )[4];
4
gap> size_u:= 2 * SizesCentralizers( m )[ 2 ];
53059069440
gap> Size( s ) / size_u;
3360
```

Using the list of maximal subgroups of $O_8^+(3)$, we see that only the maximal subgroups of the type $3^6 : L_4(3)$ have index dividing 3360 in $O_8^+(3)$. (There are three classes of such subgroups.) This implies that U contains a subgroup of the type $S_3 \times 3^6 : L_4(3)$.

```
──────── Example ────────
gap> o8p3:= CharacterTable( "O8+(3)" );;
gap> mx:= List( Maxes( o8p3 ), CharacterTable );;
gap> filt:= Filtered( mx, x -> 3360 mod Index( o8p3, x ) = 0 );
[ CharacterTable( "3^6:L4(3)" ), CharacterTable( "O8+(3)M8" ),
  CharacterTable( "O8+(3)M9" ) ]
gap> List( filt, x -> Index( o8p3, x ) );
[ 1120, 1120, 1120 ]
```

We compute the possible class fusions from U into M and S in two steps, because this is faster. First the possible class fusions from $U'' \cong 3^6 : L_4(3)$ into M and S are computed, and then these fusions are used to derive approximations for the fusions from U into M and S.

```
──────── Example ────────
gap> uu:= filt[1];;
gap> u:= CharacterTable( "Symmetric", 3 ) * uu;
CharacterTable( "Sym(3)x3^6:L4(3)" )
gap> uufusm:= PossibleClassFusions( uu, m );;
gap> Length( uufusm );
8
gap> approx:= List( uufusm, map -> CompositionMaps( map,
```

```
>                     InverseMap( GetFusionMap( uu, u ) ) ) );;
gap> ufusm:= Concatenation( List( approx, map ->
>         PossibleClassFusions( u, m, rec( fusionmap:= map ) ) ) );;
gap> Length( ufusm );
8
gap> uufuss:= PossibleClassFusions( uu, s );;
gap> Length( uufuss );
8
gap> approx:= List( uufuss, map -> CompositionMaps( map,
>                InverseMap( GetFusionMap( uu, u ) ) ) );;
gap> ufuss:= Concatenation( List( approx, map ->
>   PossibleClassFusions( u, s, rec( fusionmap:= map ) ) ) );;
gap> Length( ufuss );
8
```

Now we compute the possible class fusions from *S* into *G*, and the compositions of these maps with the possible class fusions from *U* into *S*.

```
─────────────────────────── Example ───────────────────────────
gap> sfust:= PossibleClassFusions( s, t );;
gap> comp:= SetOfComposedClassFusions( sfust, ufuss );;
gap> Length( comp );
8
```

It turns out that only one orbit of the possible class fusions from *M* to *G* is compatible with these possible class fusions from *U* to *G*.

```
─────────────────────────── Example ───────────────────────────
gap> filt:= Filtered( mfust, map2 -> ForAny( ufusm, map1 ->
>         CompositionMaps( map2, map1 ) in comp ) );;
gap> Length( filt );
4
gap> Length( RepresentativesFusions( m, filt, t ) );
1
```

The class fusion stored in the GAP Character Table Library is one of them.

```
─────────────────────────── Example ───────────────────────────
gap> GetFusionMap( m, t ) in filt;
true
```

9.3.7 $^2E_6(2)N3C \rightarrow {}^2E_6(2)$ (**January 2019**)

Let $G = {}^2E_6(2)$, and $g \in G$ in the conjugacy class 3C. Using a permutation representation of *G*, Frank Lübeck has computed a representation and the character table of the maximal subgroup $N = N_G(\langle g \rangle)$ of *G*.

```
─────────────────────────── Example ───────────────────────────
gap> t:= CharacterTable( "2E6(2)" );;
gap> pos3CinG:= Position( ClassNames( t ), "3c" );
7
gap> n:= CharacterTable( "2E6(2)N3C" );;
gap> nclasses:= SizesConjugacyClasses( n );;
```

```
gap> pos3CinN:= Filtered( [ 1 .. NrConjugacyClasses( n ) ],
>                          i -> nclasses[i] = 2 );
[ 2 ]
gap> nfust:= PossibleClassFusions( n, t );;
gap> ForAll( nfust, x -> x[ pos3CinN[1] ] = pos3CinG );
true
gap> Size( n ) = 2 * SizesCentralizers( t )[ pos3CinG ];
true
gap> ForAll( Irr( n ), x -> IsInt( x[ pos3CinN[1] ] ) );
true
```

The class fusion of N in G is ambiguous.

```
─────────────── Example ───────────────
gap> rep:= RepresentativesFusions( n, nfust, t );;
gap> Length( rep );
4
```

We use the fact that g is contained in a subgroup $S \cong Fi_{22}$ of G, ...

```
─────────────── Example ───────────────
gap> s:= CharacterTable( "Fi22" );;
gap> sfust:= PossibleClassFusions( s, t );;
gap> ForAll( sfust, x -> x[6] = pos3CinG );
true
gap> pos3CinS:= 6;;
```

... and that $U = N_S(\langle g \rangle) \cong 3^{1+6} : 2^{3+4} : 3^2 : 2$ is a maximal subgroup of S whose character table is available. Thus $U \leq N$, of index four.

```
─────────────── Example ───────────────
gap> u:= CharacterTable( Maxes( s )[11] );
CharacterTable( "3^(1+6):2^(3+4):3^2:2" )
gap> uclasses:= SizesConjugacyClasses( u );;
gap> pos3CinU:= Filtered( [ 1 .. NrConjugacyClasses( u ) ],
>                          i -> uclasses[i] = 2 );
[ 2 ]
gap> ufuss:= PossibleClassFusions( u, s );;
gap> ForAll( ufuss, x -> x[ pos3CinU[1] ] = pos3CinS );
true
gap> Size( n ) / Size( u );
4
```

Composing the class fusions of U in N and N in G must be equal to the composition of the class fusions of U in S and S in G. This reduces the number of candidates for the fusion of N in G from four to two.

```
─────────────── Example ───────────────
gap> ufusn:= PossibleClassFusions( u, n );;
gap> comp:= SetOfComposedClassFusions( sfust, ufuss );;
gap> good:= Filtered( nfust, map2 -> ForAny( ufusn,
>              map1 -> CompositionMaps( map2, map1 ) in comp ) );;
gap> Length( good );
1728
```

```
gap> goodrep:= RepresentativesFusions( n, good, t );;
gap> Length( goodrep );
2
```

Next we use the fact that g and thus N is invariant under an outer automorphism α, say, of order three of G. Note that such an automorphism acts nontrivially on the conjugacy classes of G, for example because the class fusion of G into $G.3 = \langle G, \alpha \rangle$ shows the existence of orbits of length three, and that the permutation action of α on the classes of G is given by the unique subgroup of order three in the group of table automorphisms of G.

```
                                ─── Example ───
  gap> tfust3:= GetFusionMap( t, CharacterTable( "2E6(2).3" ) );;
  gap> Number( InverseMap( tfust3 ), IsList );
  14
  gap> autt:= AutomorphismsOfTable( t );;
  gap> ord3:= Filtered( autt, x -> Order( x ) = 3 );;
  gap> Length( ord3 );
  2
  gap> alpha:= ord3[1];;
  gap> pos3CinG ^ alpha = pos3CinG;
  true
```

The character table of N has 26 table automorphisms of order three. We do not know which of them (or perhaps the identity permutation) is induced by the restriction α_N of α to N, but the embedding $\iota : N \to G$ satisfies $\alpha \circ \iota = \iota \circ \alpha_N$, and we can check each fusion candidate for the existence of a candidate for α_N such that this relation holds.

```
                                ─── Example ───
  gap> autn:= AutomorphismsOfTable( n );;
  gap> ord3:= Filtered( autn, x -> Order( x ) = 3 );;
  gap> Length( ord3 );
  26
  gap> Add( ord3, () );
  gap> filt:= Filtered( rep, map -> ForAny( ord3, beta ->
  > OnTuples( map, alpha ) = Permuted( map, beta ) ) );;
  gap> Length( filt );
  2
```

Again, the number of candidates for the fusion of N in G is reduced from four to two. Moreover, we are lucky because only one candidate satifies also the first criterion we have checked.

```
                                ─── Example ───
  gap> inter:= Intersection( good, filt );
  [ [ 1, 7, 5, 6, 7, 2, 3, 4, 27, 30, 24, 32, 25, 26, 9, 11, 12, 13,
        10, 14, 19, 19, 19, 16, 17, 18, 21, 58, 61, 62, 67, 68, 69, 57,
        72, 59, 75, 76, 77, 78, 79, 80, 64, 65, 66, 60, 81, 82, 5, 6,
        7, 6, 7, 7, 7, 7, 6, 7, 6, 7, 7, 24, 25, 27, 26, 28, 30, 29,
        31, 32, 31, 32, 32, 32, 32, 31, 32, 31, 32, 51, 52, 52, 52, 52,
        74, 76, 77, 77, 75, 74, 76, 74, 75, 99, 100, 101, 102, 4, 20,
        29, 31, 32, 36, 36, 42, 42, 39, 40, 41, 49, 49, 49, 49, 49, 49,
        71, 112, 112, 114, 115, 116 ] ]
```

The class fusion stored in the **GAP** Character Table Library is this candidate.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> GetFusionMap( n, t ) = inter[1];
true
```

Remark:

Note that the structure of N is $3^{1+6} : 2^{3+6} : 3^2 : 2$, as is stated in [Nor]. The structure $3^{1+6}.2^{3+6}.(S_3 \times 3)$ claimed in the Atlas [CCN+85, p. 191] is wrong, as we can read off for example from the fact that N has exactly two linear characters.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> Length( LinearCharacters( n ) );
2
```

9.4 Fusions Determined Using Commutative Diagrams Involving Factor Groups

9.4.1 $3.A_7 \rightarrow 3.Suz$ (December 2010)

The maximal subgroups of type A_7 in the sporadic simple Suzuki group Suz lift to groups of the type $3.A_7$ in $3.Suz$. This can be seen from the fact that $3.Suz$ does not admit a class fusion from A_7.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> t:= CharacterTable( "Suz" );;
gap> 3t:= CharacterTable( "3.Suz" );;
gap> s:= CharacterTable( "A7" );;
gap> 3s:= CharacterTable( "3.A7" );;
gap> PossibleClassFusions( s, 3t );
[  ]
```

The class fusion of $3.A_7$ into $3.Suz$ is ambiguous.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> 3sfus3t:= PossibleClassFusions( 3s, 3t );;
gap> Length( 3sfus3t );
6
gap> RepresentativesFusions( 3s, 3sfus3t, 3t );
[ [ 1, 2, 3, 7, 8, 9, 16, 16, 26, 27, 28, 32, 33, 34, 47, 47, 47, 48,
      49, 50, 48, 49, 50 ],
  [ 1, 11, 12, 4, 36, 37, 13, 16, 23, 82, 83, 32, 100, 101, 44, 38,
      41, 48, 112, 116, 48, 115, 113 ] ]
gap> ClassPositionsOfCentre( 3t );
[ 1, 2, 3 ]
```

We see that the possible fusions in the second orbit avoid the centre of $3.Suz$. Since the preimages in $3.Suz$ of the A_7 type subgroups of Suz contain the centre of $3.Suz$, we know that the class fusion of these preimages belong to the first orbit. This can be formalized by checking the commutativity of the diagram of fusions between $3.A_7$, $3.Suz$, and their factors A_7 and Suz.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> sfust:= PossibleClassFusions( s, t );;
gap> Length( sfust );
```

```
  1
gap> filt:= Filtered( 3sfus3t, map -> CompositionMaps( GetFusionMap( 3t, t ),
>                                     map )
>              = CompositionMaps( sfust[1], GetFusionMap( 3s, s ) ) );
[ [ 1, 2, 3, 7, 8, 9, 16, 16, 26, 27, 28, 32, 33, 34, 47, 47, 47, 48,
      49, 50, 48, 49, 50 ],
  [ 1, 3, 2, 7, 9, 8, 16, 16, 26, 28, 27, 32, 34, 33, 47, 47, 47, 48,
      50, 49, 48, 50, 49 ] ]
```

So the class fusion of maximal $3.A_7$ type subgroups of $3.Suz$ is determined up to table automorphisms. One of these fusions is stored on the table of $3.A_7$.

```
───────────── Example ─────────────
gap> RepresentativesFusions( 3s, filt, 3t );
[ [ 1, 2, 3, 7, 8, 9, 16, 16, 26, 27, 28, 32, 33, 34, 47, 47, 47, 48,
      49, 50, 48, 49, 50 ] ]
gap> GetFusionMap( 3s, 3t ) in filt;
true
```

Also the class fusions in the other orbit belong to subgroups of type $3.A_7$ in $3.Suz$. Note that Suz contains maximal subgroups of the type $3_2.U_4(3).2'_3$ (see [CCN$^+$85, p. 131]), and the A_7 type subgroups of $U_4(3)$ (see [CCN$^+$85, p. 52]) lift to groups of the type $3.A_7$ in $3_2.U_4(3)$ because $3_2.U_4(3)$ does not admit a class fusion from A_7. The preimages in $3.Suz$ of the $3.A_7$ type subgroups of Suz have the structure $3 \times 3.A_7$.

```
───────────── Example ─────────────
gap> u:= CharacterTable( "3_2.U4(3)" );;
gap> PossibleClassFusions( s, u );
[ ]
gap> Length( PossibleClassFusions( 3s, u ) );
8
```

9.4.2 $S_6 \to U_4(2)$ (September 2011)

The simple group $G = U_4(2)$ contains a maximal subgroup U of type S_6. The class fusion from U to G is unique up to table automorphisms.

```
───────────── Example ─────────────
gap> s:= CharacterTable( "S6" );
CharacterTable( "A6.2_1" )
gap> t:= CharacterTable( "U4(2)" );
CharacterTable( "U4(2)" )
gap> sfust:= PossibleClassFusions( s, t );
[ [ 1, 3, 6, 7, 9, 10, 3, 2, 9, 16, 15 ],
  [ 1, 3, 7, 6, 9, 10, 2, 3, 9, 15, 16 ] ]
gap> Length( RepresentativesFusions( s, sfust, t ) );
1
```

In the double cover $2.G$ of G, U lifts to the double cover $2.U$ of U (which is unique up to isomorphism). Also the class fusion from $2.U$ to $2.G$ is unique up to table automorphisms.

```
                           ─── Example ───
  gap> 2t:= CharacterTable( "2.U4(2)" );
  CharacterTable( "2.U4(2)" )
  gap> 2s:= CharacterTable( "2.A6.2_1" );
  CharacterTable( "2.A6.2_1" )
  gap> 2sfus2t:= PossibleClassFusions( 2s, 2t );
  [ [ 1, 2, 4, 11, 12, 9, 10, 15, 16, 17, 3, 4, 15, 24, 25, 26, 26 ],
    [ 1, 2, 4, 11, 12, 9, 10, 15, 16, 17, 3, 4, 15, 25, 24, 26, 26 ] ]
  gap> Length( RepresentativesFusions( 2s, 2sfus2t, 2t ) );
  1
```

However, the two possible fusions from $2.U$ to $2.G$ are lifts of the same class fusion from U to G.

```
                           ─── Example ───
  gap> 2sfuss:= GetFusionMap( 2s, s );
  [ 1, 1, 2, 3, 3, 4, 4, 5, 6, 6, 7, 8, 9, 10, 10, 11, 11 ]
  gap> 2tfust:= GetFusionMap( 2t, t );;
  gap> induced:= Set( 2sfus2t, x -> CompositionMaps( 2tfust,
  >       CompositionMaps( x, InverseMap( 2sfuss ) ) ) );
  [ [ 1, 3, 7, 6, 9, 10, 2, 3, 9, 15, 16 ] ]
```

The point is that the outer automorphism of S_6 that makes the two fusions from U to G equivalent does not lift to $2.U$, and that we have silently assumed a fixed factor fusion from $2.U$ to U. Note that composing this factor fusion with the automorphism of U would also yield a factor fusion, and w. r. t. the commutative diagram involving this factor fusion, the other possible class fusion from U to G is induced by the possible fusions from $2.U$ to $2.G$.

```
                           ─── Example ───
  gap> auts:= AutomorphismsOfTable( s );
  Group([ (3,4)(7,8)(10,11) ])
  gap> other:= OnTuples( 2sfuss, GeneratorsOfGroup( auts )[1] );
  [ 1, 1, 2, 4, 4, 3, 3, 5, 6, 6, 8, 7, 9, 11, 11, 10, 10 ]
  gap> Set( 2sfus2t, x -> CompositionMaps( 2tfust,
  >       CompositionMaps( x, InverseMap( other ) ) ) );
  [ [ 1, 3, 6, 7, 9, 10, 3, 2, 9, 16, 15 ] ]
```

The library table of U stores the class fusion to G that is compatible with the stored factor fusion from $2.U$ to U.

```
                           ─── Example ───
  gap> GetFusionMap( s, t ) in induced;
  true
```

9.5 Fusions Determined Using Commutative Diagrams Involving Automorphic Extensions

9.5.1 $U_3(8).3_1 \to {}^2E_6(2)$ (December 2010)

According to the Atlas (see [CCN+85, p. 191]), the group $G = {}^2E_6(2)$ contains a maximal subgroup U of the type $U_3(8).3_1$. The class fusion of U into G is ambiguous.

```
─────────────────── Example ───────────────────
gap> s:= CharacterTable( "U3(8).3_1" );;
gap> t:= CharacterTable( "2E6(2)" );;
gap> sfust:= PossibleClassFusions( s, t );;
gap> Length( sfust );
24
gap> Length( RepresentativesFusions( s, sfust, t ) );
2
```

In the automorphic extension $G.2 = {}^2E_6(2).2$ of G, the subgroup U extends to a group $U.2$ of the type $U_3(8).6$ (again, see [CCN$^+$85, p. 191]). The class fusion of $U.2$ into $G.2$ is unique up to table automorphisms.

```
─────────────────── Example ───────────────────
gap> s2:= CharacterTable( "U3(8).6" );;
gap> t2:= CharacterTable( "2E6(2).2" );;
gap> s2fust2:= PossibleClassFusions( s2, t2 );;
gap> Length( s2fust2 );
2
gap> Length( RepresentativesFusions( s2, s2fust2, t2 ) );
1
```

Only half of the possible class fusions from U into G are compatible with the embeddings of U into $G.2$ via $U.2$ and G, and the compatible maps form one orbit under table automorphisms.

```
─────────────────── Example ───────────────────
gap> sfuss2:= PossibleClassFusions( s, s2 );;
gap> comp:= SetOfComposedClassFusions( s2fust2, sfuss2 );;
gap> tfust2:= PossibleClassFusions( t, t2 );;
gap> filt:= Filtered( sfust, map -> ForAny( tfust2,
>             map2 -> CompositionMaps( map2, map ) in comp ) );;
gap> Length( filt );
12
gap> Length( RepresentativesFusions( s, filt, t ) );
1
```

Let us see which classes of U and G are involved in the disambiguation of the class fusion. The "good" fusion candidates differ from the excluded ones on the classes at the positions 31 to 36: Under all possible class fusions, two pairs of classes are mapped to the classes 81 and 82 of G; from these classes, the excluded maps fuse classes at odd positions with classes at even positions, whereas the "good" class fusions do not have this property.

```
─────────────────── Example ───────────────────
gap> Set( filt, x -> x{ [ 31 .. 36 ] } );
[ [ 74, 74, 81, 82, 81, 82 ], [ 74, 74, 82, 81, 82, 81 ],
  [ 81, 82, 74, 74, 81, 82 ], [ 81, 82, 81, 82, 74, 74 ],
  [ 82, 81, 74, 74, 82, 81 ], [ 82, 81, 82, 81, 74, 74 ] ]
gap> Set( Difference( sfust, filt ), x -> x{ [ 31 .. 36 ] } );
[ [ 74, 74, 81, 82, 82, 81 ], [ 74, 74, 82, 81, 81, 82 ],
  [ 81, 82, 74, 74, 82, 81 ], [ 81, 82, 82, 81, 74, 74 ],
  [ 82, 81, 74, 74, 81, 82 ], [ 82, 81, 81, 82, 74, 74 ] ]
```

None of the possible class fusions from U to $U.2$ fuses classes at odd positions in the interval from 31 to 36 with classes at even positions.

```
——————————————————— Example ———————————————————
gap> Set( sfuss2, x -> x{ [ 31 .. 36 ] } );
[ [ 28, 29, 30, 31, 30, 31 ], [ 29, 28, 31, 30, 31, 30 ],
  [ 30, 31, 28, 29, 30, 31 ], [ 30, 31, 30, 31, 28, 29 ],
  [ 31, 30, 29, 28, 31, 30 ], [ 31, 30, 31, 30, 29, 28 ] ]
```

This suffices to exclude the "bad" fusion candidates because no further fusion of the relevant classes of G happens in $G.2$.

```
——————————————————— Example ———————————————————
gap> List( tfust2, x -> x{ [ 74, 81, 82 ] } );
[ [ 65, 70, 71 ], [ 65, 70, 71 ], [ 65, 71, 70 ], [ 65, 71, 70 ],
  [ 65, 70, 71 ], [ 65, 70, 71 ], [ 65, 71, 70 ], [ 65, 71, 70 ],
  [ 65, 70, 71 ], [ 65, 70, 71 ], [ 65, 71, 70 ], [ 65, 71, 70 ] ]
```

(The same holds for the fusion of the relevant classes of $U.2$ in $G.2$.)

```
——————————————————— Example ———————————————————
gap> List( s2fust2, x -> x{ [ 28 .. 31 ] } );
[ [ 65, 65, 70, 71 ], [ 65, 65, 71, 70 ] ]
```

Finally, we check that a correct map is stored on the library table.

```
——————————————————— Example ———————————————————
gap> GetFusionMap( s, t ) in filt;
true
```

9.5.2 $L_3(4).2_1 \to U_6(2)$ (December 2010)

According to the Atlas (see [CCN+85, p. 115]), the group $G = U_6(2)$ contains a maximal subgroup U of the type $L_3(4).2_1$. The class fusion of U into G is ambiguous.

```
——————————————————— Example ———————————————————
gap> s:= CharacterTable( "L3(4).2_1" );;
gap> t:= CharacterTable( "U6(2)" );;
gap> sfust:= PossibleClassFusions( s, t );;
gap> Length( sfust );
27
gap> Length( RepresentativesFusions( s, sfust, t ) );
3
```

In the automorphic extension $G.3 = U_6(2).3$ of G, the subgroup U extends to a group $U.3$ of the type $L_3(4).6$ (again, see [CCN+85, p. 115]). The class fusion of $U.3$ into $G.3$ is unique up to table automorphisms.

```
——————————————————— Example ———————————————————
gap> s3:= CharacterTable( "L3(4).6" );;
gap> t3:= CharacterTable( "U6(2).3" );;
gap> s3fust3:= PossibleClassFusions( s3, t3 );;
gap> Length( s3fust3 );
2
gap> Length( RepresentativesFusions( s3, s3fust3, t3 ) );
1
```

Here the argument used in Section 9.5.1 does not work, because all possible class fusions from U into G are compatible with the embeddings of U into $G.3$ via $U.3$ and G.

```
────────────────────── Example ──────────────────────
  gap> sfuss3:= PossibleClassFusions( s, s3 );;
  gap> comp:= SetOfComposedClassFusions( s3fust3, sfuss3 );;
  gap> tfust3:= PossibleClassFusions( t, t3 );;
  gap> sfust = Filtered( sfust, map -> ForAny( tfust3,
  >               map2 -> CompositionMaps( map2, map ) in comp ) );
  true
```

Consider the elements of order four in U. There are three such classes inside $U' \cong L_3(4)$, which fuse to one class of $U.3$.

```
────────────────────── Example ──────────────────────
  gap> OrdersClassRepresentatives( s );
  [ 1, 2, 3, 4, 4, 4, 5, 7, 2, 4, 6, 8, 8, 8 ]
  gap> sfuss3;
  [ [ 1, 2, 3, 4, 4, 4, 5, 6, 7, 8, 9, 10, 10, 10 ] ]
```

These classes of U fuse into some of the classes 10 to 12 of G. In $G.3$, these three classes fuse into one class.

```
────────────────────── Example ──────────────────────
  gap> Set( sfust, map -> map{ [ 4 .. 6 ] } );
  [ [ 10, 10, 10 ], [ 10, 10, 11 ], [ 10, 10, 12 ], [ 10, 11, 10 ],
    [ 10, 11, 11 ], [ 10, 11, 12 ], [ 10, 12, 10 ], [ 10, 12, 11 ],
    [ 10, 12, 12 ], [ 11, 10, 10 ], [ 11, 10, 11 ], [ 11, 10, 12 ],
    [ 11, 11, 10 ], [ 11, 11, 11 ], [ 11, 11, 12 ], [ 11, 12, 10 ],
    [ 11, 12, 11 ], [ 11, 12, 12 ], [ 12, 10, 10 ], [ 12, 10, 11 ],
    [ 12, 10, 12 ], [ 12, 11, 10 ], [ 12, 11, 11 ], [ 12, 11, 12 ],
    [ 12, 12, 10 ], [ 12, 12, 11 ], [ 12, 12, 12 ] ]
  gap> Set( tfust3, map -> map{ [ 10 .. 12 ] } );
  [ [ 10, 10, 10 ] ]
```

This means that the automorphism α of G that is induced by the action of $G.3$ permutes the classes 10 to 12 of G transitively. The fact that U extends to $U.3$ in $G.3$ means that U is invariant under α. This implies that U contains either no elements from the classes 10 to 12 or elements from all of these classes. The possible class fusions from U to G satisfying this condition form one orbit under table automprhisms.

```
────────────────────── Example ──────────────────────
  gap> Filtered( sfust, map -> Intersection( map, [ 10 .. 12 ] ) = [] );
  [ ]
  gap> filt:= Filtered( sfust, map -> IsSubset( map, [ 10 .. 12 ] ) );
  [ [ 1, 3, 7, 10, 11, 12, 15, 24, 4, 14, 23, 26, 27, 28 ],
    [ 1, 3, 7, 10, 12, 11, 15, 24, 4, 14, 23, 26, 28, 27 ],
    [ 1, 3, 7, 11, 10, 12, 15, 24, 4, 14, 23, 27, 26, 28 ],
    [ 1, 3, 7, 11, 12, 10, 15, 24, 4, 14, 23, 27, 28, 26 ],
    [ 1, 3, 7, 12, 10, 11, 15, 24, 4, 14, 23, 28, 26, 27 ],
    [ 1, 3, 7, 12, 11, 10, 15, 24, 4, 14, 23, 28, 27, 26 ] ]
  gap> Length( RepresentativesFusions( s, filt, t ) );
  1
```

Finally, we check that a correct map is stored on the library table.

```
 ───────────────────────── Example ─────────────────────────
  gap> GetFusionMap( s, t ) in filt;
  true
```

9.6 Conditions Imposed by Brauer Tables

The examples in this section show that symmetries can be broken as soon as the class fusions between two ordinary tables shall be compatible with the corresponding Brauer character tables. More precisely, we assume that the class fusion from each Brauer table to its ordinary table is already fixed; choosing these fusions consistently can be a nontrivial task, solving so-called "generality problems" may require the construction of certain modules, similar to the arguments used in 9.6.3 below.

9.6.1 $L_2(16).4 \to J_3.2$ (January 2004)

It can happen that Brauer tables decide ambiguities of class fusions between the corresponding ordinary tables. An easy example is the class fusion of $L_2(16).4$ into $J_3.2$. The ordinary tables admit four possible class fusions, of which two are essentially different.

```
 ───────────────────────── Example ─────────────────────────
  gap> s:= CharacterTable( "L2(16).4" );;
  gap> t:= CharacterTable( "J3.2" );;
  gap> fus:= PossibleClassFusions( s, t );
  [ [ 1, 2, 3, 6, 14, 15, 16, 2, 5, 7, 12, 5, 5, 8, 8, 13, 13 ],
    [ 1, 2, 3, 6, 14, 15, 16, 2, 5, 7, 12, 19, 19, 22, 22, 23, 23 ],
    [ 1, 2, 3, 6, 14, 16, 15, 2, 5, 7, 12, 5, 5, 8, 8, 13, 13 ],
    [ 1, 2, 3, 6, 14, 16, 15, 2, 5, 7, 12, 19, 19, 22, 22, 23, 23 ] ]
  gap> RepresentativesFusions( s, fus, t );
  [ [ 1, 2, 3, 6, 14, 15, 16, 2, 5, 7, 12, 5, 5, 8, 8, 13, 13 ],
    [ 1, 2, 3, 6, 14, 15, 16, 2, 5, 7, 12, 19, 19, 22, 22, 23, 23 ] ]
```

Using Brauer tables, we will see that just one fusion is admissible.

We can exclude two possible fusions by the fact that their images all lie inside the normal subgroup J_3, but J_3 does not contain a subgroup of type $L_2(16).4$; so still one orbit of length two remains.

```
 ───────────────────────── Example ─────────────────────────
  gap> j3:= CharacterTable( "J3" );;
  gap> PossibleClassFusions( s, j3 );
  [ ]
  gap> GetFusionMap( j3, t );
  [ 1, 2, 3, 4, 5, 6, 6, 7, 8, 9, 10, 11, 12, 12, 13, 14, 14, 15, 16,
    17, 17 ]
  gap> filt:= Filtered( fus,
  >          x -> not IsSubset( ClassPositionsOfDerivedSubgroup( t ), x ) );
  [ [ 1, 2, 3, 6, 14, 15, 16, 2, 5, 7, 12, 19, 19, 22, 22, 23, 23 ],
    [ 1, 2, 3, 6, 14, 16, 15, 2, 5, 7, 12, 19, 19, 22, 22, 23, 23 ] ]
```

Now the remaining wrong fusion is excluded by the fact that the table automorphism of $J_3.2$ that swaps the two classes of element order 17 –which swaps two of the possible class fusions– does not live in the 2-modular table.

```
——————————————— Example ———————————————
gap> smod2:= s mod 2;;
gap> tmod2:= t mod 2;;
gap> admissible:= [];;  modmap:= "dummy";;
gap> for map in filt do
>       modmap:= CompositionMaps( InverseMap( GetFusionMap( tmod2, t ) ),
>                   CompositionMaps( map, GetFusionMap( smod2, s ) ) );
>     if not fail in Decomposition( Irr( smod2 ),
>           List( Irr( tmod2 ), chi -> chi{ modmap } ), "nonnegative" ) then
>       AddSet( admissible, map );
>     fi;
>   od;
gap> admissible;
[ [ 1, 2, 3, 6, 14, 16, 15, 2, 5, 7, 12, 19, 19, 22, 22, 23, 23 ] ]
```

The test of all available Brauer tables is implemented in the function CTblLib.Test.Decompositions of the GAP Character Table Library ([Bre25]).

```
——————————————— Example ———————————————
gap> CTblLib.Test.Decompositions( s, fus, t ) = admissible;
true
```

We see that p-modular tables alone determine the class fusion uniquely; in fact the primes 2 and 3 suffice for that.

```
——————————————— Example ———————————————
gap> GetFusionMap( s, t ) in admissible;
true
```

Remark:

In May 2015, the 19-modular character table of J_3 has been corrected, by swapping the two classes of element order 17. Since the class fusion of $L_2(16).4$ into $J_3.2$ is uniquely determined by the 2-modular tables of $L_2(16).4$ and $J_3.2$ and since this class fusion has been compatible with the previous version of the 19-modular table of J_3, the correction does not affect the above arguments.

9.6.2 $L_2(17) \rightarrow S_8(2)$ (**July 2004**)

The class fusion of the maximal subgroup $M \cong L_2(17)$ of $G = S_8(2)$ is ambiguous.

```
——————————————— Example ———————————————
gap> m:= CharacterTable( "L2(17)" );;
gap> t:= CharacterTable( "S8(2)" );;
gap> mfust:= PossibleClassFusions( m, t );;
gap> Length( RepresentativesFusions( m, mfust, t ) );
4
```

The Brauer tables for M and G determine the class fusion up to table automorphisms.

```
——————————————— Example ———————————————
gap> filt:= CTblLib.Test.Decompositions( m, mfust, t );;
gap> repr:= RepresentativesFusions( m, filt, t );;
gap> Length( repr );
1
gap> GetFusionMap( m, t ) in repr;
true
```

9.6.3 $L_2(19) \rightarrow J_3$ (**April 2003**)

It can happen that Brauer tables impose conditions such that ambiguities arise which are not visible if one considers only ordinary tables.

The class fusion between the ordinary character tables of $L_2(19)$ and J_3 is unique up to table automorphisms.

```
                          _____ Example _____
  gap> s:= CharacterTable( "L2(19)" );;
  gap> t:= CharacterTable( "J3" );;
  gap> sfust:= PossibleClassFusions( s, t );
  [ [ 1, 2, 4, 6, 7, 10, 11, 12, 13, 14, 20, 21 ],
    [ 1, 2, 4, 6, 7, 10, 11, 12, 13, 14, 21, 20 ],
    [ 1, 2, 4, 6, 7, 11, 12, 10, 13, 14, 20, 21 ],
    [ 1, 2, 4, 6, 7, 11, 12, 10, 13, 14, 21, 20 ],
    [ 1, 2, 4, 6, 7, 12, 10, 11, 13, 14, 20, 21 ],
    [ 1, 2, 4, 6, 7, 12, 10, 11, 13, 14, 21, 20 ],
    [ 1, 2, 4, 7, 6, 10, 11, 12, 14, 13, 20, 21 ],
    [ 1, 2, 4, 7, 6, 10, 11, 12, 14, 13, 21, 20 ],
    [ 1, 2, 4, 7, 6, 11, 12, 10, 14, 13, 20, 21 ],
    [ 1, 2, 4, 7, 6, 11, 12, 10, 14, 13, 21, 20 ],
    [ 1, 2, 4, 7, 6, 12, 10, 11, 14, 13, 20, 21 ],
    [ 1, 2, 4, 7, 6, 12, 10, 11, 14, 13, 21, 20 ] ]
  gap> fusreps:= RepresentativesFusions( s, sfust, t );
  [ [ 1, 2, 4, 6, 7, 10, 11, 12, 13, 14, 20, 21 ] ]
```

The Galois automorphism that permutes the three classes of element order 9 in the tables of ($L_2(19)$ and) J_3 does not live in characteristic 19. For example, the unique irreducible Brauer character of degree 110 in the 19-modular table of J_3 is φ_3, and the value of this character on the class 9A is $-1+2y9+\&4$.

```
                          _____ Example _____
  gap> tmod19:= t mod 19;
  BrauerTable( "J3", 19 )
  gap> deg110:= Filtered( Irr( tmod19 ), phi -> phi[1] = 110 );
  [ Character( BrauerTable( "J3", 19 ),
    [ 110, -2, 5, 2, 2, 0, 0, 1, 0,
        -2*E(9)^2+E(9)^3-E(9)^4-E(9)^5+E(9)^6-2*E(9)^7,
        E(9)^2+E(9)^3-E(9)^4-E(9)^5+E(9)^6+E(9)^7,
        E(9)^2+E(9)^3+2*E(9)^4+2*E(9)^5+E(9)^6+E(9)^7, -2, -2, -1, 0,
        0, E(17)^3+E(17)^5+E(17)^6+E(17)^7+E(17)^10+E(17)^11+E(17)^12
          +E(17)^14,
        E(17)+E(17)^2+E(17)^4+E(17)^8+E(17)^9+E(17)^13+E(17)^15+E(17)^16
      ] ) ]
  gap> 9A:= Position( OrdersClassRepresentatives( tmod19 ), 9 );
  10
  gap> deg110[1][ 9A ];
  -2*E(9)^2+E(9)^3-E(9)^4-E(9)^5+E(9)^6-2*E(9)^7
  gap> AtlasIrrationality( "-1+2y9+&4" ) = deg110[1][ 9A ];
  true
```

It turns out that four of the twelve possible class fusions are not compatible with the 19-modular tables.

```
──────────────────── Example ────────────────────
gap> smod19:= s mod 19;
BrauerTable( "L2(19)", 19 )
gap> compatible:= [];;
gap> for map in sfust do
>       comp:= CompositionMaps( InverseMap( GetFusionMap( tmod19, t ) ),
>         CompositionMaps( map, GetFusionMap( smod19, s ) ) );
>       rest:= List( Irr( tmod19 ), phi -> phi{ comp } );
>       if not fail in Decomposition( Irr( smod19 ), rest, "nonnegative" ) then
>         Add( compatible, map );
>       fi;
>     od;
gap> compatible;
[ [ 1, 2, 4, 6, 7, 11, 12, 10, 13, 14, 20, 21 ],
  [ 1, 2, 4, 6, 7, 11, 12, 10, 13, 14, 21, 20 ],
  [ 1, 2, 4, 6, 7, 12, 10, 11, 13, 14, 20, 21 ],
  [ 1, 2, 4, 6, 7, 12, 10, 11, 13, 14, 21, 20 ],
  [ 1, 2, 4, 7, 6, 11, 12, 10, 14, 13, 20, 21 ],
  [ 1, 2, 4, 7, 6, 11, 12, 10, 14, 13, 21, 20 ],
  [ 1, 2, 4, 7, 6, 12, 10, 11, 14, 13, 20, 21 ],
  [ 1, 2, 4, 7, 6, 12, 10, 11, 14, 13, 21, 20 ] ]
```

Moreover, the subgroups of those table automorphisms of the ordinary tables that leave the set of compatible fusions invariant make two orbits on this set. Indeed, the two orbits belong to essentially different decompositions of the restriction of φ_3.

```
──────────────────── Example ────────────────────
gap> reps:= RepresentativesFusions( s, compatible, t );
[ [ 1, 2, 4, 6, 7, 11, 12, 10, 13, 14, 20, 21 ],
  [ 1, 2, 4, 6, 7, 12, 10, 11, 13, 14, 20, 21 ] ]
gap> compatiblemod19:= List( reps, map -> CompositionMaps(
>         InverseMap( GetFusionMap( tmod19, t ) ),
>         CompositionMaps( map, GetFusionMap( smod19, s ) ) ) );
[ [ 1, 2, 4, 6, 7, 11, 12, 10, 13, 14 ],
  [ 1, 2, 4, 6, 7, 12, 10, 11, 13, 14 ] ]
gap> rest:= List( compatiblemod19, map -> Irr( tmod19 )[3]{ map } );;
gap> dec:= Decomposition( Irr( smod19 ), rest, "nonnegative" );
[ [ 0, 0, 1, 2, 1, 2, 2, 1, 0, 1 ], [ 0, 2, 0, 2, 0, 1, 2, 0, 2, 1 ] ]
gap> List( Irr( smod19 ), phi -> phi[1] );
[ 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 ]
```

In order to decide which class fusion is correct, we take the matrix representation of J_3 that affords φ_3, restrict it to $L_2(19)$, which is the second maximal subgroup of J_3, and compute the composition factors. For that, we use a representation from the Atlas of Group Representations [WWT+], and access it via the GAP package AtlasRep ([WPN+22]).

```
──────────────────── Example ────────────────────
gap> LoadPackage( "atlasrep", false );
true
gap> prog:= AtlasProgram( "J3", "maxes", 2 );
rec( groupname := "J3", identifier := [ "J3", "J3G1-max2W1", 1 ],
  program := <straight line program>, size := 3420,
  standardization := 1, subgroupname := "L2(19)", version := "1" )
```

```
gap> gens:= OneAtlasGeneratingSetInfo( "J3", Characteristic, 19,
>                Dimension, 110 );;
gap> gens:= AtlasGenerators( gens );
rec( charactername := "110a", constituents := [ 3 ],
  contents := "core", dim := 110,
  generators := [ < immutable compressed matrix 110x110 over GF(19) >,
      < immutable compressed matrix 110x110 over GF(19) > ],
  groupname := "J3", id := "",
  identifier := [ "J3", [ "J3G1-f19r110B0.m1", "J3G1-f19r110B0.m2" ],
      1, 19 ], repname := "J3G1-f19r110B0", repnr := 35,
  ring := GF(19), size := 50232960, standardization := 1,
  type := "matff" )
gap> restgens:= ResultOfStraightLineProgram( prog.program, gens.generators );
[ < immutable compressed matrix 110x110 over GF(19) >,
  < immutable compressed matrix 110x110 over GF(19) > ]
gap> module:= GModuleByMats( restgens, GF( 19 ) );;
gap> facts:= SMTX.CollectedFactors( module );;
gap> Length( facts );
7
gap> List( facts, x -> x[1].dimension );
[ 5, 7, 9, 11, 13, 15, 19 ]
gap> List( facts, x -> x[2] );
[ 1, 2, 1, 2, 2, 1, 1 ]
```

This means that there are seven pairwise nonisomorphic composition factors, the smallest one of dimension five. In other words, the first of the two maps is the correct one. Let us check whether this map equals the one that is stored on the library table.

```
─────────────── Example ───────────────
gap> GetFusionMap( s, t ) = reps[1];
true
```

Remark:

In May 2015, the 19-modular character table of J_3 has been corrected, by swapping the two classes of element order 17. This affects the above computations only in one place, where the values of the character deg110 are shown.

9.7 Fusions Determined by Information about the Groups

In the examples in this section, character theoretic arguments do not suffice for determining the class fusions. So we use computations with the groups in question or information about these groups beyond the character table, and perhaps additionally character theoretic arguments.

The group representations are taken from the Atlas of Group Representations [WWT⁺] and are accessed via the GAP package AtlasRep ([WPN⁺22]).

```
─────────────── Example ───────────────
gap> LoadPackage( "atlasrep", false );
true
```

9.7.1 $U_3(3).2 \rightarrow Fi'_{24}$ (November 2002)

The group $G = Fi'_{24}$ contains a maximal subgroup H of type $U_3(3).2$. From the character tables of G and H, one gets a lot of essentially different possibilities (and additionally this takes quite some time). We use the description of H as the normalizer in G of a $U_3(3)$ type subgroup containing elements in the classes 2B, 3D, 3E, 4C, 4C, 6J, 7B, 8C, and 12M (see [BN95]).

```
————————————————————————— Example —————————————————————————
  gap> t:= CharacterTable( "F3+" );
  CharacterTable( "F3+" )
  gap> s:= CharacterTable( "U3(3).2" );
  CharacterTable( "U3(3).2" )
  gap> tnames:= ClassNames( t, "ATLAS" );
  [ "1A", "2A", "2B", "3A", "3B", "3C", "3D", "3E", "4A", "4B", "4C",
    "5A", "6A", "6B", "6C", "6D", "6E", "6F", "6G", "6H", "6I", "6J",
    "6K", "7A", "7B", "8A", "8B", "8C", "9A", "9B", "9C", "9D", "9E",
    "9F", "10A", "10B", "11A", "12A", "12B", "12C", "12D", "12E",
    "12F", "12G", "12H", "12I", "12J", "12K", "12L", "12M", "13A",
    "14A", "14B", "15A", "15B", "15C", "16A", "17A", "18A", "18B",
    "18C", "18D", "18E", "18F", "18G", "18H", "20A", "20B", "21A",
    "21B", "21C", "21D", "22A", "23A", "23B", "24A", "24B", "24C",
    "24D", "24E", "24F", "24G", "26A", "27A", "27B", "27C", "28A",
    "29A", "29B", "30A", "30B", "33A", "33B", "35A", "36A", "36B",
    "36C", "36D", "39A", "39B", "39C", "39D", "42A", "42B", "42C",
    "45A", "45B", "60A" ]
  gap> OrdersClassRepresentatives( s );
  [ 1, 2, 3, 3, 4, 4, 6, 7, 8, 12, 2, 4, 6, 8, 12, 12 ]
  gap> sfust:= List( [ "1A", "2B", "3D", "3E", "4C", "4C", "6J", "7B", "8C",
  >                    "12M" ], x -> Position( tnames, x ) );
  [ 1, 3, 7, 8, 11, 11, 22, 25, 28, 50 ]
  gap> sfust:= PossibleClassFusions( s, t, rec( fusionmap:= sfust ) );
  [ [ 1, 3, 7, 8, 11, 11, 22, 25, 28, 50, 3, 9, 23, 28, 43, 43 ],
    [ 1, 3, 7, 8, 11, 11, 22, 25, 28, 50, 3, 11, 23, 28, 50, 50 ] ]
  gap> OrdersClassRepresentatives( s );
  [ 1, 2, 3, 3, 4, 4, 6, 7, 8, 12, 2, 4, 6, 8, 12, 12 ]
```

So we still have two possibilities, which differ on the outer classes of element order 4 and 12.

Our idea is to take a subgroup U of H that contains such elements, and to compute the possible class fusions of U into G, via the factorization through a suitable maximal subgroup M of G.

We take $U = N_H(\langle g \rangle)$ where g is an element in the first class of order three elements of H; this is a maximal subgroup of H, of order 216.

```
————————————————————————— Example —————————————————————————
  gap> Maxes( s );
  [ "U3(3)", "3^(1+2):SD16", "L3(2).2", "2^(1+4).S3", "4^2:D12" ]
  gap> SizesCentralizers( s );
  [ 12096, 192, 216, 18, 96, 32, 24, 7, 8, 12, 48, 48, 6, 8, 12, 12 ]
  gap> u:= CharacterTable( Maxes( s )[2] );;
  gap> ufuss:= GetFusionMap( u, s );
  [ 1, 2, 11, 3, 4, 5, 12, 7, 13, 9, 9, 15, 16, 10 ]
```

Candidates for M are those subgroups of G that contain elements in the class 3D of G whose centralizer is the full 3D centralizer in G.

```
─────────────────────────────── Example ───────────────────────────────
gap> 3Dcentralizer:= SizesCentralizers( t )[7];
153055008
gap> cand:= [];;
gap> for name in Maxes( t ) do
>      m:= CharacterTable( name );
>      mfust:= GetFusionMap( m, t );
>      if ForAny( [ 1 .. Length( mfust ) ],
>          i -> mfust[i] = 7 and SizesCentralizers( m )[i] = 3Dcentralizer )
>      then
>        Add( cand, m );
>      fi;
>    od;
gap> cand;
[ CharacterTable( "3^7.07(3)" ),
  CharacterTable( "3^2.3^4.3^8.(A5x2A4).2" ) ]
```

For these two groups M, we show that the possible class fusions from U to G via M factorize through H only if the second possible class fusion from H to G is chosen.

```
─────────────────────────────── Example ───────────────────────────────
gap> possufust:= List( sfust, x -> CompositionMaps( x, ufuss ) );
[ [ 1, 3, 3, 7, 8, 11, 9, 22, 23, 28, 28, 43, 43, 50 ],
  [ 1, 3, 3, 7, 8, 11, 11, 22, 23, 28, 28, 50, 50, 50 ] ]
gap> m:= cand[1];;
gap> ufusm:= PossibleClassFusions( u, m );;
gap> Length( ufusm );
242
gap> comp:= List( ufusm, x -> CompositionMaps( GetFusionMap( m, t ), x ) );;
gap> Intersection( possufust, comp );
[ [ 1, 3, 3, 7, 8, 11, 11, 22, 23, 28, 28, 50, 50, 50 ] ]
gap> m:= cand[2];;
gap> ufusm:= PossibleClassFusions( u, m );;
gap> Length( ufusm );
256
gap> comp:= List( ufusm, x -> CompositionMaps( GetFusionMap( m, t ), x ) );;
gap> Intersection( possufust, comp );
[ [ 1, 3, 3, 7, 8, 11, 11, 22, 23, 28, 28, 50, 50, 50 ] ]
```

Finally, we check that the correct fusion is stored in the GAP Character Table Library.

```
─────────────────────────────── Example ───────────────────────────────
gap> GetFusionMap( s, t ) = sfust[2];
true
```

9.7.2 $L_2(13).2 \rightarrow Fi'_{24}$ (September 2002)

The class fusion of maximal subgroups U of type $L_2(13).2$ in $G = Fi'_{24}$ is ambiguous.

```
─────────────────────────────── Example ───────────────────────────────
gap> t:= CharacterTable( "F3+" );;
gap> u:= CharacterTable( "L2(13).2" );;
gap> fus:= PossibleClassFusions( u, t );;
```

```
gap> repr:= RepresentativesFusions( u, fus, t );;
gap> Length( repr );
3
```

In [LW91, p. 155], it is stated that U' contains elements in the classes 2B, 3D, and 7B of G. (Note that the two conjugacy classes of groups isomorphic to U have the same class fusion because the outer automorphism of G fixes the relevant classes.)

```
                          _____ Example _____
gap> filt:= Filtered( repr, x -> t.2b in x and t.3d in x and t.7b in x );
[ [ 1, 3, 7, 22, 25, 25, 25, 51, 3, 9, 43, 43, 53, 53, 53 ],
  [ 1, 3, 7, 22, 25, 25, 25, 51, 3, 11, 50, 50, 53, 53, 53 ] ]
gap> ClassNames( t ){ [ 43, 50 ] };
[ "12f", "12m" ]
```

So we have to decide whether U contains elements in the class 12F or in 12M of G.

The order 12 elements in question lie inside subgroups of type 13 : 12 in U. These subgroups are clearly contained in the Sylow 13 normalizers of G, which are contained in maximal subgroups of type $(3^2 : 2 \times G_2(3)).2$ in G; the class fusion of the latter groups is unique up to table automorphisms.

```
                          _____ Example _____
gap> pos:= Position( OrdersClassRepresentatives( t ), 13 );
51
gap> SizesCentralizers( t )[ pos ];
234
gap> ClassOrbit( t, pos );
[ 51 ]
gap> cand:= [];;
gap> for name in Maxes( t ) do
>      m:= CharacterTable( name );
>      pos:= Position( OrdersClassRepresentatives( m ), 13 );
>      if pos <> fail and
>         SizesCentralizers( m )[ pos ] = 234
>         and ClassOrbit( m, pos ) = [ pos ] then
>        Add( cand, m );
>      fi;
>    od;
gap> cand;
[ CharacterTable( "(3^2:2xG2(3)).2" ) ]
gap> s:= cand[1];;
gap> sfust:= PossibleClassFusions( s, t );;
```

As no 13 : 12 type subgroup is contained in the derived subgroup of $(3^2 : 2 \times G_2(3)).2$, we look at the elements of order 12 in the outer half.

```
                          _____ Example _____
gap> der:= ClassPositionsOfDerivedSubgroup( s );;
gap> outer:= Difference( [ 1 .. NrConjugacyClasses( s ) ], der );;
gap> sfust:= PossibleClassFusions( s, t );;
gap> imgs:= Set( Flat( List( sfust, x -> x{ outer } ) ) );
[ 2, 3, 10, 11, 15, 17, 18, 19, 21, 22, 26, 44, 45, 49, 50, 52, 62,
  83, 87, 98 ]
gap> t.12f in imgs;
```

```
false
gap> t.12m in imgs;
true
```

So $L_2(13).2 \setminus L_2(13)$ does not contain 12F elements of G, i. e., we have determined the class fusion of U in G.

Finally, we check whether the correct fusion is stored in the GAP Character Table Library.

```
                              ── Example ──
gap> GetFusionMap( u, t ) = filt[2];
true
```

9.7.3 $M_{11} \rightarrow B$ (April 2009)

The sporadic simple group B contains a maximal subgroup M of the type M_{11} whose class fusion is ambiguous.

```
                              ── Example ──
gap> b:= CharacterTable( "B" );;
gap> m11:= CharacterTable( "M11" );;
gap> m11fusb:= PossibleClassFusions( m11, b );;
gap> Length( m11fusb );
31
gap> CompositionMaps( ClassNames( b, "ATLAS" ), Parametrized( m11fusb ) );
[ "1A", [ "2B", "2D" ], [ "3A", "3B" ],
  [ "4B", "4E", "4G", "4H", "4J" ], [ "5A", "5B" ],
  [ "6C", "6E", "6H", "6I", "6J" ],
  [ "8B", "8E", "8G", "8J", "8K", "8L", "8M", "8N" ],
  [ "8B", "8E", "8G", "8J", "8K", "8L", "8M", "8N" ], "11A", "11A" ]
```

According to [Wil93a, Thm. 12.1], M contains no 5A elements of B. By the proof of [Wil99, Prop. 4.1], the involutions in any S_5 type subgroup U of M lie in the class 2C or 2D of B, and since the possible class fusions of M computed above admit only involutions in the class 2B or 2D, all involutions of U lie in the class 2D. Again by the proof of [Wil99, Prop. 4.1], U is contained in a maximal subgroup of type Th in B.

Now we use the embedding of U into B via M and Th for determining the class fusion of M into B. The class fusion of the embedding of U via Th is uniquely determined.

```
                              ── Example ──
gap> th:= CharacterTable( "Th" );;
gap> s5:= CharacterTable( "S5" );;
gap> s5fusth:= PossibleClassFusions( s5, th );
[ [ 1, 2, 4, 8, 2, 7, 11 ] ]
gap> thfusb:= PossibleClassFusions( th, b );;
gap> s5fusb:= Set( thfusb, x -> CompositionMaps( x, s5fusth[1] ) );
[ [ 1, 5, 7, 19, 5, 17, 29 ] ]
```

Also the class fusion of U into M is unique, and this determines the class fusion of M into B.

```
                              ── Example ──
gap> s5fusm11:= PossibleClassFusions( s5, m11 );
[ [ 1, 2, 3, 5, 2, 4, 6 ] ]
```

```
gap> m11fusb:= Filtered( m11fusb,
>                    map -> CompositionMaps( map, s5fusm11[1] ) = s5fusb[1] );
[ [ 1, 5, 7, 17, 19, 29, 45, 45, 54, 54 ] ]
gap> CompositionMaps( ClassNames( b, "ATLAS" ), m11fusb[1] );
[ "1A", "2D", "3B", "4J", "5B", "6J", "8N", "8N", "11A", "11A" ]
```

(Using the information that the M_{10} type subgroups of M are also contained in Th type subgroups would not have helped us, since these subgroups do not contain elements of order 6, and two possibilities would have remained.)

9.7.4 $L_2(11) : 2 \to B$ (April 2009)

The sporadic simple group B contains a maximal subgroup L of the type $L_2(11) : 2$ whose class fusion is ambiguous.

```
————————————————————— Example —————————————————————
gap> b:= CharacterTable( "B" );;
gap> l:= CharacterTable( "L2(11).2" );;
gap> lfusb:= PossibleClassFusions( l, b );;
gap> Length( lfusb );
16
gap> CompositionMaps( ClassNames( b, "ATLAS" ), Parametrized( lfusb ) );
[ "1A", [ "2B", "2D" ], [ "3A", "3B" ], [ "5A", "5B" ],
  [ "5A", "5B" ], [ "6C", "6H", "6I", "6J" ], "11A", [ "2C", "2D" ],
  [ "4D", "4E", "4F", "4G", "4H", "4J" ], [ "10C", "10E", "10F" ],
  [ "10C", "10E", "10F" ],
  [ "12E", "12F", "12H", "12I", "12J", "12L", "12N", "12P", "12Q",
      "12R", "12S" ],
  [ "12E", "12F", "12H", "12I", "12J", "12L", "12N", "12P", "12Q",
      "12R", "12S" ] ]
```

According to [Wil93a, Thm. 12.1], L contains no 5A elements of B. By the proof of [Wil99, Prop. 4.1], B contains exactly one class of $L_2(11)$ type subgroups with this property. Hence the subgroup U of index two in L is contained in a maximal subgroup M of type M_{11} in B, whose class fusion was determined in Section 9.7.3.

In the same way as we proceeded in Section 9.7.3, we use the embedding of U into B via L and M for determining the class fusion of L into B.

```
————————————————————— Example —————————————————————
gap> m:= CharacterTable( "M11" );;
gap> u:= CharacterTable( "L2(11)" );;
gap> ufusm:= PossibleClassFusions( u, m );;
gap> mfusb:= GetFusionMap( m, b );;
gap> ufusb:= Set( ufusm, x -> CompositionMaps( mfusb, x ) );
[ [ 1, 5, 7, 19, 19, 29, 54, 54 ] ]
gap> ufusl:= PossibleClassFusions( u, l );
[ [ 1, 2, 3, 4, 5, 6, 7, 7 ], [ 1, 2, 3, 5, 4, 6, 7, 7 ] ]
gap> lfusb:= Filtered( lfusb,
>               map2 -> ForAny( ufusl,
>                       map1 -> CompositionMaps( map2, map1 ) in ufusb ) );
[ [ 1, 5, 7, 19, 19, 29, 54, 5, 15, 53, 53, 73, 73 ] ]
```

9.7.5 $L_3(3) \to B$ (April 2009)

The sporadic simple group B contains a maximal subgroup T of the type $L_3(3)$ whose class fusion is ambiguous.

```
 ─────────────────────────── Example ───────────────────────────
  gap> b:= CharacterTable( "B" );;
  gap> t:= CharacterTable( "L3(3)" );;
  gap> tfusb:= PossibleClassFusions( t, b );;
  gap> Length( tfusb );
  36
```

According to [Wil99, Section 9], T contains a subgroup U of the type $3^2 : 2S_4$ that is contained also in a maximal subgroup M of the type $3^2.3^3.3^6.(S_4 \times 2S_4)$. So we throw away the possible fusions from T to B that are not compatible with the compositions of the embeddings of U into B via T and M.

```
 ─────────────────────────── Example ───────────────────────────
  gap> m:= CharacterTable( "3^2.3^3.3.3^6.(S4x2S4)" );;
  gap> g:= PSL(3,3);;
  gap> mx:= MaximalSubgroupClassReps( g );;
  gap> u:= First( mx, x -> Size( x ) = 432 );;
  gap> u:= CharacterTable( u );;
  gap> ufusm:= PossibleClassFusions( u, m );;
  gap> ufust:= PossibleClassFusions( u, t );;
  gap> mfusb:= GetFusionMap( m, b );;
  gap> ufusb:= Set( ufusm, map -> CompositionMaps( mfusb, map ) );;
  gap> tfusb:= Filtered( tfusb, map -> ForAny( ufust,
  >        map2 -> CompositionMaps( map, map2 ) in ufusb ) );;
  gap> tfusb;
  [ [ 1, 5, 6, 7, 12, 27, 41, 41, 75, 75, 75, 75 ],
    [ 1, 5, 7, 6, 12, 28, 41, 41, 75, 75, 75, 75 ],
    [ 1, 5, 7, 7, 12, 28, 41, 41, 75, 75, 75, 75 ],
    [ 1, 5, 7, 7, 12, 29, 41, 41, 75, 75, 75, 75 ],
    [ 1, 5, 7, 7, 17, 29, 45, 45, 75, 75, 75, 75 ] ]
```

Now we use that T does not contain 4E elements of B (again see [Wil99, Section 9]). Thus the last of the five candidates is the correct class fusion.

```
 ─────────────────────────── Example ───────────────────────────
  gap> ClassNames( b, "ATLAS" ){ [ 12, 17 ] };
  [ "4E", "4J" ]
```

We check that this map is stored on the library table.

```
 ─────────────────────────── Example ───────────────────────────
  gap> GetFusionMap( t, b ) = tfusb[5];
  true
```

9.7.6 $L_2(17).2 \to B$ (March 2004)

The sporadic simple group B contains a maximal subgroup U of the type $L_2(17).2$ whose class fusion is ambiguous.

```
                            ──────── Example ────────
  gap> b:= CharacterTable( "B" );;
  gap> u:= CharacterTable( "L2(17).2" );;
  gap> ufusb:= PossibleClassFusions( u, b );
  [ [ 1, 5, 7, 15, 42, 42, 47, 47, 47, 91, 4, 30, 89, 89, 89, 89, 97,
        97, 97 ],
    [ 1, 5, 7, 15, 44, 44, 46, 46, 46, 91, 5, 29, 90, 90, 90, 90, 96,
        96, 96 ],
    [ 1, 5, 7, 15, 44, 44, 47, 47, 47, 91, 5, 29, 90, 90, 90, 90, 95,
        95, 95 ] ]
```

According to [Wil99, Prop. 11.1], *U* contains elements in the classes 8M and 9A of *B*. This determines the fusion map.

```
                            ──────── Example ────────
  gap> names:= ClassNames( b, "ATLAS" );;
  gap> pos:= List( [ "8M", "9A" ], x -> Position( names, x ) );
  [ 44, 46 ]
  gap> ufusb:= Filtered( ufusb, map -> IsSubset( map, pos ) );
  [ [ 1, 5, 7, 15, 44, 44, 46, 46, 46, 91, 5, 29, 90, 90, 90, 90, 96,
        96, 96 ] ]
```

We check that this map is stored on the library table.

```
                            ──────── Example ────────
  gap> GetFusionMap( u, b ) = ufusb[1];
  true
```

9.7.7 $L_2(49).2_3 \rightarrow B$ (**June 2006**)

The sporadic simple group *B* contains a class of maximal subgroups of the type $L_2(49).2_3$ (a non-split extension of $L_2(49)$, see [Wil93b, Theorem 2]). Let *U* be such a subgroup. The class fusion of *U* in *B* is not determined by the character tables of *U* and *B*.

```
                            ──────── Example ────────
  gap> u:= CharacterTable( "L2(49).2_3" );;
  gap> b:= CharacterTable( "B" );;
  gap> ufusb:= PossibleClassFusions( u, b );;
  gap> Length( RepresentativesFusions( u, ufusb, b ) );
  2
  gap> ufusb;
  [ [ 1, 5, 7, 15, 19, 28, 31, 42, 42, 71, 125, 125, 128, 128, 128,
        128, 128, 15, 71, 71, 89, 89, 89, 89 ],
    [ 1, 5, 7, 15, 19, 28, 31, 42, 42, 71, 125, 125, 128, 128, 128,
        128, 128, 17, 72, 72, 89, 89, 89, 89 ] ]
```

We show that the fusion is determined by the embeddings of the Sylow 7 normalizer *N*, say, of *U* into *U* and into the Sylow 7 normalizer of *B*. (Note that the fusion of the latter group into *B* has been determined in Section 9.3.1.)

For that, we compute the character table of *N* from a representation of *U*. Note that *U* is a non-split extension of the simple group $L_2(49)$ by the product of a diagonal automorphism and a field automorphism. In [Wil93b], the structure of *N* is described as $7^2 : (3 \times Q_{16})$.

```
──────────────── Example ────────────────
gap> g:= SL( 2, 49 );;
gap> gens:= GeneratorsOfGroup( g );;
gap> f:= GF(49);;
gap> mats:= List( gens, x -> IdentityMat( 4, f ) );;
gap> for i in [ 1 .. Length( gens ) ] do
>       mats[i]{ [ 1, 2 ] }{ [ 1, 2 ] }:= gens[i];
>       mats[i]{ [ 3, 4 ] }{ [ 3, 4 ] }:= List( gens[i],
>                                          x -> List( x, y -> y^7 ) );
>    od;
gap> fieldaut:= PermutationMat( (1,3)(2,4), 4, f );;
gap> diagaut:= IdentityMat( 4, f );;
gap> diagaut[1][1]:= Z(49);;
gap> diagaut[3][3]:= Z(49)^7;;
gap> g:= Group( Concatenation( mats, [ fieldaut * diagaut ] ) );;
gap> v:= [ 1, 0, 0, 0 ] * Z(7)^0;;
gap> orb:= Orbit( g, v, OnLines );;
gap> act:= Action( g, orb, OnLines );;
gap> n:= Normalizer( act, SylowSubgroup( act, 7 ) );;
gap> ntbl:= CharacterTable( n );;
```

Now we compute the possible class fusions of *N* into *B*, via the Sylow 7 normalizer in *B*.

```
──────────────── Example ────────────────
gap> bn7:= CharacterTable( "BN7" );;
gap> nfusbn7:= PossibleClassFusions( ntbl, bn7 );;
gap> Length( RepresentativesFusions( ntbl, nfusbn7, bn7 ) );
3
gap> nfusb:= SetOfComposedClassFusions( PossibleClassFusions( bn7, b ),
>                                       nfusbn7 );;
gap> Length( RepresentativesFusions( ntbl, nfusb, b ) );
5
```

Although there are several possibilities, this information is enough to exclude one of the possible fusions of *U* into *B*.

```
──────────────── Example ────────────────
gap> nfusu:= PossibleClassFusions( ntbl, u );;
gap> Length( nfusu );
4
gap> filt:= Filtered( ufusb,
>             x -> ForAny( nfusu, y -> CompositionMaps( x, y ) in nfusb ) );
[ [ 1, 5, 7, 15, 19, 28, 31, 42, 42, 71, 125, 125, 128, 128, 128,
      128, 128, 17, 72, 72, 89, 89, 89, 89 ] ]
gap> ClassNames( b, "ATLAS" ){ filt[1] };
[ "1A", "2D", "3B", "4H", "5B", "6I", "7A", "8K", "8K", "12Q", "24L",
  "24L", "25A", "25A", "25A", "25A", "25A", "4J", "12R", "12R",
  "16G", "16G", "16G", "16G" ]
```

So the class fusion of *U* into *B* can be described by the property that the elements of order four inside and outside the simple subgroup $L_2(49)$ are not conjugate in *B*.

We check that the correct map is stored on the library table.

```
—————————— Example ——————————
gap> GetFusionMap( u, b ) in filt;
true
```

Let us confirm that the two groups of the types $L_2(49).2_1$ and $L_2(49).2_2$ cannot occur as subgroups of B. First we show that $L_2(49).2_1$ is isomorphic with PGL$(2, 49)$, an extension of $L_2(49)$ by a diagonal automorphism, and $L_2(49).2_2$ is an extension by a field automorphism.

```
—————————— Example ——————————
gap> NrConjugacyClasses( u );  NrConjugacyClasses( act );
24
24
gap> u:= CharacterTable( "L2(49).2_1" );;
gap> g:= Group( Concatenation( mats, [ diagaut ] ) );;
gap> orb:= Orbit( g, v, OnLines );;
gap> act:= Action( g, orb, OnLines );;
gap> Size(act );
117600
gap> NrConjugacyClasses( u );  NrConjugacyClasses( act );
51
51
gap> u:= CharacterTable( "L2(49).2_2" );;
gap> g:= Group( Concatenation( mats, [ fieldaut ] ) );;
gap> orb:= Orbit( g, v, OnLines );;
gap> act:= Action( g, orb, OnLines );;
gap> NrConjugacyClasses( u );  NrConjugacyClasses( act );
27
27
```

The group $L_2(49).2_1$ can be excluded because no class fusion into B is possible.

```
—————————— Example ——————————
gap> PossibleClassFusions( CharacterTable( "L2(49).2_1" ), b );
[ ]
```

For $L_2(49).2_2$, it is not that easy. We would get several possible class fusions into B. However, the Sylow 7 normalizer of $L_2(49).2_2$ does not admit a class fusion into the Sylow 7 normalizer of B.

```
—————————— Example ——————————
gap> n:= Normalizer( act, SylowSubgroup( act, 7 ) );;
gap> Length( PossibleClassFusions( CharacterTable( n ), bn7 ) );
0
```

9.7.8 $2^3.L_3(2) \to G_2(5)$ **(January 2004)**

The Chevalley group $G = G_2(5)$ contains a maximal subgroup U of the type $2^3.L_3(2)$ whose class fusion is ambiguous.

```
—————————— Example ——————————
gap> t:= CharacterTable( "G2(5)" );;
gap> s:= CharacterTable( "2^3.L3(2)" );;
gap> sfust:= PossibleClassFusions( s, t );;
gap> RepresentativesFusions( s, sfust, t );
```

```
  [ [ 1, 2, 2, 5, 6, 4, 13, 16, 17, 15, 15 ],
    [ 1, 2, 2, 5, 6, 4, 14, 16, 17, 15, 15 ] ]
gap> OrdersClassRepresentatives( s );
[ 1, 2, 2, 4, 4, 3, 6, 8, 8, 7, 7 ]
```

So the question is whether U contains elements in the class 6B or 6C of G (position 13 or 14 in the Atlas table). We use a permutation representation of G, restrict it to U, and compute the centralizer in G of a suitable element of order 6 in U.

```
                          ____ Example ____
gap> g:= AtlasGroup( "G2(5)" );;
gap> u:= AtlasSubgroup( "G2(5)", 7 );;
gap> Size( u );
1344
gap> repeat
>       x:= Random( u );
>    until Order( x ) = 6;
gap> siz:= Size( Centralizer( g, x ) );
36
gap> Filtered( [ 1 .. NrConjugacyClasses( t ) ],
>            i -> SizesCentralizers( t )[i] = siz );
[ 14 ]
```

So U contains 6C elements in $G_2(5)$.

```
                          ____ Example ____
gap> GetFusionMap( s, t ) in Filtered( sfust, map -> 14 in map );
true
```

9.7.9 $5^{1+4}.2^{1+4}.A_5.4 \to B$ (April 2009)

The sporadic simple group B contains a maximal subgroup M of the type $5^{1+4}.2^{1+4}.A_5.4$ whose class fusion is ambiguous.

```
                          ____ Example ____
gap> b:= CharacterTable( "B" );;
gap> m:= CharacterTable( "5^(1+4).2^(1+4).A5.4" );;
gap> mfusb:= PossibleClassFusions( m, b );;
gap> Length( mfusb );
4
gap> repres:= RepresentativesFusions( m, mfusb, b );;
gap> Length( repres );
2
```

The restriction of the unique irreducible character of degree 4371 distinguishes the two possibilities,

```
                          ____ Example ____
gap> char:= Filtered( Irr( b ), x -> x[1] = 4371 );;
gap> Length( char );
1
gap> rest:= List( repres, map -> char[1]{ map } );;
gap> scprs:= MatScalarProducts( m, Irr( m ), rest );;
```

```
gap> constit:= List( scprs,
>                      x -> Filtered( [1 .. Length(x) ], i -> x[i] <> 0 ) );
[ [ 2, 27, 60, 63, 73, 74, 75, 79, 82 ],
  [ 2, 27, 60, 63, 70, 72, 75, 79, 84 ] ]
gap> List( constit, x -> List( Irr( m ){ x }, Degree ) );
[ [ 1, 6, 384, 480, 400, 400, 500, 1000, 1200 ],
  [ 1, 6, 384, 480, 100, 300, 500, 1000, 1600 ] ]
```

The database [WWT$^+$] contains the 3-modular reduction of the irreducible representation of degree 4371 and also a straight line program for restricting this representation to *M*. We access these data via the GAP package AtlasRep (see [WPN$^+$22]), and compute the composition factors of the natural module of this restriction.

```
─────────── Example ───────────
gap> g:= AtlasSubgroup( "B", Dimension, 4371, Ring, GF(3), 21 );;
gap> module:= GModuleByMats( GeneratorsOfGroup( g ), GF(3) );;
gap> dec:= MTX.CompositionFactors( module );;
gap> SortedList( List( dec, x -> x.dimension ) );
[ 1, 6, 100, 384, 400, 400, 400, 480, 1000, 1200 ]
```

We see that exactly one ordinary constituent does not stay irreducible upon restriction to characteristic 3. Thus the first of the two possible class fusions is the correct one.

9.7.10 The fusion from the character table of $7^2 : 2L_2(7).2$ into the table of marks (January 2004)

It can happen that the class fusion from the ordinary character table of a group *G* into the table of marks of *G* is not unique up to table automorphisms of the character table of *G*.

As an example, consider $G = 7^2 : 2L_2(7).2$, a maximal subgroup in the sporadic simple group *He*.

G contains four classes of cyclic subgroups of order 7. One contains the elements in the normal subgroup of type 7^2, and the other three are preimages of the order 7 elements in the factor group $L_2(7)$. The conjugacy classes of nonidentity elements in the latter three classes split into two Galois conjugates each, which are permuted cyclicly by the table automorphisms of the character table of *G*, but on which the stabilizer of one class acts trivially. This means that determining one of the three classes determines also the other two.

```
─────────── Example ───────────
gap> tbl:= CharacterTable( "7^2:2psl(2,7)" );
CharacterTable( "7^2:2psl(2,7)" )
gap> tom:= TableOfMarks( tbl );
TableOfMarks( "7^2:2L2(7)" )
gap> fus:= PossibleFusionsCharTableTom( tbl, tom );
[ [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 8, 10, 9, 16, 7, 10, 9, 8, 16 ],
  [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 9, 8, 10, 16, 7, 8, 10, 9, 16 ],
  [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 10, 9, 8, 16, 7, 9, 8, 10, 16 ],
  [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 8, 9, 10, 16, 7, 9, 10, 8, 16 ],
  [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 10, 8, 9, 16, 7, 8, 9, 10, 16 ],
  [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 9, 10, 8, 16, 7, 10, 8, 9, 16 ] ]
gap> reps:= RepresentativesFusions( tbl, fus, Group(()) );
[ [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 8, 9, 10, 16, 7, 9, 10, 8, 16 ],
  [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 8, 10, 9, 16, 7, 10, 9, 8, 16 ] ]
gap> AutomorphismsOfTable( tbl );
```

```
Group([ (9,14)(10,17)(11,15)(12,16)(13,18), (7,8), (10,11,12)
  (15,16,17) ])
gap> OrdersClassRepresentatives( tbl );
[ 1, 7, 2, 4, 3, 6, 8, 8, 7, 7, 7, 7, 14, 7, 7, 7, 7, 14 ]
gap> perms1:= PermCharsTom( reps[1], tom );;
gap> perms2:= PermCharsTom( reps[2], tom );;
gap> perms1 = perms2;
false
gap> Set( perms1 ) = Set( perms2 );
true
```

The table of marks of *G* does not distinguish the three classes of cyclic subgroups, there are permutations of rows and columns that act as an S_3 on them.

Note that an S_3 acts on the classes in question in the *rational* character table. So it is due to the irrationalities in the character table that it contains more information.

```
—————————————————————— Example ——————————————————————
gap> Display( tbl );
7^2:2psl(2,7)

      2  4  .  4  3  1  1  3  3  1  .  .  .  1  1  .  .  .
      3  1  .  1  .  1  1  .  .  .  .  .  .  .  .  .  .  .
      7  3  3  1  .  .  .  .  .  2  2  2  2  1  2  2  2  2

         1a 7a 2a 4a 3a 6a 8a 8b 7b 7c 7d 7e 14a 7f 7g 7h 7i
     2P  1a 7a 1a 2a 3a 3a 4a 4a 7b 7c 7d 7e  7b 7f 7g 7h 7i
     3P  1a 7a 2a 4a 1a 2a 8b 8a 7f 7i 7g 7h 14b 7b 7d 7e 7c
     5P  1a 7a 2a 4a 3a 6a 8b 8a 7f 7i 7g 7h 14b 7b 7d 7e 7c
     7P  1a 1a 2a 4a 3a 6a 8a 8b 1a 1a 1a 1a  2a 1a 1a 1a 1a
    11P  1a 7a 2a 4a 3a 6a 8b 8a 7b 7c 7d 7e 14a 7f 7g 7h 7i
    13P  1a 7a 2a 4a 3a 6a 8b 8a 7f 7i 7g 7h 14b 7b 7d 7e 7c

X.1      1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
X.2      3  3  3 -1  .  .  1  1  B  B  B  B  B /B /B /B /B
X.3      3  3  3 -1  .  .  1  1 /B /B /B /B /B  B  B  B  B
X.4      6  6  6  2  .  .  .  . -1 -1 -1 -1 -1 -1 -1 -1 -1
X.5      7  7  7 -1  1  1 -1 -1  .  .  .  .  .  .  .  .  .
X.6      8  8  8  . -1 -1  .  .  1  1  1  1  1  1  1  1  1
X.7      4  4 -4  .  1 -1  .  . -B -B -B -B  B -/B -/B -/B -/B
X.8      4  4 -4  .  1 -1  .  . -/B -/B -/B -/B /B -B -B -B -B
X.9      6  6 -6  .  .  .  A -A -1 -1 -1 -1  1 -1 -1 -1 -1
X.10     6  6 -6  .  .  . -A  A -1 -1 -1 -1  1 -1 -1 -1 -1
X.11     8  8 -8  . -1  1  .  .  1  1  1  1 -1  1  1  1  1
X.12    48 -1  .  .  .  .  .  .  6 -1 -1 -1  .  6 -1 -1 -1
X.13    48 -1  .  .  .  .  .  .  C -1 /C /D  . /C  C  D -1
X.14    48 -1  .  .  .  .  .  .  C /C /D -1  . /C  D -1  C
X.15    48 -1  .  .  .  .  .  . /C  D -1  C  .  C -1 /C /D
X.16    48 -1  .  .  .  .  .  .  C /D -1 /C  . /C -1  C  D
X.17    48 -1  .  .  .  .  .  . /C  C  D -1  .  C /D -1 /C
X.18    48 -1  .  .  .  .  .  . /C -1  C  D  .  C /C /D -1

      2  1
      3  .
```

```
          7    1

               14b
          2P  7f
          3P 14a
          5P 14a
          7P  2a
         11P 14b
         13P 14a

   X.1         1
   X.2        /B
   X.3         B
   X.4        -1
   X.5         .
   X.6         1
   X.7        /B
   X.8         B
   X.9         1
   X.10        1
   X.11       -1
   X.12        .
   X.13        .
   X.14        .
   X.15        .
   X.16        .
   X.17        .
   X.18        .

   A = E(8)-E(8)^3
     = Sqrt(2) = r2
   B = E(7)+E(7)^2+E(7)^4
     = (-1+Sqrt(-7))/2 = b7
   C = 2*E(7)+2*E(7)^2+2*E(7)^4
     = -1+Sqrt(-7) = 2b7
   D = -3*E(7)-3*E(7)^2-2*E(7)^3-3*E(7)^4-2*E(7)^5-2*E(7)^6
     = (5-Sqrt(-7))/2 = 2-b7
   gap> mat:= MatTom( tom );;
   gap> mataut:= MatrixAutomorphisms( mat );;
   gap> Print( mataut, "\n" );
   Group( [ (11,12)(23,24)(27,28)(46,47)(53,54)(56,57),
     ( 9,10)(20,21)(31,32)(38,39), ( 8, 9)(20,22)(31,33)(38,40) ] )
   gap> RepresentativesFusions( Group( () ), reps, mataut );
   [ [ 1, 6, 2, 4, 3, 5, 13, 13, 7, 8, 9, 10, 16, 7, 9, 10, 8, 16 ] ]
```

We could say that thus the fusion is unique up to table automorphisms and automorphisms of the table of marks. But since a group is associated with the table of marks, we compute the character table from the group, and decide which class fusion is correct.

```
───── Example ─────
   gap> g:= UnderlyingGroup( tom );;
   gap> tg:= CharacterTable( g );;
   gap> tgfustom:= FusionCharTableTom( tg, tom );;
```

```
gap> trans:= TransformingPermutationsCharacterTables( tg, tbl );;
gap> tblfustom:= Permuted( tgfustom, trans.columns );;
gap> orbits:= List( reps, map -> OrbitFusions( AutomorphismsOfTable( tbl ),
>                                               map, Group( () ) ) );;
gap> PositionProperty( orbits, orb -> tblfustom in orb );
2
gap> PositionProperty( orbits, orb -> FusionToTom( tbl ).map in orb );
2
```

So we see that the second one of the possibilities above is the right one.

9.7.11 $3 \times U_4(2) \to 3_1.U_4(3)$ (**March 2010**)

According to the Atlas (see [CCN+85, p. 52]), the simple group $U_4(3)$ contains two classes of maximal subgroups of the type $U_4(2)$. The class fusion of $U_4(2)$ into $U_4(3)$ is unique up to table automorphisms.

```
———————— Example ————————
gap> u42:= CharacterTable( "U4(2)" );;
gap> u43:= CharacterTable( "U4(3)" );;
gap> u42fusu43:= PossibleClassFusions( u42, u43 );;
gap> Length( u42fusu43 );
4
gap> Length( RepresentativesFusions( u42, u42fusu43, u43 ) );
1
```

More precisely, take the outer automorphism group of $U_4(3)$, which is a dihedral group of order eight, and consider the subgroup generated by its central involution (this automorphism is denoted by 2_1 in the Atlas) and another involution called 2_3 in the Atlas. This subgroup is a Klein four group that induces a permutation group on the classes of $U_4(3)$ and thus acts on the four possible class fusions of $U_4(2)$ into $U_4(3)$. In fact, this action is transitive.

The automorphism 2_1 swaps each pair of mutually inverse classes of order nine, that is, 9A is swapped with 9B and 9C is swapped with 9D. All $U_4(2)$ type subgroups of $U_4(3)$ are invariant under this automorphism, they extend to subgroups of the type $U_4(2).2$ in $U_4(3).2_1$.

```
———————— Example ————————
gap> u43_21:= CharacterTable( "U4(3).2_1" );;
gap> fus1:= GetFusionMap( u43, u43_21 );
[ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 16, 17, 17,
  18 ]
gap> act1:= Filtered( InverseMap( fus1 ), IsList );
[ [ 16, 17 ], [ 18, 19 ] ]
gap> CompositionMaps( ClassNames( u43, "Atlas" ), act1 );
[ [ "9A", "9B" ], [ "9C", "9D" ] ]
```

The automorphism 2_3 swaps 6B with 6C, 9A with 9C, and 9B with 9D. The two classes of $U_4(2)$ type subgroups of $U_4(3)$ are swapped by this automorphism.

```
———————— Example ————————
gap> u43_23:= CharacterTable( "U4(3).2_3" );;
gap> fus3:= GetFusionMap( u43, u43_23 );
[ 1, 2, 3, 4, 4, 5, 6, 7, 8, 9, 10, 10, 11, 11, 12, 13, 14, 13, 14,
```

```
    15 ]
gap> act3:= Filtered( InverseMap( fus3 ), IsList );
[ [ 4, 5 ], [ 11, 12 ], [ 13, 14 ], [ 16, 18 ], [ 17, 19 ] ]
gap> CompositionMaps( ClassNames( u43, "Atlas" ), act3 );
[ [ "3B", "3C" ], [ "6B", "6C" ], [ "7A", "7B" ], [ "9A", "9C" ],
  [ "9B", "9D" ] ]
```

The **Atlas** states that the permutation character induced by the first class of $U_4(2)$ type subgroups is 1a+35a+90a, which means that the subgroups in this class contain 9A and 9B elements. Then the permutation character induced by the second class of $U_4(2)$ type subgroups is 1a+35b+90a, and the subgroups in this class contain 9C and 9D elements.

So we choose appropriate fusions for the two classes of maximal $U_4(2)$ type subgroups.

```
――――――――――――― Example ―――――――――――――
gap> firstfus:= First( u42fusu43, x -> IsSubset( x, [ 16, 17 ] ) );
[ 1, 2, 2, 3, 3, 5, 4, 7, 8, 9, 10, 10, 12, 12, 11, 12, 16, 17, 20,
  20 ]
gap> secondfus:= First( u42fusu43, x -> IsSubset( x, [ 18, 19 ] ) );
[ 1, 2, 2, 3, 3, 4, 5, 7, 8, 9, 10, 10, 11, 11, 12, 11, 18, 19, 20,
  20 ]
```

Let us now consider the central extension $3_1.U_4(3)$. Since the Schur multiplier of $U_4(2)$ has order two, the $U_4(2)$ type subgroups of $U_4(3)$ lift to groups of the structure $3 \times U_4(2)$ in $3_1.U_4(3)$. There are eight possible class fusions from $3 \times U_4(2)$ to $3_1.U_4(3)$, in two orbits of length four under the action of table automorphisms.

```
――――――――――――― Example ―――――――――――――
gap> 3u42:= CharacterTable( "Cyclic", 3 ) * u42;
CharacterTable( "C3xU4(2)" )
gap> 3u43:= CharacterTable( "3_1.U4(3)" );
CharacterTable( "3_1.U4(3)" )
gap> 3u42fus3u43:= PossibleClassFusions( 3u42, 3u43 );;
gap> Length( 3u42fus3u43 );
8
gap> Length( RepresentativesFusions( 3u42, 3u42fus3u43, 3u43 ) );
2
```

More precisely, each of the four fusions from $U_4(2)$ to $U_4(3)$ has exactly two lifts. The four lifts of those fusions from $U_4(2)$ to $U_4(3)$ with 9A and 9B in their image form one orbit under the action of table automorphisms. The other orbit consists of the lifts of those fusions with 9C and 9D in their image.

```
――――――――――――― Example ―――――――――――――
gap> inducedmaps:= List( 3u42fus3u43, map -> CompositionMaps(
>        GetFusionMap( 3u43, u43 ), CompositionMaps( map,
>        InverseMap( GetFusionMap( 3u42, u42 ) ) ) ) );;
gap> List( inducedmaps, map -> Position( u42fusu43, map ) );
[ 1, 1, 2, 2, 4, 4, 3, 3 ]
```

This solves the ambiguity: Fusions from each of the two orbits occur, and we can assign them to the two classes of subgroups by the choice of the fusions from $U_4(2)$ to $U_4(3)$.

The reason for the asymmetry is that the automorphism 2_3 of $U_4(3)$ does not lift to $3_1.U_4(3)$. Note that each of the classes 9A, 9B of $U_4(3)$ has three preimages in $3_1.U_4(3)$, whereas each of the classes 9C, 9D has only one preimage.

In fact the two classes of $3 \times U_4(2)$ type subgroups of $3_1.U_4(3)$ behave differently. For example, inducing the irreducible characters of a $3 \times U_4(2)$ type subgroup in the first class of maximal subgroups of $3_1.U_4(3)$ yields no irreducible character, whereas the two irreducible characters of degree 630 are obtained by inducing the irreducible characters of a subgroup in the second class.

```
———————————————————————— Example ————————————————————————
gap> rep:= RepresentativesFusions( 3u42, 3u42fus3u43, 3u43 );
[ [ 1, 4, 4, 7, 7, 10, 13, 15, 18, 21, 24, 24, 27, 27, 30, 27, 48,
      49, 50, 50, 2, 5, 5, 8, 8, 11, 13, 16, 19, 22, 25, 25, 28, 28,
      31, 28, 48, 49, 51, 51, 3, 6, 6, 9, 9, 12, 13, 17, 20, 23, 26,
      26, 29, 29, 32, 29, 48, 49, 52, 52 ],
  [ 1, 4, 4, 8, 9, 13, 10, 15, 18, 21, 25, 26, 31, 32, 27, 30, 46,
      44, 51, 52, 2, 5, 5, 9, 7, 13, 11, 16, 19, 22, 26, 24, 32, 30,
      28, 31, 47, 42, 52, 50, 3, 6, 6, 7, 8, 13, 12, 17, 20, 23, 24,
      25, 30, 31, 29, 32, 45, 43, 50, 51 ] ]
gap> irr:= Irr( 3u42 );;
gap> ind:= InducedClassFunctionsByFusionMap( 3u42, 3u43, irr, rep[1] );;
gap> Intersection( ind, Irr( 3u43 ) );
[ Character( CharacterTable( "3_1.U4(3)" ),
  [ 630, 630*E(3)^2, 630*E(3), 6, 6*E(3)^2, 6*E(3), 9, 9*E(3)^2,
      9*E(3), -9, -9*E(3)^2, -9*E(3), 0, 0, 2, 2*E(3)^2, 2*E(3), -2,
      -2*E(3)^2, -2*E(3), 0, 0, 0, -3, -3*E(3)^2, -3*E(3), 3,
      3*E(3)^2, 3*E(3), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, -1, -E(3)^2, -E(3) ] ),
  Character( CharacterTable( "3_1.U4(3)" ),
  [ 630, 630*E(3), 630*E(3)^2, 6, 6*E(3), 6*E(3)^2, 9, 9*E(3),
      9*E(3)^2, -9, -9*E(3), -9*E(3)^2, 0, 0, 2, 2*E(3), 2*E(3)^2,
      -2, -2*E(3), -2*E(3)^2, 0, 0, 0, -3, -3*E(3), -3*E(3)^2, 3,
      3*E(3), 3*E(3)^2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, -1, -E(3), -E(3)^2 ] ) ]
gap> ind:= InducedClassFunctionsByFusionMap( 3u42, 3u43, irr, rep[2] );;
gap> Intersection( ind, Irr( 3u43 ) );
[  ]
```

For $6_1.U_4(3)$ and $12_1.U_4(3)$, one gets the same phenomenon: We have two orbits of class fusions, one corresponding to each of the two classes of subgroups of the type $3 \times 4Y2.U_4(2)$. We get 10 irreducible induced characters from a subgroup in the second class (four faithful ones, four with kernel of order two, and the two abovementioned degree 630 characters with kernel of order four) and no irreducible character from a subgroup in the first class.

9.7.12 $2.3^4.2^3.S_4 \rightarrow 2.A12$ (September 2011)

The double cover G of the alternating group A_{12} contains a maximal subgroup M of the type $2.3^4.2^3.S_4$ whose class fusion is ambiguous.

```
———————————————————————— Example ————————————————————————
gap> 2a12:= CharacterTable( "2.A12" );;
gap> mtbl:= CharacterTable( "2.3^4.2^3.S4" );;
gap> mtblfus2a12:= PossibleClassFusions( mtbl, 2a12 );;
```

```
gap> Length( mtblfus2a12 );
32
gap> repres:= RepresentativesFusions( mtbl, mtblfus2a12, 2a12 );;
gap> Length( repres );
2
```

We decide the question which of the essentially different two possible class fusion is the right one, by explicitly constructing M as a subgroup of G.

For that, let π denote the natural epimorphism from G to A_{12}, and note that $\pi(M)$ can be described as the intersection of a $S_3 \wr S_4$ type subgroup of S_{12} with A_{12}. Further note that the generators for G and A_{12} provided by [WWT$^+$] are compatible in the sense that π can be defined by mapping the generators of G to those of A_{12}.

We need π only for computing one preimage of each given element. Therefore, we represent π implicitly by two epimorphisms from a free group to G and A_{12}, respectively, in order to avoid that GAPprecomputes a lot of unnecessary information for G. This way, computing a preimage of an element of A_{12} under π is cheap. However, computing the preimage of a subgroup of A_{12} would be very expensive. So we construct the subgroup of G that is generated by preimages of a set of generators of $\pi(M)$; later we see that this subgroup is in fact equal to M.

```
————————————————————————— Example ——————————————————————————
  gap> g:= AtlasGroup( "A12" );
  Group([ (1,2,3), (2,3,4,5,6,7,8,9,10,11,12) ])
  gap> 2g:= AtlasGroup( "2.A12" );
  <matrix group of size 479001600 with 2 generators>
  gap> f:= FreeGroup( 2 );;
  gap> pi1:= GroupHomomorphismByImagesNC( f, 2g, GeneratorsOfGroup( f ),
  >             GeneratorsOfGroup( 2g ) );;
  gap> pi2:= GroupHomomorphismByImagesNC( f, g, GeneratorsOfGroup( f ),
  >             GeneratorsOfGroup( g ) );;
  gap> w:= WreathProduct( SymmetricGroup( 3 ), SymmetricGroup(4) );
  <permutation group of size 31104 with 10 generators>
  gap> NrMovedPoints( w );
  12
  gap> s:= Intersection( w, g );  Size( s );
  <permutation group with 8 generators>
  15552
  gap> m:= SubgroupNC( 2g, List( SmallGeneratingSet( s ),
  >           x -> ImagesRepresentative( pi1,
  >                  PreImagesRepresentative( pi2, x ) ) ) );;
```

Now we compute the character table of M, using a faithful permutation representation of M.

```
————————————————————————— Example ——————————————————————————
  gap> iso:= IsomorphismPermGroup( m );;
  gap> t:= CharacterTable( Image( iso ) );;
  gap> Size( t );
  31104
  gap> trans:= TransformingPermutationsCharacterTables( mtbl, t );;
  gap> IsRecord( trans );
  true
```

Now let us see where the two fusion candidates differ.

```
                                  Example
  gap> para:= Parametrized( repres );
  [ 1, 2, 6, 10, 8, 12, 7, 11, 9, 13, 5, 5, 17, 17, 17, 17, 3, 4, 24,
    22, 27, 25, 12, 10, 13, 11, 28, 29, 35, 37, 39, 36, 38, 40, 5, 23,
    28, 29, 26, 14, 14, 16, 16, 33, 34, [ 33, 34 ], [ 33, 34 ], 49, 49,
    48, 48 ]
  gap> PositionsProperty( para, IsList );
  [ 46, 47 ]
  gap> List( repres, map -> map{ [ 44 .. 47 ] } );
  [ [ 33, 34, 33, 34 ], [ 33, 34, 34, 33 ] ]
```

So the question is whether the elements in class 44 are conjugate in *G* to the elements in class 46 or in class 47. In order to answer this question, we compute preimages of the relevant class representatives in the matrix group *M*.

```
                                  Example
  gap> positions:= OnTuples( [ 44 .. 47 ], trans.columns );;
  gap> classreps:= List( ConjugacyClasses( t ){ positions },
  >          c -> PreImagesRepresentative( iso, Representative( c ) ) );;
  gap> traces:= List( classreps, TraceMat );;
  gap> List( traces, x -> Position( traces, x ) );
  [ 1, 2, 2, 1 ]
```

We are lucky, already the traces of the elements allow us to decide which pairs of elements are *G*-conjugate; there is no need for an explicit (and expensive) conjugacy test in the matrix group *G*.

Finally, we check whether the stored fusion is correct.

```
                                  Example
  gap> good:= First( repres,
  >                  map -> map{ [ 44 .. 47 ] } = [ 33, 34, 34, 33 ] );;
  gap> GetFusionMap( mtbl, 2a12 ) = good;
  true
```

9.7.13 $127 : 7 \rightarrow L_7(2)$ (**January 2012**)

The simple group $G = L_7(2)$ contains a maximal subgroup *M* of the type $127 : 7$ (the normalizer of an extension field type subgroup $\mathrm{GL}(1, 2^7)$) whose class fusion is ambiguous.

```
                                  Example
  gap> t:= CharacterTable( "L7(2)" );;
  gap> s:= CharacterTable( "127:7" );;
  gap> fus:= PossibleClassFusions( s, t );;
  gap> repr:= RepresentativesFusions( s, fus, t );
  [ [ 1, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
        112, 113, 114, 115, 117, 116, 76, 76, 77, 76, 77, 77 ],
    [ 1, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
        112, 113, 114, 115, 117, 116, 83, 83, 83, 83, 83, 83 ] ]
```

The two fusion candidates differ only for elements of order 7.

```
                                  Example
  gap> diff:= Filtered( [ 1 .. Length( repr[1] ) ],
  >                      i -> repr[1][i] <> repr[2][i] );
```

```
[ 20, 21, 22, 23, 24, 25 ]
gap> OrdersClassRepresentatives( s ){ diff };
[ 7, 7, 7, 7, 7, 7 ]
gap> List( repr, l -> l{ diff } );
[ [ 76, 76, 77, 76, 77, 77 ], [ 83, 83, 83, 83, 83, 83 ] ]
gap> SizesCentralizers( t ){ [ 76, 77, 83 ] };
[ 3528, 3528, 49 ]
```

We can decide which candidate is the correct one if we know the centralizer order in G of the elements of order 7 in *M*. So we compute this centralizer order.

```
─────────────────── Example ───────────────────
gap> g:= Image( IsomorphismPermGroup( GL(7,2) ) );;
gap> repeat x:= Random( g ); until Order(x) = 127;
gap> n:= Normalizer( g, SubgroupNC( g, [ x ] ) );;
gap> Size( n ) / 127;
7
gap> repeat x:= Random( n ); until Order( x ) = 7;
gap> c:= Centralizer( g, x );;
gap> Size( c );
49
```

We see that the second candidate is the fusion from *M* into *G*.

```
─────────────────── Example ───────────────────
gap> GetFusionMap( s, t ) = repr[2];
true
```

9.7.14 $L_2(59) \to M$ (May 2009) – Do not use this!

It had been claimed that the sporadic simple Monster group *M* contains a maximal subgroup *G* of the type $L_2(59)$, see [HW04]. Later it was shown that *M* does *not* contain such subgroups, see [DLPP24].

Here we show that the character tables of *G* and *M* do not provide enough information to determine a unique class fusion. (In particular, the tables do not help to show that *G* is not a subgroup of *M*.) The class fusion of *G* into *M* is ambiguous.

```
─────────────────── Example ───────────────────
gap> t:= CharacterTable( "M" );;
gap> s:= CharacterTable( "L2(59)" );;
gap> fus:= PossibleClassFusions( s, t );;
gap> repr:= RepresentativesFusions( s, fus, t );
[ [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
      97, 98, 52, 32, 52, 14, 12, 98, 52, 32, 5, 98, 12, 98, 52, 3 ],
  [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
      97, 100, 50, 30, 50, 15, 11, 100, 50, 30, 4, 100, 11, 100, 50,
      3 ],
  [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
      97, 101, 51, 30, 51, 14, 11, 101, 51, 30, 5, 101, 11, 101, 51,
      3 ],
  [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
      97, 102, 53, 32, 53, 18, 12, 102, 53, 32, 6, 102, 12, 102, 53,
      3 ],
  [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
```

```
        97, 104, 52, 33, 52, 17, 12, 104, 52, 33, 5, 104, 12, 104, 52,
        3 ] ]
```

The candidates differ on the classes of element order 30.

```
 ─────────────────────────────── Example ───────────────────────────────
 gap> ord:= OrdersClassRepresentatives( s );;
 gap> ord30:= Filtered( [ 1 .. Length( ord ) ], i -> ord[i] = 30 );
 [ 18, 24, 28, 30 ]
 gap> List( repr, x -> x{ ord30 } );
 [ [ 98, 98, 98, 98 ], [ 100, 100, 100, 100 ], [ 101, 101, 101, 101 ],
   [ 102, 102, 102, 102 ], [ 104, 104, 104, 104 ] ]
```

According to [HW04], any subgroup *G* of *M* must contain elements in the class 30G of *M*. This determines the class fusion up to Galois automorphisms. (But note that we do not store the fusion, since *G* is not a subgroup of *M*.)

```
 ─────────────────────────────── Example ───────────────────────────────
 gap> pos:= Position( ClassNames( t, "Atlas" ), "30G" );;
 gap> good:= Filtered( fus, map -> pos in map );
 [ [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
       97, 104, 52, 33, 52, 17, 12, 104, 52, 33, 5, 104, 12, 104, 52,
       3 ],
   [ 1, 153, 152, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
       97, 104, 52, 33, 52, 17, 12, 104, 52, 33, 5, 104, 12, 104, 52,
       3 ] ]
 gap> repr:= RepresentativesFusions( s, good, t );
 [ [ 1, 152, 153, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97, 97,
       97, 104, 52, 33, 52, 17, 12, 104, 52, 33, 5, 104, 12, 104, 52,
       3 ] ]
 gap> GetFusionMap( s, t ) = fail;
 true
```

9.7.15 $L_2(71) \rightarrow M$ (May 2009)

The sporadic simple Monster group *M* contains a maximal subgroup *G* of the type $L_2(71)$, see [HW08]. The class fusion of *G* into *M* is ambiguous.

```
 ─────────────────────────────── Example ───────────────────────────────
 gap> t:= CharacterTable( "M" );;
 gap> s:= CharacterTable( "L2(71)" );;
 gap> fus:= PossibleClassFusions( s, t );;
 gap> repr:= RepresentativesFusions( s, fus, t );
 [ [ 1, 169, 170, 112, 112, 112, 112, 19, 112, 11, 112, 112, 19, 112,
       112, 112, 11, 19, 112, 112, 114, 60, 36, 27, 114, 17, 114, 27,
       7, 60, 114, 5, 114, 60, 36, 27, 114, 3 ],
   [ 1, 169, 170, 112, 112, 112, 112, 19, 112, 11, 112, 112, 19, 112,
       112, 112, 11, 19, 112, 112, 115, 61, 36, 28, 115, 17, 115, 28,
       7, 61, 115, 5, 115, 61, 36, 28, 115, 3 ],
   [ 1, 169, 170, 112, 112, 112, 112, 19, 112, 11, 112, 112, 19, 112,
       112, 112, 11, 19, 112, 112, 117, 61, 43, 28, 117, 17, 117, 28,
       9, 61, 117, 5, 117, 61, 43, 28, 117, 3 ],
   [ 1, 169, 170, 113, 113, 113, 113, 20, 113, 12, 113, 113, 20, 113,
```

```
       113, 113, 12, 20, 113, 113, 114, 60, 36, 27, 114, 17, 114, 27,
       7, 60, 114, 5, 114, 60, 36, 27, 114, 3 ],
  [ 1, 169, 170, 113, 113, 113, 113, 20, 113, 12, 113, 113, 20, 113,
       113, 113, 12, 20, 113, 113, 115, 61, 36, 28, 115, 17, 115, 28,
       7, 61, 115, 5, 115, 61, 36, 28, 115, 3 ],
  [ 1, 169, 170, 113, 113, 113, 113, 20, 113, 12, 113, 113, 20, 113,
       113, 113, 12, 20, 113, 113, 117, 61, 43, 28, 117, 17, 117, 28,
       9, 61, 117, 5, 117, 61, 43, 28, 117, 3 ] ]
```

The candidates differ on the classes of the element orders 7 and 36.

```
────────────────────────────── Example ──────────────────────────────
gap> ord:= OrdersClassRepresentatives( s );;
gap> ord36:= Filtered( [ 1 .. Length( ord ) ], i -> ord[i] = 36 );
[ 21, 25, 27, 31, 33, 37 ]
gap> List( repr, x -> x{ ord36 } );
[ [ 114, 114, 114, 114, 114, 114 ], [ 115, 115, 115, 115, 115, 115 ],
  [ 117, 117, 117, 117, 117, 117 ], [ 114, 114, 114, 114, 114, 114 ],
  [ 115, 115, 115, 115, 115, 115 ], [ 117, 117, 117, 117, 117, 117 ] ]
```

According to [NW02, Table 3], G contains elements in the classes 7B and 36D of M. This determines the class fusion up to Galois automorphisms.

```
────────────────────────────── Example ──────────────────────────────
gap> pos1:= Position( ClassNames( t, "Atlas" ), "7B" );;
gap> pos2:= Position( ClassNames( t, "Atlas" ), "36D" );;
gap> pos:= [ pos1, pos2 ];;
gap> good:= Filtered( fus, map -> IsSubset( map, pos ) );
[ [ 1, 169, 170, 113, 113, 113, 113, 20, 113, 12, 113, 113, 20, 113,
       113, 113, 12, 20, 113, 113, 117, 61, 43, 28, 117, 17, 117, 28,
       9, 61, 117, 5, 117, 61, 43, 28, 117, 3 ],
   [ 1, 170, 169, 113, 113, 113, 113, 20, 113, 12, 113, 113, 20, 113,
       113, 113, 12, 20, 113, 113, 117, 61, 43, 28, 117, 17, 117, 28,
       9, 61, 117, 5, 117, 61, 43, 28, 117, 3 ] ]
gap> repr:= RepresentativesFusions( s, good, t );
[ [ 1, 169, 170, 113, 113, 113, 113, 20, 113, 12, 113, 113, 20, 113,
       113, 113, 12, 20, 113, 113, 117, 61, 43, 28, 117, 17, 117, 28,
       9, 61, 117, 5, 117, 61, 43, 28, 117, 3 ] ]
gap> GetFusionMap( s, t ) = repr[1];
true
```

9.7.16 $L_2(41) \to M$ (April 2012)

The sporadic simple Monster group M contains a maximal subgroup G of the type $L_2(41)$, see [NW13]. The class fusion of G into M is ambiguous.

```
────────────────────────────── Example ──────────────────────────────
gap> t:= CharacterTable( "M" );;
gap> s:= CharacterTable( "L2(41)" );;
gap> fus:= PossibleClassFusions( s, t );;
gap> repr:= RepresentativesFusions( s, fus, t );
[ [ 1, 127, 127, 64, 30, 64, 11, 7, 30, 64, 11, 64, 3, 70, 70, 19,
       70, 70, 19, 4, 70, 19, 70 ],
```

```
  [ 1, 127, 127, 64, 30, 64, 11, 7, 30, 64, 11, 64, 3, 72, 72, 19,
      72, 72, 19, 6, 72, 19, 72 ],
  [ 1, 127, 127, 64, 30, 64, 11, 7, 30, 64, 11, 64, 3, 73, 73, 20,
      73, 73, 20, 5, 73, 20, 73 ],
  [ 1, 127, 127, 66, 33, 66, 12, 7, 33, 66, 12, 66, 3, 72, 72, 19,
      72, 72, 19, 6, 72, 19, 72 ],
  [ 1, 127, 127, 66, 33, 66, 12, 7, 33, 66, 12, 66, 3, 73, 73, 20,
      73, 73, 20, 5, 73, 20, 73 ],
  [ 1, 127, 127, 67, 30, 67, 11, 10, 30, 67, 11, 67, 3, 72, 72, 19,
      72, 72, 19, 6, 72, 19, 72 ],
  [ 1, 127, 127, 67, 30, 67, 11, 10, 30, 67, 11, 67, 3, 73, 73, 20,
      73, 73, 20, 5, 73, 20, 73 ],
  [ 1, 127, 127, 68, 32, 68, 12, 10, 32, 68, 12, 68, 3, 72, 72, 19,
      72, 72, 19, 6, 72, 19, 72 ],
  [ 1, 127, 127, 68, 32, 68, 12, 10, 32, 68, 12, 68, 3, 73, 73, 20,
      73, 73, 20, 5, 73, 20, 73 ],
  [ 1, 127, 127, 69, 33, 69, 12, 9, 33, 69, 12, 69, 3, 72, 72, 19,
      72, 72, 19, 6, 72, 19, 72 ],
  [ 1, 127, 127, 69, 33, 69, 12, 9, 33, 69, 12, 69, 3, 73, 73, 20,
      73, 73, 20, 5, 73, 20, 73 ] ]
```

The candidates differ on the classes of the element orders 3–8.

```
  ─────────────────────────── Example ───────────────────────────
gap> ambig:= Parametrized( repr );;
gap> ambigpos:= PositionsProperty( ambig, IsList );
[ 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22,
  23 ]
gap> Set( OrdersClassRepresentatives( t ){ ambigpos } );
[ 3, 4, 5, 6, 7, 8 ]
```

According to [NW13, Theorem 3], *G* contains elements in the classes 3B and 4C of *M*. This determines the class fusion uniquely.

```
  ─────────────────────────── Example ───────────────────────────
gap> pos1:= Position( ClassNames( t, "Atlas" ), "3B" );;
gap> pos2:= Position( ClassNames( t, "Atlas" ), "4C" );;
gap> pos:= [ pos1, pos2 ];;
gap> good:= Filtered( fus, map -> IsSubset( map, pos ) );
[ [ 1, 127, 127, 69, 33, 69, 12, 9, 33, 69, 12, 69, 3, 73, 73, 20,
      73, 73, 20, 5, 73, 20, 73 ] ]
gap> GetFusionMap( s, t ) = good[1];
true
```

Chapter 10

GAP computations needed in the proof of [DNT13, Theorem 6.1 (ii)]

Date: September 19th, 2011

    (This is joint work with Klaus Lux.)

    This is a collection of example computations that are cited in the Appendix of [DNT13]. In each case, the aim is to show that the extension of a given finite simple group by an elementary abelian group of given rank has the property that not all complex irreducible characters of the same degree are Galois conjugate.

    The purpose of this writeup is twofold. On the one hand, the details of the computations are documented this way. On the other hand, the **GAP** code shown for the examples can be used as test input for automatic checking of the data and the functions used.} For the computations, we need some Brauer character tables from [JLPW95], some generating matrices from [WWT$^+$], and some functions from the **GAP** system [GAP24] and its packages `AtlasRep` [WPN$^+$22], `cohomolo` [Hol08], `CTblLib` [Bre25], and `TomLib` [MNP19].

    First we load the necessary **GAP** packages.

```
_____ Example _____
  gap> LoadPackage( "AtlasRep", "1.5", false );
  true
  gap> LoadPackage( "cohomolo", "1.6", false );
  true
  gap> LoadPackage( "CTblLib", "1.2", false );
  true
  gap> LoadPackage( "TomLib", "1.2.1", false );
  true
```

10.1 $G/N \cong Sz(8)$ **and** $|N| = 2^{12}$

The group $S = Sz(8)$ has exactly one irreducible 12-dimensional module over the field with two elements, up to isomorphism. This module can be obtained from any of the three absolutely irreducible 4-dimensional S-modules in characteristic two, by regarding it as a module over the prime field $GF(2)$.

```
_____ Example _____
  gap> p:= 2;;  d:= 12;;
  gap> t:= CharacterTable( "Sz(8)" ) mod p;
```

```
BrauerTable( "Sz(8)", 2 )
gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
gap> Display( t, rec( chars:= irr, powermap:= false,
>                       centralizers:= false ) );
Sz(8)mod2

        1a 5a 7a 7b 7c 13a 13b 13c

Y.1     1  1  1  1  1   1   1   1
Y.2     4 -1  A  C  B   D   F   E
Y.3     4 -1  B  A  C   E   D   F
Y.4     4 -1  C  B  A   F   E   D


A = E(7)^2+E(7)^3+E(7)^4+E(7)^5
B = E(7)+E(7)^2+E(7)^5+E(7)^6
C = E(7)+E(7)^3+E(7)^4+E(7)^6
D = E(13)+E(13)^5+E(13)^8+E(13)^12
E = E(13)^4+E(13)^6+E(13)^7+E(13)^9
F = E(13)^2+E(13)^3+E(13)^10+E(13)^11
gap> List( irr, x -> SizeOfFieldOfDefinition( x, p ) );
[ 2, 8, 8, 8 ]
```

First we construct the 12-dimensional irreducible representation of *S* over $GF(2)$, using that the Atlas of Group Representations provides matrix generators for *S* in the 4-dimensional representation over $GF(8)$.

```
_____ Example _____
gap> info:= OneAtlasGeneratingSetInfo( "Sz(8)", Dimension, 4,
>                 Characteristic, p );
rec( charactername := "4a", constituents := [ 2 ], contents := "core",
  dim := 4, groupname := "Sz(8)", id := "a",
  identifier := [ "Sz(8)", [ "Sz8G1-f8r4aB0.m1", "Sz8G1-f8r4aB0.m2" ],
      1, 8 ], repname := "Sz8G1-f8r4aB0", repnr := 17,
  ring := GF(2^3), size := 29120, standardization := 1,
  type := "matff" )
gap> gens_dim4:= AtlasGenerators( info ).generators;;
gap> b:= Basis( GF(8) );;
gap> gens_dim12:= List( gens_dim4, x -> BlownUpMatrix( b, x ) );;
```

We claim that any extension of *S* with the given module splits.

```
_____ Example _____
gap> s:= AtlasGroup( "Sz(8)", IsPermGroup, true );;
gap> chr:= CHR( s, p, 0, gens_dim12 );;
gap> SizeScreen( [ 100 ] );;
gap> SecondCohomologyDimension( chr );
0
gap> SizeScreen( [ 72 ] );;
```

(The function CHR takes as its arguments a permutation group, the characteristic of the module, a finitely presented group (or zero), and a list of matrices that define the module in the sense that they correspond to the generators of the given permutation group. Note that this condition is satisfied

because the generators provided by the Atlas of Group Representations are compatible.) So it is enough to consider the semidirect product $G = 2^{12} : Sz(8)$. If we would like then we could represent this group as a group of 13×13 matrices over $GF(2)$, as follows. For each element of G, the submatrix consisting of the first 12 rows and columns describes the part from the complement $Sz(8)$, in its action on the module in question, and the last row describes the part from the elementary abelian normal group N; the last column is zero, except for an identity entry in the last row. In order to write down generators of this group, it suffices to take the two generators of the complement plus one nonidentity element from N. (Note that N is irreducible.)

```
──────────────────────── Example ────────────────────────
  gap> mats:= List( [ 1 .. 3 ], x -> IdentityMat( d+1, GF(p) ) );;
  gap> v:= mats[1][ d+1 ];;
  gap> mats[1]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_dim12[1];;
  gap> mats[2]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_dim12[2];;
  gap> mats[3][ d+1 ][1]:= Z(p)^0;;
  gap> grp:= Group( mats );;
  gap> g:= Image( IsomorphismPermGroup( grp ) );;
  gap> Size( g );
  119275520
  gap> NrConjugacyClasses( g );
  41
```

The GAP Character Table Library contains the ordinary character table of G. We check this as follows. By the above cohomology result, the group G is uniquely determined, up to isomorphism, by the group order and the property that G has a minimal normal subgroup N such that G/N is a simple group isomorphic with S.

(Since $|G|/|S|$ is a power of two, N is a 2-group. By the minimality condition, N is elementary abelian and the action of S on N affords the desired S-module. Note that the isomorphism type of a finite simple group is determined by its character table.)

```
──────────────────────── Example ────────────────────────
  gap> iso:= IsomorphismTypeInfoFiniteSimpleGroup( s );
  rec( name := "2B(2,8) = 2C(2,8) = Sz(8)", parameter := 8,
    series := "2B", shortname := "Sz(8)" )
  gap> names:= AllCharacterTableNames( Size, 2^12 * Size( s ) );;
  gap> cand:= List( names, CharacterTable );;
  gap> cand:= Filtered( cand,
  >       t -> ForAny( ClassPositionsOfMinimalNormalSubgroups( t ),
  >            n -> IsomorphismTypeInfoFiniteSimpleGroup( t / n ) = iso ) );
  [ CharacterTable( "2^12:Sz(8)" ) ]
```

So we can easily check that G has eight rational valued irreducibles of the degree 455 (or of the degree 3 640).

```
──────────────────────── Example ────────────────────────
  gap> t:= cand[1];;
  gap> rationals:= Filtered( Irr( t ), x -> IsSubset( Integers, x ) );;
  gap> Collected( List( rationals, x -> x[1] ) );
  [ [ 1, 1 ], [ 64, 1 ], [ 91, 1 ], [ 455, 8 ], [ 3640, 8 ] ]
```

10.2 $G/N \cong M_{22}$ **and** $|N| = 2^{10}$

The group $S = M_{22}$ has exactly two irreducible 10-dimensional modules over the field with two elements, up to isomorphism. These modules are in fact absolutely irreducible.

```
——————————————————————— Example ———————————————————————
  gap> p:= 2;;  d:= 10;;
  gap> t:= CharacterTable( "M22" ) mod p;
  BrauerTable( "M22", 2 )
  gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
  gap> Display( t, rec( chars:= irr, powermap:= false,
  >                     centralizers:= false ) );
  M22mod2

          1a 3a 5a 7a 7b 11a 11b

  Y.1      1  1  1  1  1   1   1
  Y.2     10  1  .  A /A  -1  -1
  Y.3     10  1  . /A  A  -1  -1

  A = E(7)+E(7)^2+E(7)^4
    = (-1+Sqrt(-7))/2 = b7
  gap> List( irr, x -> SizeOfFieldOfDefinition( x, p ) );
  [ 2, 2, 2 ]
```

First we construct the two irreducible 10-dimensional representations of S over $GF(2)$, again using that the Atlas of Group Representations provides the matrix generators in question.

```
——————————————————————— Example ———————————————————————
  gap> info:= AllAtlasGeneratingSetInfos( "M22", Dimension, d,
  >               Characteristic, p );
  [ rec( charactername := "10a", constituents := [ 2 ],
        contents := "core", dim := 10, groupname := "M22", id := "a",
        identifier :=
          [ "M22", [ "M22G1-f2r10aB0.m1", "M22G1-f2r10aB0.m2" ], 1, 2 ],
        repname := "M22G1-f2r10aB0", repnr := 13, ring := GF(2),
        size := 443520, standardization := 1, type := "matff" ),
    rec( charactername := "10b", constituents := [ 3 ],
        contents := "core", dim := 10, groupname := "M22", id := "b",
        identifier :=
          [ "M22", [ "M22G1-f2r10bB0.m1", "M22G1-f2r10bB0.m2" ], 1, 2 ],
        repname := "M22G1-f2r10bB0", repnr := 14, ring := GF(2),
        size := 443520, standardization := 1, type := "matff" ) ]
  gap> gens:= List( info, r -> AtlasGenerators( r ).generators );;
```

We claim that any extension of S with any of the two given modules splits.

```
——————————————————————— Example ———————————————————————
  gap> s:= AtlasGroup( "M22", IsPermGroup, true );;
  gap> chr:= CHR( s, p, 0, gens[1] );;
  gap> SizeScreen( [ 100 ] );;
  gap> SecondCohomologyDimension( chr );
  0
  gap> chr:= CHR( s, p, 0, gens[2] );;
```

```
gap> SecondCohomologyDimension( chr );
0
gap> SizeScreen( [ 72 ] );;
```

Again we see that it is enough to consider semidirect products $G = 2^{10} : M_{22}$, but this time for the two nonisomorphic modules.

We could use the same method as in the first case for constructing the two groups.

```
                         ──── Example ────
gap> gens_1:= gens[1];;
gap> mats:= List( [1 .. 3 ], x -> IdentityMat( d+1, GF(p) ) );;
gap> v:= mats[1][ d+1 ];;
gap> mats[1]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_1[1];;
gap> mats[2]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_1[2];;
gap> mats[3][ d+1 ][1]:= Z(p)^0;;
gap> grp_1:= Group( mats );;
gap> Size( grp_1 );
454164480
gap> gens_2:= gens[1];;
gap> mats:= List( [1 .. 3 ], x -> IdentityMat( d+1, GF(p) ) );;
gap> v:= mats[1][ d+1 ];;
gap> mats[1]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_2[1];;
gap> mats[2]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_2[2];;
gap> mats[3][ d+1 ][1]:= Z(p)^0;;
gap> grp_2:= Group( mats );;
gap> Size( grp_2 );
454164480
```

The GAP Character Table Library contains the ordinary character tables of the two groups in question. We check this with the same approach as in the previous examples.

```
                         ──── Example ────
gap> iso:= IsomorphismTypeInfoFiniteSimpleGroup( s );
rec( name := "M(22)", series := "Spor", shortname := "M22" )
gap> names:= AllCharacterTableNames( Size, 2^10 * Size( s ) );;
gap> cand:= List( names, CharacterTable );;
gap> cand:= Filtered( cand,
>      t -> ForAny( ClassPositionsOfMinimalNormalSubgroups( t ),
>           n -> IsomorphismTypeInfoFiniteSimpleGroup( t / n ) = iso ) );
[ CharacterTable( "2^10:M22'" ), CharacterTable( "2^10:m22" ) ]
gap> List( cand, NrConjugacyClasses );
[ 47, 43 ]
```

So we can easily check that in both cases, G has two rational valued irreducibles of the degree 1155.

```
                         ──── Example ────
gap> t:= cand[1];;
gap> rationals:= Filtered( Irr( t ), x -> IsSubset( Integers, x ) );;
gap> Collected( List( rationals, x -> x[1] ) );
[ [ 1, 1 ], [ 21, 1 ], [ 22, 1 ], [ 55, 1 ], [ 99, 1 ], [ 154, 1 ],
  [ 210, 1 ], [ 231, 3 ], [ 385, 1 ], [ 440, 1 ], [ 770, 5 ],
  [ 924, 2 ], [ 1155, 2 ], [ 1386, 1 ], [ 1408, 1 ], [ 3080, 2 ],
```

```
    [ 3465, 4 ], [ 4620, 2 ], [ 6930, 3 ], [ 9240, 1 ] ]
gap> t:= cand[2];;
gap> rationals:= Filtered( Irr( t ), x -> IsSubset( Integers, x ) );;
gap> Collected( List( rationals, x -> x[1] ) );
[ [ 1, 1 ], [ 21, 1 ], [ 55, 1 ], [ 77, 1 ], [ 99, 1 ], [ 154, 1 ],
  [ 210, 1 ], [ 231, 1 ], [ 330, 1 ], [ 385, 3 ], [ 616, 2 ],
  [ 693, 1 ], [ 770, 1 ], [ 1155, 2 ], [ 1980, 1 ], [ 2310, 4 ],
  [ 2640, 1 ], [ 3465, 2 ], [ 4620, 1 ], [ 5544, 2 ], [ 6160, 1 ],
  [ 6930, 2 ], [ 9856, 1 ] ]
```

10.3 $G/N \cong J_2$ and $|N| = 2^{12}$

The group $S = J_2$ has exactly one irreducible 12-dimensional module over the field with two elements, up to isomorphism. This module can be obtained from any of the two absolutely irreducible 6-dimensional S-modules in characteristic two, by regarding it as a module over the prime field $GF(2)$.

———————————————— Example ————————————————
```
gap> p:= 2;;  d:= 12;;
gap> t:= CharacterTable( "J2" ) mod p;
BrauerTable( "J2", 2 )
gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
gap> Display( t, rec( chars:= irr, powermap:= false,
>                     centralizers:= false ) );
J2mod2


      1a 3a 3b 5a 5b 5c 5d 7a 15a 15b

Y.1    1  1  1  1  1  1  1  1   1   1
Y.2    6 -3  .  A *A  B *B -1   C  *C
Y.3    6 -3  . *A  A *B  B -1  *C   C

A = -2*E(5)-2*E(5)^4
  = 1-Sqrt(5) = 1-r5
B = E(5)+2*E(5)^2+2*E(5)^3+E(5)^4
  = (-3-Sqrt(5))/2 = -2-b5
C = E(5)+E(5)^4
  = (-1+Sqrt(5))/2 = b5
gap> List( irr, x -> SizeOfFieldOfDefinition( x, p ) );
[ 2, 4, 4 ]
```

First we construct the irreducible 12-dimensional representation of S over $GF(2)$, using that the Atlas of Group Representations provides matrix generators for S in the 6-dimensional representation over $GF(4)$.

———————————————— Example ————————————————
```
gap> info:= OneAtlasGeneratingSetInfo( "J2", Dimension, 6,
>             Characteristic, p );
rec( charactername := "6a", constituents := [ 2 ], contents := "core",
  dim := 6, groupname := "J2", id := "a",
  identifier := [ "J2", [ "J2G1-f4r6aB0.m1", "J2G1-f4r6aB0.m2" ], 1,
      4 ], repname := "J2G1-f4r6aB0", repnr := 16, ring := GF(2^2),
```

```
    size := 604800, standardization := 1, type := "matff" )
gap> gens_dim6:= AtlasGenerators( info ).generators;;
gap> b:= Basis( GF(4) );;
gap> gens_dim12:= List( gens_dim6, x -> BlownUpMatrix( b, x ) );;
```

We claim that any extension of *S* with the given module splits.

```
――――――――――――――――――― Example ―――――――――――――――――――
gap> s:= AtlasGroup( "J2", IsPermGroup, true );;
gap> chr:= CHR( s, p, 0, gens_dim12 );;
gap> SizeScreen( [ 100 ] );;
gap> SecondCohomologyDimension( chr );
0
gap> SizeScreen( [ 72 ] );;
```

Again we see that it is enough to consider a semidirect product $G = 2^{12}{:}J_2$.
Here is a description how we could construct the group.

```
――――――――――――――――――― Example ―――――――――――――――――――
gap> mats:= List( [ 1 .. 3 ], x -> IdentityMat( d+1, GF(p) ) );;
gap> v:= mats[1][ d+1 ];;
gap> mats[1]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_dim12[1];;
gap> mats[2]{ [ 1 .. d ] }{ [ 1 .. d ] }:= gens_dim12[2];;
gap> mats[3][ d+1 ][1]:= Z(p)^0;;
gap> grp:= Group( mats );;
gap> g:= Image( IsomorphismPermGroup( grp ) );;
gap> Size( g );
2477260800
```

The GAP Character Table Library contains the ordinary character table of *G*. We check this with the same approach as in the previous examples.

```
――――――――――――――――――― Example ―――――――――――――――――――
gap> iso:= IsomorphismTypeInfoFiniteSimpleGroup( s );
rec( name := "HJ = J(2) = F(5-)", series := "Spor", shortname := "J2"
 )
gap> names:= AllCharacterTableNames( Size, 2^12 * Size( s ) );;
gap> cand:= List( names, CharacterTable );;
gap> cand:= Filtered( cand,
>       t -> ForAny( ClassPositionsOfMinimalNormalSubgroups( t ),
>              n -> IsomorphismTypeInfoFiniteSimpleGroup( t / n ) = iso ) );
[ CharacterTable( "2^12:J2" ) ]
```

So we can easily check that *G* has two rational valued irreducibles of the degree 1575.

```
――――――――――――――――――― Example ―――――――――――――――――――
gap> t:= cand[1];;
gap> rationals:= Filtered( Irr( t ), x -> IsSubset( Integers, x ) );;
gap> Collected( List( rationals, x -> x[1] ) );
[ [ 1, 1 ], [ 36, 1 ], [ 63, 1 ], [ 90, 1 ], [ 126, 1 ], [ 160, 1 ],
  [ 175, 1 ], [ 225, 1 ], [ 288, 1 ], [ 300, 1 ], [ 336, 1 ],
  [ 1575, 2 ], [ 2520, 4 ], [ 3150, 1 ], [ 4725, 6 ], [ 9450, 1 ],
  [ 10080, 4 ], [ 12600, 4 ], [ 18900, 2 ] ]
```

10.4 $G/N \cong J_2$ and $|N| = 5^{14}$

The group $S = J_2$ has exactly one irreducible 14-dimensional module over the field with 5 elements, up to isomorphism. This module is in fact absolutely irreducible.

```
───────────── Example ─────────────
gap> p:= 5;;  d:= 14;;
gap> t:= CharacterTable( "J2" ) mod p;
BrauerTable( "J2", 5 )
gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
gap> Display( t, rec( chars:= irr, powermap:= false,
>                     centralizers:= false ) );
J2mod5


      1a 2a 2b 3a 3b 4a 6a 6b 7a 8a 12a

Y.1    1  1  1  1  1  1  1  1  1  1   1
Y.2   14 -2  2  5 -1  2  1 -1  .  .  -1
```

In this case, we do not attempt to compute the complete character table of G. Instead, we show that G/N has at least five regular orbits on the dual space of N, and apply \cite[Lemma 5.1 (i)]{DNT}. (Note that N is in fact self-dual.)

For that, we use GAP's table of marks of S. The information stored for this table of marks allows us to compute, for each class of subgroups U of S, the numbers of orbits in the dual space of N for which contain the point stabilizers in S are exactly the conjugates of U. The following GAP function takes the table of marks `tom` of S, a list `matgens` of matrices that describe the action of the generators of S on the vector space in question, and the size `q` of its field of scalars. The return value is a record with the components `fixed` (the vector of numbers of fixed points of the subgroups of S on the dual of N), `decomp` (the numbers of orbits with the corresponding point stabilizers), `nonzeropos` (the positions of subgroups that occur as point stabilizers), and `staborders` (the list of orders of the subgroups that occur as point stabilizers).

```
───────────── Example ─────────────
gap> orbits_from_tom:= function( tom, matgens, q )
>     local slp, fixed, idmat, i, rest, decomp, nonzeropos;
>
>     slp:= StraightLineProgramsTom( tom );
>     fixed:= [];
>     idmat:= matgens[1]^0;
>     for i in [ 1 .. Length( slp ) ] do
>       if IsList( slp[i] ) then
>         # Each subgroup generator has a program of its own.
>         rest:= List( slp[i],
>                   prg -> ResultOfStraightLineProgram( prg, gens ) );
>       else
>         # The subgroup generators are computed with one common program.
>         rest:= ResultOfStraightLineProgram( slp[i], gens );
>       fi;
>       if IsEmpty( rest ) then
>         # The subgroup is trivial.
>         fixed[i]:= q^Length( idmat );
>       else
```

```
>          # Compute the intersection of fixed spaces of the transposed
>          # matrices, since we act on Irr(N) not on N.
>          fixed[i]:= q^Length( NullspaceMat( TransposedMat( Concatenation(
>                       List( rest, x -> x - idmat ) ) ) ) );
>        fi;
>      od;
>
>      decomp:= DecomposedFixedPointVector( tom, fixed );
>      nonzeropos:= Filtered( [ 1 .. Length( decomp ) ],
>                             i -> decomp[i] <> 0 );
>
>      return rec( fixed:= fixed,
>                  decomp:= decomp,
>                  nonzeropos:= nonzeropos,
>                  staborders:= OrdersTom( tom ){ nonzeropos },
>                );
> end;;
```

Note that this function assumes that the generators of *S* obtained from the Atlas of Group Representations are compatible with the generators from GAP's table of marks of *S*. This fact can be read off from the `true` value of the `ATLAS` component in the `StandardGeneratorsInfo` (**TomLib: StandardGeneratorsInfo for groups**) value of the table of marks.

```
——————————————————— Example ———————————————————
gap> tom:= TableOfMarks( "J2" );
TableOfMarks( "J2" )
gap> StandardGeneratorsInfo( tom );
[ rec( ATLAS := true,
      description := "|z|=10, z^5=a, |b|=3, |C(b)|=36, |ab|=7",
      generators := "a, b",
      script :=
        [ [ 1, 10, 5 ], [ 2, 3 ], [ [ 2, 1 ], [ "|C(",, ")|" ], 36 ],
          [ 1, 1, 2, 1, 7 ] ], standardization := 1 ) ]
```

Alternatively, we can compute whether the generators are compatible, as follows.

```
——————————————————— Example ———————————————————
gap> info:= OneAtlasGeneratingSetInfo( "J2", Dimension, d, Ring, GF(p) );
rec( charactername := "14a", constituents := [ 2 ],
  contents := "core", dim := 14, givenRing := GF(5),
  groupname := "J2", id := "",
  identifier := [ "J2", [ "J2G1-f5r14B0.m1", "J2G1-f5r14B0.m2" ], 1,
      5 ], repname := "J2G1-f5r14B0", repnr := 19, ring := GF(5),
  size := 604800, standardization := 1, type := "matff" )
gap> gens:= AtlasGenerators( info ).generators;;
gap> map:= GroupGeneralMappingByImages( UnderlyingGroup( tom ),
>     Group( gens ), GeneratorsOfGroup( UnderlyingGroup( tom ) ), gens );;
gap> IsGroupHomomorphism( map );
true
```

Now we are sure that we may apply the function `orbits_from_tom`.

```
                              ─── Example ───
 gap> orbits_from_tom( tom, gens, p );
 rec(
   decomp := [ 8600, 0, 2512, 359, 10, 0, 0, 212, 5, 0, 0, 4, 0, 240,
       16, 10, 0, 0, 0, 0, 10, 0, 0, 0, 0, 2, 0, 0, 36, 0, 0, 0, 26,
       0, 0, 0, 0, 0, 0, 0, 20, 0, 10, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 10, 0, 0, 5, 0, 0, 0, 26, 0, 10, 0, 0, 0, 0, 10, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 2, 0,
       0, 0, 0, 2, 4, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 4, 0, 0, 0, 4, 0, 0, 1 ],
   fixed := [ 6103515625, 15625, 390625, 390625, 625, 25, 3125, 3125,
       625, 625, 625, 625, 5, 3125, 125, 625, 25, 25, 125, 5, 125, 25,
       125, 25, 25, 25, 5, 125, 125, 125, 25, 25, 3125, 1, 1, 5, 5,
       25, 5, 25, 125, 5, 25, 25, 25, 25, 25, 25, 5, 25, 25, 5, 25, 5,
       5, 5, 5, 25, 25, 1, 125, 1, 5, 5, 125, 1, 25, 5, 25, 1, 5, 25,
       5, 5, 25, 25, 5, 5, 5, 1, 5, 5, 1, 1, 1, 5, 1, 25, 25, 25, 1,
       5, 25, 5, 5, 1, 1, 125, 5, 5, 5, 25, 5, 5, 5, 1, 1, 5, 5, 1, 5,
       1, 5, 1, 1, 25, 5, 5, 1, 1, 1, 1, 5, 1, 1, 25, 1, 1, 5, 1, 1,
       5, 1, 5, 1, 1, 5, 1, 5, 1, 1, 1, 5, 1, 1, 1 ],
   nonzeropos := [ 1, 3, 4, 5, 8, 9, 12, 14, 15, 16, 21, 26, 29, 33,
       41, 43, 44, 58, 61, 65, 67, 72, 89, 93, 98, 99, 105, 116, 126,
       139, 143, 146 ],
   staborders := [ 1, 2, 3, 3, 4, 4, 5, 6, 6, 6, 8, 9, 10, 12, 12, 12,
       14, 20, 24, 24, 24, 30, 48, 50, 60, 60, 72, 120, 192, 600,
       1920, 604800 ] )
```

We see that *S* has 8600 regular orbits on (the dual space of) *N*.

10.5 $G/N \cong J_2$ and $|N| = 2^{28}$

The group $S = J_2$ has exactly one irreducible 28-dimensional module over the field with two elements, up to isomorphism. This module can be obtained from any of the two absolutely irreducible 14-dimensional *S*-modules in characteristic two, by regarding it as a module over the prime field $GF(2)$.

```
                              ─── Example ───
 gap> p:= 2;;   d:= 28;;
 gap> t:= CharacterTable( "J2" ) mod p;
 BrauerTable( "J2", 2 )
 gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
 gap> Display( t, rec( chars:= irr, powermap:= false,
 >                     centralizers:= false ) );
 J2mod2

         1a 3a 3b 5a 5b  5c  5d 7a 15a 15b

 Y.1     1  1  1  1  1   1   1  1   1   1
 Y.2     6 -3  .  A *A   C  *C -1   D  *D
 Y.3     6 -3  . *A  A  *C   C -1  *D   D
 Y.4    14  5 -1  B *B  -C -*C  .   .   .
 Y.5    14  5 -1 *B  B -*C  -C  .   .   .
```

```
 A = -2*E(5)-2*E(5)^4
   = 1-Sqrt(5) = 1-r5
 B = -3*E(5)-3*E(5)^4
   = (3-3*Sqrt(5))/2 = -3b5
 C = E(5)+2*E(5)^2+2*E(5)^3+E(5)^4
   = (-3-Sqrt(5))/2 = -2-b5
 D = E(5)+E(5)^4
   = (-1+Sqrt(5))/2 = b5
gap> List( irr, x -> SizeOfFieldOfDefinition( x, p ) );
[ 2, 4, 4, 4, 4 ]
```

We use the same approach as in the previous example.

```
––––––––––––––––––– Example –––––––––––––––––––
gap> tom:= TableOfMarks( "J2" );;
gap> info:= OneAtlasGeneratingSetInfo( "J2", Dimension, 14, Ring, GF(4) );;
gap> gens:= List( AtlasGenerators( info ).generators,
>                 x -> BlownUpMat( Basis(GF(4)), x ) );;
gap> orbits_from_tom( tom, gens, p );
rec(
  decomp := [ 235, 33, 282, 38, 0, 0, 6, 31, 36, 0, 0, 0, 3, 66, 9,
      0, 0, 0, 0, 0, 0, 2, 18, 0, 0, 1, 0, 0, 15, 0, 0, 0, 6, 0, 0,
      0, 0, 0, 0, 0, 12, 0, 0, 5, 0, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 3, 1, 3, 0, 9, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
      3, 0, 0, 0, 0, 0, 3, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 9, 0, 0, 0,
      0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0,
      0, 0, 3, 0, 0, 1 ],
  fixed := [ 268435456, 65536, 65536, 65536, 256, 1024, 4096, 1024,
      1024, 256, 256, 256, 64, 1024, 64, 256, 16, 16, 64, 64, 64,
      256, 256, 64, 16, 16, 64, 64, 64, 64, 16, 16, 1024, 4, 4, 4, 4,
      16, 16, 16, 64, 16, 16, 16, 16, 64, 16, 16, 16, 64, 16, 16, 16,
      16, 4, 16, 16, 16, 16, 1, 64, 4, 16, 4, 64, 4, 16, 4, 16, 1, 4,
      16, 4, 4, 16, 16, 4, 4, 16, 1, 4, 16, 1, 1, 1, 16, 4, 16, 16,
      16, 1, 4, 16, 4, 4, 1, 4, 64, 4, 4, 4, 16, 4, 4, 4, 1, 1, 4,
      16, 1, 4, 1, 4, 1, 4, 16, 4, 4, 1, 1, 1, 1, 4, 1, 1, 16, 1, 1,
      4, 1, 4, 4, 1, 4, 1, 1, 4, 1, 4, 1, 1, 1, 4, 1, 1, 1 ],
  nonzeropos := [ 1, 2, 3, 4, 7, 8, 9, 13, 14, 15, 22, 23, 26, 29,
      33, 41, 44, 46, 50, 61, 62, 63, 65, 72, 82, 93, 99, 105, 109,
      116, 126, 131, 139, 143, 146 ],
  staborders := [ 1, 2, 2, 3, 4, 4, 4, 6, 6, 6, 8, 8, 9, 10, 12, 12,
      14, 16, 16, 24, 24, 24, 24, 30, 40, 50, 60, 72, 96, 120, 192,
      240, 600, 1920, 604800 ] )
```

We see that *S* has 235 regular orbits on (the dual space of) *N*.

10.6 $G/N \cong {}^3D_4(2)$ and $|N| = 2^{26}$

The group $S = {}^3D_4(2)$ has exactly one irreducible 26–dimensional module over the field with two elements, up to isomorphism. This module is in fact absolutely irreducible.

```
────────────────── Example ──────────────────
gap> p:= 2;;   d:= 26;;
gap> t:= CharacterTable( "3D4(2)" ) mod p;
BrauerTable( "3D4(2)", 2 )
gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
gap> Display( t, rec( chars:= irr, powermap:= false,
>                     centralizers:= false ) );
3D4(2)mod2

        1a 3a 3b 7a 7b 7c 7d 9a 9b 9c 13a 13b 13c 21a 21b 21c

Y.1      1  1  1  1  1  1  1  1  1  1   1   1   1   1   1   1
Y.2      8  2 -1  A  C  B  1  D  F  E   G   I   H   J   L   K
Y.3      8  2 -1  B  A  C  1  E  D  F   H   G   I   K   J   L
Y.4      8  2 -1  C  B  A  1  F  E  D   I   H   G   L   K   J
Y.5     26 -1 -1  5  5  5 -2  2  2  2   .   .   .  -1  -1  -1


A = 3*E(7)^2+E(7)^3+E(7)^4+3*E(7)^5
B = 3*E(7)+E(7)^2+E(7)^5+3*E(7)^6
C = E(7)+3*E(7)^3+3*E(7)^4+E(7)^6
D = -E(9)^2+E(9)^3-2*E(9)^4-2*E(9)^5+E(9)^6-E(9)^7
E = -E(9)^2+E(9)^3+E(9)^4+E(9)^5+E(9)^6-E(9)^7
F = 2*E(9)^2+E(9)^3+E(9)^4+E(9)^5+E(9)^6+2*E(9)^7
G = E(13)+E(13)^2+E(13)^3+E(13)^5+E(13)^8+E(13)^10+E(13)^11+E(13)^12
H = E(13)+E(13)^4+E(13)^5+E(13)^6+E(13)^7+E(13)^8+E(13)^9+E(13)^12
I = E(13)^2+E(13)^3+E(13)^4+E(13)^6+E(13)^7+E(13)^9+E(13)^10+E(13)^11
J = E(7)^3+E(7)^4
K = E(7)^2+E(7)^5
L = E(7)+E(7)^6
```

We try the same approach as in the examples about the group J_2.

```
────────────────── Example ──────────────────
gap> tom:= TableOfMarks( "3D4(2)" );
TableOfMarks( "3D4(2)" )
gap> StandardGeneratorsInfo( tom );
[ rec( ATLAS := true,
      description := "|z|=8, z^4=a, |b|=9, |ab|=13, |abb|=8",
      generators := "a, b",
      script := [ [ 1, 8, 4 ], [ 2, 9 ], [ 1, 1, 2, 1, 13 ],
          [ 1, 1, 2, 1, 2, 1, 8 ] ], standardization := 1 ) ]
gap> info:= OneAtlasGeneratingSetInfo( "3D4(2)", Dimension, 26, Ring, GF(2) );;
gap> gens:= AtlasGenerators( info ).generators;;
gap> map:= GroupGeneralMappingByImages( UnderlyingGroup( tom ),
>     Group( gens ), GeneratorsOfGroup( UnderlyingGroup( tom ) ), gens );;
gap> IsGroupHomomorphism( map );
true
```

Now we apply the function orbits_from_tom.

```
────────────────── Example ──────────────────
gap> orbsinfo:= orbits_from_tom( tom, gens, p );;
gap> orbsinfo.fixed[1];
```

```
67108864
gap> orbsinfo.decomp[1];
0
```

Unfortunately, *S* has no regular orbit on (the dual of) *N*. However, there is one orbit whose point stabilizer in *S* is a dihedral group D_{18} of order 18.

```
 ———————————————————————————— Example ————————————————————————————
gap> orbsinfo.staborders;
[ 16, 16, 18, 42, 48, 52, 64, 72, 392, 1008, 1536, 3024, 3072, 3584,
  258048, 211341312 ]
gap> orbsinfo.nonzeropos[3];
446
gap> orbsinfo.decomp[446];
1
gap> u:= RepresentativeTom( tom, 446 );
<permutation group of size 18 with 2 generators>
gap> IsDihedralGroup( u );
true
```

Thus there ia a linear character λ of *N* whose inertia subgroup $T = I_G(\lambda)$ has the structure $N.D_{18}$. Now $\mathrm{Irr}(T|\lambda)$ can be identified with those irreducibles of $T/\ker(\lambda)$ that restrict nontrivially to $N/\ker(\lambda)$, and there are only two groups, up to isomorphism, that can occur as $T/\ker(\lambda)$.

```
 ———————————————————————————— Example ————————————————————————————
gap> cand:= Filtered( AllSmallGroups( 36 ),
>              x -> Size( Centre( x ) ) = 2 and
>                    IsDihedralGroup( x / Centre( x ) ) );
[ <pc group of size 36 with 4 generators>,
  <pc group of size 36 with 4 generators> ]
gap> List( cand, StructureDescription );
[ "C9 : C4", "D36" ]
```

These two groups are a split and a nonsplit extension of the cyclic group of order 18 with a group of order two that acts by inverting. In other words, these two groups are the direct product of D_{18} with a cyclic group of order two and the subdirect product of D_{18} with a cyclic group of order four.

Both groups possess irreducible characters of degree two, one rational valued and the other not, which restrict nontrivially to the centre.

```
 ———————————————————————————— Example ————————————————————————————
gap> Display( CharacterTable( "Dihedral", 18 ) );
Dihedral(18)

    2  1  .  .  .  .  1
    3  2  2  2  2  2  .

       1a 9a 9b 3a 9c 2a
    2P 1a 9b 9c 3a 9a 1a
    3P 1a 3a 3a 1a 3a 2a

X.1     1  1  1  1  1  1
X.2     1  1  1  1  1 -1
```

```
   X.3    2   A   B  -1   C   .
   X.4    2   B   C  -1   A   .
   X.5    2  -1  -1   2  -1   .
   X.6    2   C   A  -1   B   .


   A = -E(9)^2-E(9)^4-E(9)^5-E(9)^7
   B = E(9)^2+E(9)^7
   C = E(9)^4+E(9)^5
```

By \cite[Lemma 5.1 (ii)]{DNT}, we are done.

10.7 $G/N \cong {}^3D_4(2)$ **and** $|N| = 3^{25}$

The group $S = {}^3D_4(2)$ has exactly one irreducible 25-dimensional module over the field with three elements, up to isomorphism. This module is in fact absolutely irreducible.

```
────────────────────────── Example ──────────────────────────
 gap> p:= 3;;   d:= 25;;
 gap> t:= CharacterTable( "3D4(2)" ) mod p;
 BrauerTable( "3D4(2)", 3 )
 gap> irr:= Filtered( Irr( t ), x -> x[1] <= d );;
 gap> Display( t, rec( chars:= irr, powermap:= false,
 >                      centralizers:= false ) );
 3D4(2)mod3

         1a 2a 2b 4a 4b 4c 7a 7b 7c 7d 8a 8b 13a 13b 13c 14a 14b 14c 28a

 Y.1      1  1  1  1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1
 Y.2     25 -7  1  5 -3  1  4  4  4 -3 -1 -1  -1  -1  -1   .   .   .  -2

         28b 28c

 Y.1       1   1
 Y.2      -2  -2
```

We use the same approach as in the examples about the group J_2.

```
────────────────────────── Example ──────────────────────────
 gap> tom:= TableOfMarks( "3D4(2)" );;
 gap> info:= OneAtlasGeneratingSetInfo( "3D4(2)", Dimension, d, Ring, GF(p) );;
 gap> gens:= AtlasGenerators( info ).generators;;
 gap> orbsinfo:= orbits_from_tom( tom, gens, p );;
 gap> orbsinfo.fixed[1];
 847288609443
 gap> orbsinfo.decomp[1];
 3551
```

We see that S has 3551 regular orbits on (the dual space of) N.

Chapter 11

GAP Computations Concerning Probabilistic Generation of Finite Simple Groups

Date: March 28th, 2012

This is a collection of examples showing how the GAP system [GAP24] can be used to compute information about the probabilistic generation of finite almost simple groups. It includes all examples that were needed for the computational results in [BGK08].

The purpose of this writeup is twofold. On the one hand, the computations are documented this way. On the other hand, the GAP code shown for the examples can be used as test input for automatic checking of the data and the functions used.

A first version of this document, which was based on GAP 4.4.10, had been accessible in the web since April 2006 and is available in the arXiv (no. 0710.3267) since October 2007. The differences between that document and the current version are as follows.

- The format of the GAP output was adjusted to the changed behaviour of GAP until version 4.10. This affects mainly the way how GAP records are printed.

- Several computations are now easier because more character tables of almost simple groups and maximal subgroups of such groups are available in the GAP Character Table Library. (The more involved computations from the original version have been kept in the file.)

- The computation of all conjugacy classes of a subgroup of $P\Omega^+(12,3)$ has been replaced by the computation of the conjugacy classes of elements of prime order in this subgroup.

- The irreducible element chosen in the simple group $P\Omega^-(10,3)$ has order 61 not 122.

11.1 Overview

The main purpose of this note is to document the GAP computations that were carried out in order to obtain the computational results in [BGK08]. Table I lists the simple groups among these examples. The first column gives the group names, the second and third columns contain a plus sign $+$ or a minus sign $-$, depending on whether the quantities $\sigma(G,s)$ and $P(G,s)$, respectively, are less than $1/3$. The fourth column lists the orders of elements s which either prove the $+$ signs or cover most of the cases

for proving these signs. The fifth column lists the sections in this note where the example is treated. The rows of the table are ordered alphabetically w.r.t. the group names.

In order to keep this note self-contained, we first describe the theory needed, in Section 11.2. The translation of the relevant formulae into GAP functions can be found in Section 11.3. Then Section 11.4 describes the computations that only require (ordinary) character tables in the GAP Character Table Library [Bre25]. Computations using also the groups are shown in Section 11.5. In each of the last two sections, the examples are ordered alphabetically w.r.t. the names of the simple groups.

G	$\sigma < \frac{1}{3}$	$P < \frac{1}{3}$	$|s|$	see
A_5	$-$	$-$	5	11.5.2
A_6	$-$	$-$	4	11.5.3
A_7	$-$	$-$	7	11.5.4
A_8	$+$		15	11.4.4, 11.5.5
A_9	$+$		9	11.4.4, 11.5.1
A_{11}	$+$		11	11.4.4, 11.5.1
A_{13}	$+$		13	11.4.4, 11.5.1
A_{15}	$+$		15	11.5.1
A_{17}	$+$		17	11.5.1
A_{19}	$+$		19	11.5.1
A_{21}	$+$		21	11.5.1
A_{23}	$+$		23	11.5.1
$L_3(2)$	$+$		7	11.4.4, 11.4.5, 11.5.5, 11.5.8
$L_3(3)$	$+$		13	11.4.4, 11.4.5, 11.5.5
$L_3(4)$	$+$		7	11.4.4, 11.4.5
$L_4(3)$	$+$		20	11.4.4, 11.5.5
$L_4(4)$	$+$		85	11.5.5
$L_6(2)$	$+$		63	11.5.5
$L_6(3)$	$+$		182	11.5.5
$L_6(4)$	$+$		455	11.5.5
$L_6(5)$	$+$		1953	11.5.5
$L_8(2)$	$+$		255	11.5.5
$L_{10}(2)$	$+$		1023	11.5.5
M_{11}	$-$	$-$	11	11.5.9
M_{12}	$-$	$+$	10	11.4.3, 11.5.10
$O_8^+(2)$	$-$	$-$	15	11.5.12
$O_8^+(3)$	$-$	$-$	20	11.5.13
$O_8^+(4)$	$+$		65	11.5.14
$O_{10}^+(2)$	$+$		45	11.4.7
$O_{12}^+(2)$	$+$		85	11.4.9
$O_{12}^+(3)$	$+$		205	11.5.18
$O_8^-(2)$	$+$		17	11.4.4
$O_8^-(3)$	$+$		41	11.4.6
$O_{10}^-(2)$	$+$		33	11.4.8
$O_{10}^-(3)$	$+$		122	11.5.16
$O_{12}^-(2)$	$+$		65	11.4.10
$O_{14}^-(2)$	$+$		129	11.5.17
$O_7(3)$	$-$	$-$	14	11.5.11
$S_4(4)$	$+$		17	11.4.4, 11.4.5
$S_6(2)$	$-$	$-$	9	11.5.20
$S_6(3)$	$+$		14	11.4.4, 11.4.5
$S_6(4)$	$+$		65	11.4.11
$S_8(2)$	$-$	$-$	17	11.5.21
$S_8(3)$	$+$		41	11.4.13
$U_3(3)$	$+$		6	11.4.4, 11.4.5
$U_3(5)$	$+$		10	11.4.4, 11.4.5
$U_4(2)$	$-$	$-$	9	11.5.23
$U_4(3)$	$-$	$+$	7	11.5.24
$U_4(4)$	$+$		65	11.4.14
$U_5(2)$	$+$		11	11.4.4
$U_6(2)$	$+$		11	11.4.15

Table: Table I: Computations needed in [BGK08]

Contrary to [BGK08], Atlas notation is used throughout this note, because the identifiers used for character tables in the GAP Character Table Library follow mainly the Atlas [CCN$^+$85]. For example, we write $L_d(q)$ for $\mathrm{PSL}(d,q)$, $S_d(q)$ for $\mathrm{PSp}(d,q)$, $U_d(q)$ for $\mathrm{PSU}(d,q)$, and $O_{2d}^+(q)$, $O_{2d}^-(q)$, $O_{2d+1}(q)$ for $\mathrm{P\Omega}^+(2d,q)$, $\mathrm{P\Omega}^-(2d,q)$, $\mathrm{P\Omega}(2d+1,q)$, respectively.

Furthermore, in the case of classical groups, the character tables of the (almost) *simple* groups are considered not the tables of the matrix groups (which are in fact often not available in the GAP Character Table Library). Consequently, also element orders and the description of maximal subgroups refer to the (almost) simple groups not to the matrix groups.

This note contains also several examples that are not needed for the proofs in [BGK08]. Besides several small simple groups G whose character table is contained in the GAP Character Table Library and for which enough information is available for computing $\sigma(G)$, in Section 11.4.4, a few such examples appear in individual sections. In the table of contents, the section headers of the latter kind of examples are marked with an asterisk ($*$).

The examples use the GAP Character Table Library, the GAP Library of Tables of Marks, and the GAP interface [WPN$^+$22] to the Atlas of Group Representations [WWT$^+$], so we first load these three packages in the required versions. The GAP output was adjusted to the versions shown below; in older versions, features necessary for the computations may be missing, and it may happen that with newer versions, the behaviour is different.

```
———————————————————— Example ————————————————————
gap> CompareVersionNumbers( GAPInfo.Version, "4.5.0" );
true
gap> LoadPackage( "ctbllib", "1.2", false );
true
gap> LoadPackage( "tomlib", "1.2", false );
true
gap> LoadPackage( "atlasrep", "1.5", false );
true
```

Some of the computations in Section 11.5 require about 800 MB of space (on 32 bit machines). Therefore we check whether GAP was started with sufficient maximal memory; the command line option for this is -o 800m.

```
———————————————————— Example ————————————————————
gap> max:= GAPInfo.CommandLineOptions.o;;
gap> if not ( ( IsSubset( max, "m" ) and
>                Int( Filtered( max, IsDigitChar ) ) ) >= 800 ) or
>            ( IsSubset( max, "g" ) and
>                Int( Filtered( max, IsDigitChar ) ) ) >= 1 ) ) then
>      Print( "the maximal allowed memory might be too small\n" );
>    fi;
```

Several computations involve calls to the GAP function `Random` (**Reference: Random**). In order to make the results of individual examples reproducible, independent of the rest of the computations, we reset the relevant random number generators whenever this is appropriate. For that, we store the initial states in the variable `staterandom`, and provide a function for resetting the random number generators. (The `Random` (**Reference: Random**) calls in the GAP library use the two random number generators `GlobalRandomSource` (**Reference: GlobalRandomSource**) and `GlobalMersenneTwister` (**Reference: GlobalMersenneTwister**).)

```
                            ─── Example ───
  gap> staterandom:= [ State( GlobalRandomSource ),
  >                    State( GlobalMersenneTwister ) ];;
  gap> ResetGlobalRandomNumberGenerators:= function()
  >       Reset( GlobalRandomSource, staterandom[1] );
  >       Reset( GlobalMersenneTwister, staterandom[2] );
  > end;;
```

11.2 Prerequisites

11.2.1 Theoretical Background

Let G be a finite group, S the socle of G, and denote by $G^\times$ the set of nonidentity elements in G. For $s, g \in G^\times$, let $P(g,s) := |\{h \in G; S \not\subseteq \langle s^h, g \rangle\}|/|G|$, the proportion of elements in the class s^G which fail to generate at least S with g; we set $P(G,s) := \max\{P(g,s); g \in G^\times\}$. We are interested in finding a class s^G of elements in S such that $P(G,s) < 1/3$ holds.

First consider $g \in S$, and let $\mathbb{M}(S,s)$ denote the set of those maximal subgroups of S that contain s. We have

$$|\{h \in S; S \not\subseteq \langle s^h, g \rangle\}| = |\{h \in S; \langle s, hgh^{-1} \rangle \neq S\}| \leq \sum_{M \in \mathbb{M}(S,s)} |\{h \in S; hgh^{-1} \in M\}|$$

Since $hgh^{-1} \in M$ holds if and only if the coset Mh is fixed by g under the permutation action of S on the right cosets of M in S, we get that $|\{h \in S; hgh^{-1} \in M\}| = |C_S(g)| \cdot |g^S \cap M| = |M| \cdot 1_M^S(g)$, where 1_M^S is the permutation character of this action, of degree $|S|/|M|$. Thus

$$|\{h \in S; \langle s, hgh^{-1} \rangle \neq S\}|/|S| \leq \sum_{M \in \mathbb{M}(S,s)} 1_M^S(g)/1_M^S(1).$$

We abbreviate the right hand side of this inequality by $\sigma(g,s)$, set $\sigma(S,s) := \max\{\sigma(g,s); g \in S^\times\}$, and choose a transversal T of S in G. Then $P(g,s) \leq |T|^{-1} \cdot \sum_{t \in T} \sigma(g^t, s)$ and thus $P(G,s) \leq \sigma(S,s)$ holds.

If $S = G$ and if $\mathbb{M}(G,s)$ consists of a single maximal subgroup M of G then equality holds, i.e., $P(g,s) = \sigma(g,s) = 1_M^S(g)/1_M^S(1)$.

The quantity $1_M^S(g)/1_M^S(1) = |g^S \cap M|/|g^S|$ is the proportion of fixed points of g in the permutation action of S on the right cosets of its subgroup M. This is called the *fixed point ratio* of g w. r. t. S/M, and is denoted as $\mu(g, S/M)$.

For a subgroup M of S, the number n of S-conjugates of M containing s is equal to $|M^S| \cdot |s^S \cap M|/|s^S|$. To see this, consider the set $\{(s^h, M^k); h, k \in S, s^h \in M^k\}$, the cardinality of which can be counted either as $|M^S| \cdot |s^S \cap M|$ or as $|s^S| \cdot n$. So we get $n = |M| \cdot 1_M^S(s)/|N_S(M)|$.

If S is a finite *nonabelian simple* group then each maximal subgroup in S is self-normalizing, and we have $n = 1_M^S(s)$ if M is maximal. So we can replace the summation over $\mathbb{M}(S,s)$ by one over a set $\tilde{\mathbb{M}}(S,s)$ of representatives of conjugacy classes of maximal subgroups of S, and get that

$$\sigma(g,s) = \sum_{M \in \tilde{\mathbb{M}}(S,s)} \frac{1_M^S(s) \cdot 1_M^S(g)}{1_M^S(1)}.$$

Furthermore, we have $|\mathbb{M}(S,s)| = \sum_{M \in \tilde{\mathbb{M}}(S,s)} 1_M^S(s)$.

In the following, we will often deal with the quantities $\sigma(S) := \min\{\sigma(S,s); s \in S^\times\}$ and $\mathscr{S}(S) := \lceil 1/\sigma(S) - 1 \rceil$. These values can be computed easily from the primitive permutation characters of S.

Analogously, we set $P(S) := \min\{P(S,s); s \in S^\times\}$ and $\mathscr{P}(S) := \lceil 1/P(S) - 1 \rceil$. Clearly we have $P(S) \le \sigma(S)$ and $\mathscr{P}(S) \ge \mathscr{S}(S)$.

One interpretation of $\mathscr{P}(S)$ is that if this value is at least k then it follows that for any $g_1, g_2, \ldots, g_k \in S^\times$, there is some $s \in S$ such that $S = \langle g_i, s \rangle$, for $1 \le i \le k$. In this case, S is said to have *spread* at least k. (Note that the lower bound $\mathscr{S}(S)$ for $\mathscr{P}(S)$ can be computed from the list of primitive permutation characters of S.)

Moreover, $\mathscr{P}(S) \ge k$ implies that the element s can be chosen uniformly from a fixed conjugacy class of S. This is called *uniform spread* at least k in [BGK08].

It is proved in [GK00] that all finite simple groups have uniform spread at least 1, that is, for any element $x \in S^\times$, there is an element y in a prescribed class of S such that $G = \langle x, y \rangle$ holds. In [BGK08, Corollary 1.3], it is shown that all finite simple groups have uniform spread at least 2, and the finite simple groups with (uniform) spread exactly 2 are listed.

Concerning the spread, it should be mentioned that the methods used here and in [BGK08] are nonconstructive in the sense that they cannot be used for finding an element s that generates G together with each of the k prescribed elements $g_1, g_2, \ldots, g_k$.

Now consider $g \in G \setminus S$. Since $P(g^k, s) \ge P(g, s)$ for any positive integer k, we can assume that g has prime order p, say. We set $H = \langle S, g \rangle \le G$, with $[H : S] = p$, choose a transversal T of H in G, let $\mathbb{M}'(H, s) := \mathbb{M}(H, s) \setminus \{S\}$, and let $\tilde{\mathbb{M}}'(H, s)$ denote a set of representatives of H-conjugacy classes of these groups. As above,

$$
\begin{aligned}
|\{h \in H; S \not\subseteq \langle s^h, g \rangle\}|/|H| &= |\{h \in H; \langle s^h, g \rangle \ne H\}|/|H| \\
&\le \sum_{M \in \mathbb{M}'(H,s)} |\{h \in H; hgh^{-1} \in M\}|/|H| \\
&= \sum_{M \in \mathbb{M}'(H,s)} 1_M^H(g)/1_M^H(1) \\
&= \sum_{M \in \tilde{\mathbb{M}}'(H,s)} 1_M^H(g) \cdot 1_M^H(s)/1_M^H(1)
\end{aligned}
$$

(Note that no summand for $M = S$ occurs, so each group in $\tilde{\mathbb{M}}'(H, s)$ is self-normalizing.) We abbreviate the right hand side by $\sigma(H, g, s)$, and set $\sigma'(H, s) = \max\{\sigma(H, g, s); g \in H \setminus S, |g| = [H : S]\}$. Then we get $P(g, s) \le |T|^{-1} \cdot \sum_{t \in T} \sigma(H^t, g^t, s)$ and thus

$$
P(G, s) \le \max\{P(S, s), \max\{\sigma'(H, s); S \le H \le G, [H : S] \text{ prime}\}\}.
$$

For convenience, we set $P'(G, s) = \max\{P(g, s); g \in G \setminus S\}$.

11.2.2 Computational Criteria

The following criteria will be used when we have to show the existence or nonexistence of $x_1, x_2, \ldots, x_k$, and $s \in G$ with the property $\langle x_i, s \rangle = G$ for $1 \le i \le k$. Note that manipulating lists of integers (representing fixed or moved points) is much more efficient than testing whether certain permutations generate a given group.

Lemma:

Let G be a finite group, $s \in G^\times$, and $X = \bigcup_{M \in \mathbb{M}(G,s)} G/M$. For $x_1, x_2, \ldots, x_k \in G$, the conjugate s' of s satisfies $\langle x_i, s' \rangle = G$ for $1 \le i \le k$ if and only if $\mathrm{Fix}_X(s') \cap \bigcup_{i=1}^k \mathrm{Fix}_X(x_i) = \emptyset$ holds.

Proof. If $s^g \in U \le G$ for some $g \in G$ then $\mathrm{Fix}_X(U) = \emptyset$ if and only if $U = G$ holds; note that $\mathrm{Fix}_X(G) = \emptyset$, and $\mathrm{Fix}_X(U) = \emptyset$ implies that $U \not\subseteq h^{-1}Mh$ holds for all $h \in G$ and $M \in \mathbb{M}(G, s)$, thus $U = G$. Applied to $U = \langle x_i, s' \rangle$, we get $\langle x_i, s' \rangle = G$ if and only if $\mathrm{Fix}_X(s') \cap \mathrm{Fix}_X(x_i) = \mathrm{Fix}_X(U) = \emptyset$.

Corollary 1:

If $\mathbb{M}(G,s) = \{M\}$ in the situation of the above Lemma then there is a conjugate s' of s that satisfies $\langle x_i, s' \rangle = G$ for $1 \le i \le k$ if and only if $\bigcup_{i=1}^{k} \mathrm{Fix}_X(x_i) \ne X$.

Corollary 2:

Let G be a finite simple group and let X be a G-set such that each $g \in G$ fixes at least one point in X but that $\mathrm{Fix}_X(G) = \emptyset$ holds. If $x_1, x_2, \ldots x_k$ are elements in G such that $\bigcup_{i=1}^{k} \mathrm{Fix}_X(x_i) = X$ holds then for each $s \in G$ there is at least one i with $\langle x_i, s \rangle \ne G$.

11.3 GAP Functions for the Computations

After the introduction of general utilities in Section 11.3.1, we distinguish two different tasks. Section 11.3.2 introduces functions that will be used in the following to compute $\sigma(g,s)$ with character-theoretic methods. Functions for computing $P(g,s)$ or an upper bound for this value will be introduced in Section 11.3.3.

The GAP functions shown in this section are collected in the file `tst/probgen.g` that is distributed with the GAP Character Table Library, see http://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib.

The functions have been designed for the examples in the later sections, they could be generalized and optimized for other examples. It is not our aim to provide a package for this functionality.

11.3.1 General Utilities

Let `list` be a dense list and `prop` be a unary function that returns `true` or `false` when applied to the entries of `list`. `PositionsProperty` returns the set of positions in `list` for which `true` is returned.

```
————— Example —————
gap> if not IsBound( PositionsProperty ) then
>      PositionsProperty:= function( list, prop )
>        return Filtered( [ 1 .. Length( list ) ], i -> prop( list[i] ) );
>      end;
>    fi;
```

The following two functions implement loops over ordered triples (and quadruples, respectively) in a Cartesian product. A prescribed function `prop` is subsequently applied to the triples (quadruples), and if the result of this call is `true` then this triple (quadruple) is returned immediately; if none of the calls to `prop` yields `true` then `fail` is returned.

```
————— Example —————
gap> BindGlobal( "TripleWithProperty", function( threelists, prop )
>      local i, j, k, test;
>
>      for i in threelists[1] do
>        for j in threelists[2] do
>          for k in threelists[3] do
>            test:= [ i, j, k ];
>            if prop( test ) then
>                return test;
>            fi;
>          od;
>        od;
>      od;
```

```
>
>      return fail;
> end );
gap> BindGlobal( "QuadrupleWithProperty", function( fourlists, prop )
>      local i, j, k, l, test;
>
>      for i in fourlists[1] do
>        for j in fourlists[2] do
>          for k in fourlists[3] do
>            for l in fourlists[4] do
>              test:= [ i, j, k, l ];
>              if prop( test ) then
>                return test;
>              fi;
>            od;
>          od;
>        od;
>      od;
>
>      return fail;
> end );
```

Of course one could do better by considering *un*ordered *n*-tuples when several of the argument lists are equal, and in practice, backtrack searches would often allow one to prune parts of the search tree in early stages. However, the above loops are not time critical in the examples presented here, so the possible improvements are not worth the effort for our purposes.

The function `PrintFormattedArray` prints the matrix `array` in a columnwise formatted way. (The only diference to the GAP library function `PrintArray` (**Reference: PrintArray**) is that `PrintFormattedArray` chooses each column width according to the entries only in this column not w.r.t. the whole matrix.)

```
──────── Example ────────
gap> BindGlobal( "PrintFormattedArray", function( array )
>      local colwidths, n, row;
>      array:= List( array, row -> List( row, String ) );
>      colwidths:= List( TransposedMat( array ),
>                       col -> Maximum( List( col, Length ) ) );
>      n:= Length( array[1] );
>      for row in List( array, row -> List( [ 1 .. n ],
>                  i -> String( row[i], colwidths[i] ) ) ) do
>        Print( "  ", JoinStringsWithSeparator( row, " " ), "\n" );
>      od;
> end );
```

Finally, `CleanWorkspace` is a utility for reducing the space needed. This is achieved by unbinding those user variables that are not write protected and are not mentioned in the list `NeededVariables` of variable names that are bound now, and by flushing the caches of tables of marks and character tables.

```
──────── Example ────────
gap> BindGlobal( "NeededVariables", NamesUserGVars() );
gap> BindGlobal( "CleanWorkspace", function()
```

```
>        local name, record;
>
>        for name in Difference( NamesUserGVars(), NeededVariables ) do
>         if not IsReadOnlyGlobal( name ) then
>           UnbindGlobal( name );
>         fi;
>        od;
>        for record in [ LIBTOMKNOWN, LIBTABLE ] do
>          for name in RecNames( record.LOADSTATUS ) do
>            Unbind( record.LOADSTATUS.( name ) );
>            Unbind( record.( name ) );
>          od;
>        od;
> end );
```

The function `PossiblePermutationCharacters` takes two ordinary character tables `sub` and `tbl`, computes the possible class fusions from `sub` to `tbl`, then induces the trivial character of `sub` to `tbl`, w.r.t. these fusions, and returns the set of these class functions. (So if `sub` and `tbl` are the character tables of groups H and G, respectively, where H is a subgroup of G, then the result contains the permutation character 1_H^G.)

Note that the columns of the character tables in the GAP Character Table Library are not explicitly associated with particular conjugacy classes of the corresponding groups, so from the character tables, we can compute only *possible* class fusions, i.e., maps between the columns of two tables that satisfy certain necessary conditions, see the section about the function `PossibleClassFusions` in the GAP Reference Manual for details. There is no problem if the permutation character is uniquely determined by the character tables, in all other cases we give ad hoc arguments for resolving the ambiguities.

```
————————————————————— Example —————————————————————
 gap> if not IsBound( PossiblePermutationCharacters ) then
 >        BindGlobal( "PossiblePermutationCharacters", function( sub, tbl )
 >         local fus, triv;
 >
 >         fus:= PossibleClassFusions( sub, tbl );
 >         if fus = fail then
 >           return fail;
 >         fi;
 >         triv:= [ TrivialCharacter( sub ) ];
 >
 >         return Set(
 >             List( fus, map -> Induced( sub, tbl, triv, map )[1] ) );
 >        end );
 >    fi;
```

11.3.2 Character-Theoretic Computations

We want to use the GAP libraries of character tables and of tables of marks, and proceed in three steps.

First we extract the primitive permutation characters from the library information if this is available; for that, we write the function `PrimitivePermutationCharacters`. Then the result can be used as the input for the function `ApproxP`, which computes the values $\sigma(g,s)$. Finally, the functions `ProbGenInfoSimple` and `ProbGenInfoAlmostSimple` compute $\mathscr{S}(G)$.

For a group G whose character table T is contained in the GAP character table library, the complete set of primitive permutation characters can be easily computed if the character tables of all maximal subgroups and their class fusions into T are known (in this case, we check whether the attribute `Maxes` (**CTblLib: Maxes**) of T is bound) or if the table of marks of G and the class fusion from T into this table of marks are known (in this case, we check whether the attribute `FusionToTom` (**CTblLib: FusionToTom**) of T is bound). If the attribute `UnderlyingGroup` (**Reference: UnderlyingGroup for tables of marks**) of T is bound then this group can be used to compute the primitive permutation characters. The latter happens if T was computed from the group object in GAP; for tables in the GAP character table library, this is not the case by default.

The GAP function `PrimitivePermutationCharacters` tries to compute the primitive permutation characters of a group using this information; it returns the required list of characters if this can be computed this way, otherwise `fail` is returned. (For convenience, we use the GAP mechanism of *attributes* in order to store the permutation characters in the character table object once they have been computed.)

```
─────────────────────────── Example ───────────────────────────
gap> DeclareAttribute( "PrimitivePermutationCharacters",
>                      IsCharacterTable );
gap> InstallOtherMethod( PrimitivePermutationCharacters,
>      [ IsCharacterTable ],
>      function( tbl )
>      local maxes, i, fus, poss, tom, G;
>
>      if HasMaxes( tbl ) then
>        maxes:= List( Maxes( tbl ), CharacterTable );
>        for i in [ 1 .. Length( maxes ) ] do
>          fus:= GetFusionMap( maxes[i], tbl );
>          if fus = fail then
>            fus:= PossibleClassFusions( maxes[i], tbl );
>            poss:= Set( fus,
>              map -> InducedClassFunctionsByFusionMap(
>                       maxes[i], tbl,
>                       [ TrivialCharacter( maxes[i] ) ], map )[1] );
>            if Length( poss ) = 1 then
>              maxes[i]:= poss[1];
>            else
>              return fail;
>            fi;
>          else
>            maxes[i]:= TrivialCharacter( maxes[i] )^tbl;
>          fi;
>        od;
>        return maxes;
>      elif HasFusionToTom( tbl ) then
>        tom:= TableOfMarks( tbl );
>        maxes:= MaximalSubgroupsTom( tom );
>        return PermCharsTom( tbl, tom ){ maxes[1] };
>      elif HasUnderlyingGroup( tbl ) then
>        G:= UnderlyingGroup( tbl );
>        return List( MaximalSubgroupClassReps( G ),
>                     M -> TrivialCharacter( M )^tbl );
>      fi;
```

```
>
>     return fail;
> end );
```

The function `ApproxP` takes a list `primitives` of primitive permutation characters of a group G, say, and the position `spos` of the class s^G in the character table of G.

Assume that the elements in `primitives` have the form 1_M^G, for suitable maximal subgroups M of G, and let $\tilde{\mathbb{M}}$ be the set of these groups M. `ApproxP` returns the class function ψ of G that is defined by $\psi(1) = 0$ and

$$\psi(g) = \sum_{M \in \tilde{\mathbb{M}}} \frac{1_M^G(s) \cdot 1_M^G(g)}{1_M^G(1)}$$

otherwise.

If `primitives` contains all those primitive permutation characters 1_M^G of G (with multiplicity according to the number of conjugacy classes of these maximal subgroups) that do not vanish at s, and if all these M are self-normalizing in G –this holds for example if G is a finite simple group– then $\psi(g) = \sigma(g, s)$ holds.

```
─────────── Example ───────────
gap> BindGlobal( "ApproxP", function( primitives, spos )
>     local sum;
>
>     sum:= ShallowCopy( Sum( List( primitives,
>                          pi -> pi[ spos ] * pi / pi[1] ) ) );
>     sum[1]:= 0;
>
>     return sum;
> end );
```

Note that for computations with permutation characters, it would make the functions more complicated (and not more efficient) if we would consider only elements g of prime order, and only one representative of Galois conjugate classes.

The next functions needed in this context compute $\sigma(S)$ and $\mathscr{S}(S)$, for a simple group S, and $\sigma'(G, s)$ for an almost simple group G with socle S, respectively.

`ProbGenInfoSimple` takes the character table `tbl` of S as its argument. If the full list of primitive permutation characters of S cannot be computed with `PrimitivePermutationCharacters` then the function returns `fail`. Otherwise `ProbGenInfoSimple` returns a list containing the identifier of the table, the value $\sigma(S)$, the integer $\mathscr{S}(S)$, a list of **Atlas** names of representatives of Galois families of those classes of elements s for which $\sigma(S) = \sigma(S, s)$ holds, and the list of the corresponding cardinalities $|\mathbb{M}(S, s)|$.

```
─────────── Example ───────────
gap> BindGlobal( "ProbGenInfoSimple", function( tbl )
>     local prim, max, min, bound, s;
>     prim:= PrimitivePermutationCharacters( tbl );
>     if prim = fail then
>       return fail;
>     fi;
>     max:= List( [ 1 .. NrConjugacyClasses( tbl ) ],
>                 i -> Maximum( ApproxP( prim, i ) ) );
>     min:= Minimum( max );
```

```
>       bound:= Inverse( min );
>       if IsInt( bound ) then
>         bound:= bound - 1;
>       else
>         bound:= Int( bound );
>       fi;
>       s:= PositionsProperty( max, x -> x = min );
>       s:= List( Set( s, i -> ClassOrbit( tbl, i ) ), i -> i[1] );
>       return [ Identifier( tbl ),
>                min,
>                bound,
>                AtlasClassNames( tbl ){ s },
>                Sum( List( prim, pi -> pi{ s } ) ) ];
> end );
```

ProbGenInfoAlmostSimple takes the character tables tblS and tblG of *S* and *G*, and a list sposS of class positions (w.r.t. tblS) as its arguments. It is assumed that *S* is simple and has prime index in *G*. If PrimitivePermutationCharacters can compute the full list of primitive permutation characters of *G* then the function returns a list containing the identifier of tblG, the maximum *m* of $\sigma'(G,s)$, for *s* in the classes described by sposS, a list of Atlas names (in *G*) of the classes of elements *s* for which this maximum is attained, and the list of the corresponding cardinalities $|\mathbb{M}'(G,s)|$. When PrimitivePermutationCharacters returns fail, also ProbGenInfoAlmostSimple returns fail.

──────── Example ────────
```
gap> BindGlobal( "ProbGenInfoAlmostSimple", function( tblS, tblG, sposS )
>       local p, fus, inv, prim, sposG, outer, approx, l, max, min,
>             s, cards, i, names;
>
>       p:= Size( tblG ) / Size( tblS );
>       if not IsPrimeInt( p )
>          or Length( ClassPositionsOfNormalSubgroups( tblG ) ) <> 3 then
>         return fail;
>       fi;
>       fus:= GetFusionMap( tblS, tblG );
>       if fus = fail then
>         return fail;
>       fi;
>       inv:= InverseMap( fus );
>       prim:= PrimitivePermutationCharacters( tblG );
>       if prim = fail then
>         return fail;
>       fi;
>       sposG:= Set( fus{ sposS } );
>       outer:= Difference( PositionsProperty(
>               OrdersClassRepresentatives( tblG ), IsPrimeInt ), fus );
>       approx:= List( sposG, i -> ApproxP( prim, i ){ outer } );
>       if IsEmpty( outer ) then
>         max:= List( approx, x -> 0 );
>       else
>         max:= List( approx, Maximum );
>       fi;
>       min:= Minimum( max);
```

```
    >       s:= sposG{ PositionsProperty( max, x -> x = min ) };
    >       cards:= List( prim, pi -> pi{ s } );
    >       for i in [ 1 .. Length( prim ) ] do
    >         # Omit the character that is induced from the simple group.
    >         if ForAll( prim[i], x -> x = 0 or x = prim[i][1] ) then
    >           cards[i]:= 0;
    >         fi;
    >       od;
    >       names:= AtlasClassNames( tblG ){ s };
    >       Perform( names, ConvertToStringRep );
    >
    >       return [ Identifier( tblG ),
    >                min,
    >                names,
    >                Sum( cards ) ];
    > end );
```

The next function computes $\sigma(G, s)$ from the character table `tbl` of a simple or almost simple group G, the name `sname` of the class of s in this table, the list `maxes` of the character tables of all subgroups M with $M \in \mathbb{M}(G, s)$, and the list `numpermchars` of the numbers of possible permutation characters induced from `maxes`. If the string `"outer"` is given as an optional argument then G is assumed to be an automorphic extension of a simple group S, with $[G : S]$ a prime, and $\sigma'(G, s)$ is returned. In both situations, the result is `fail` if the numbers of possible permutation characters induced from `maxes` do not coincide with the numbers prescribed in `numpermchars`.

```
————————————————————— Example —————————————————————
  gap> BindGlobal( "SigmaFromMaxes", function( arg )
    >     local t, sname, maxes, numpermchars, prim, spos, outer;
    >
    >     t:= arg[1];
    >     sname:= arg[2];
    >     maxes:= arg[3];
    >     numpermchars:= arg[4];
    >     prim:= List( maxes, s -> PossiblePermutationCharacters( s, t ) );
    >     spos:= Position( AtlasClassNames( t ), sname );
    >     if ForAny( [ 1 .. Length( maxes ) ],
    >                i -> Length( prim[i] ) <> numpermchars[i] ) then
    >       return fail;
    >     elif Length( arg ) = 5 and arg[5] = "outer" then
    >       outer:= Difference(
    >           PositionsProperty( OrdersClassRepresentatives( t ), IsPrimeInt ),
    >           ClassPositionsOfDerivedSubgroup( t ) );
    >       return Maximum( ApproxP( Concatenation( prim ), spos ){ outer } );
    >     else
    >       return Maximum( ApproxP( Concatenation( prim ), spos ) );
    >     fi;
    > end );
```

The following function allows us to extract information about $\mathbb{M}(G, s)$ from the character table `tbl` of G and a list `snames` of class positions of s. If `Maxes( tbl )` is stored then the names of the character tables of the subgroups in $\mathbb{M}(G, s)$ and the number of conjugates are printed, otherwise `fail` is printed.

```
                        _____ Example _____
  gap> BindGlobal( "DisplayProbGenMaxesInfo", function( tbl, snames )
  >       local mx, prim, i, spos, nonz, indent, j;
  >
  >       if not HasMaxes( tbl ) then
  >         Print( Identifier( tbl ), ": fail\n" );
  >         return;
  >       fi;
  >
  >       # Now we are sure that the order of the characters returned by
  >       # 'PrimitivePermutationCharacters' is compatible with 'Maxes( tbl )'.
  >       mx:= List( Maxes( tbl ), CharacterTable );
  >       prim:= List( PrimitivePermutationCharacters( tbl ), ShallowCopy );
  >       for i in [ 1 .. Length( prim ) ] do
  >         # Deal with the case that the subgroup is normal.
  >         if ForAll( prim[i], x -> x = 0 or x = prim[i][1] ) then
  >           prim[i]:= prim[i] / prim[i][1];
  >         fi;
  >       od;
  >
  >       spos:= List( snames,
  >                    nam -> Position( AtlasClassNames( tbl ), nam ) );
  >       nonz:= List( spos, x -> PositionsProperty( prim, pi -> pi[x] <> 0 ) );
  >       for i in [ 1 .. Length( spos ) ] do
  >         Print( Identifier( tbl ), ", ", snames[i], ": " );
  >         indent:= ListWithIdenticalEntries(
  >             Length( Identifier( tbl ) ) + Length( snames[i] ) + 4, ' ' );
  >         if not IsEmpty( nonz[i] ) then
  >           Print( Identifier( mx[ nonz[i][1] ] ), "  (",
  >                  prim[ nonz[i][1] ][ spos[i] ], ")\n" );
  >           for j in [ 2 .. Length( nonz[i] ) ] do
  >             Print( indent, Identifier( mx[ nonz[i][j] ] ), "  (",
  >                    prim[ nonz[i][j] ][ spos[i] ], ")\n" );
  >           od;
  >         else
  >           Print( "\n" );
  >         fi;
  >       od;
  > end );
```

11.3.3 Computations with Groups

Here, the task is to compute $P(g,s)$ or $P(G,s)$ using explicit computations with G, where the character-theoretic bounds are not sufficient.

We start with small utilities that make the examples shorter.

For a finite solvable group G, the function PcConjugacyClassReps returns a list of representatives of the conjugacy classes of G, which are computed using a polycyclic presentation for G.

```
                        _____ Example _____
  gap> BindGlobal( "PcConjugacyClassReps", function( G )
  >       local iso;
  >
```

```
>         iso:= IsomorphismPcGroup( G );
>         return List( ConjugacyClasses( Image( iso ) ),
>                   c -> PreImagesRepresentative( iso, Representative( c ) ) );
> end );
```

For a finite group G, a list primes of prime integers, and a normal subgroup N of G, the function ClassesOfPrimeOrder returns a list of those conjugacy classes of G that are not contained in N and whose elements' orders occur in primes.

For each prime p in primes, first class representatives of order p in a Sylow p subgroup of G are computed, then the representatives in N are discarded, and then representatives w. r. t. conjugacy in G are computed.

(Note that this approach may be inappropriate for example if a large elementary abelian Sylow p subgroup occurs, and if the conjugacy tests in G are expensive, see Section 11.5.14.)

```
─────────────────────────── Example ───────────────────────────
gap> BindGlobal( "ClassesOfPrimeOrder", function( G, primes, N )
>       local ccl, p, syl, Greps, reps, r, cr;
>
>       ccl:= [];
>       for p in primes do
>         syl:= SylowSubgroup( G, p );
>         Greps:= [];
>         reps:= Filtered( PcConjugacyClassReps( syl ),
>                   r -> Order( r ) = p and not r in N );
>         for r in reps do
>           cr:= ConjugacyClass( G, r );
>           if ForAll( Greps, c -> c <> cr ) then
>             Add( Greps, cr );
>           fi;
>         od;
>         Append( ccl, Greps );
>       od;
>
>       return ccl;
> end );
```

The function IsGeneratorsOfTransPermGroup takes a *transitive* permutation group G and a list list of elements in G, and returns true if the elements in list generate G, and false otherwise. The main point is that the return value true requires the group generated by list to be transitive, and the check for transitivity is much cheaper than the test whether this group is equal to G.

```
─────────────────────────── Example ───────────────────────────
gap> if not IsBound( IsGeneratorsOfTransPermGroup) then
>       BindGlobal( "IsGeneratorsOfTransPermGroup", function( G, list )
>         local S;
>
>         if not IsTransitive( G ) then
>           Error( "<G> must be transitive on its moved points" );
>         fi;
>         S:= SubgroupNC( G, list );
>
>         return IsTransitive( S, MovedPoints( G ) ) and
```

```
>                  Size( S ) = Size( G );
>         end );
>     fi;
```

RatioOfNongenerationTransPermGroup takes a *transitive* permutation group G and two elements g and s of G, and returns the proportion $P(g,s)$. (The function tests the (non)generation only for representatives of $C_G(g)$-$C_G(s)$-double cosets. Note that for $c_1 \in C_G(g)$, $c_2 \in C_G(s)$, and a representative $r \in G$, we have $\langle g^{c_1 r c_2}, s \rangle = \langle g^r, s \rangle^{c_2}$.)

--- Example ---
```
gap> BindGlobal( "RatioOfNongenerationTransPermGroup", function( G, g, s )
>     local nongen, pair;
>
>     if not IsTransitive( G ) then
>       Error( "<G> must be transitive on its moved points" );
>     fi;
>     nongen:= 0;
>     for pair in DoubleCosetRepsAndSizes( G, Centralizer( G, g ),
>                   Centralizer( G, s ) ) do
>       if not IsGeneratorsOfTransPermGroup( G, [ s, g^pair[1] ] ) then
>         nongen:= nongen + pair[2];
>       fi;
>     od;
>
>     return nongen / Size( G );
> end );
```

Let G be a group, and let groups be a list $[G_1, G_2, \ldots, G_n]$ of permutation groups such that G_i describes the action of G on a set Ω_i, say. Moreover, we require that for $1 \leq i, j \leq n$, mapping the GeneratorsOfGroup list of G_i to that of G_j defines an isomorphism. DiagonalProductOfPermGroups takes groups as its argument, and returns the action of G on the disjoint union of $\Omega_1, \Omega_2, \ldots, \Omega_n$.

--- Example ---
```
gap> BindGlobal( "DiagonalProductOfPermGroups", function( groups )
>     local prodgens, deg, i, gens, D, pi;
>
>     prodgens:= GeneratorsOfGroup( groups[1] );
>     deg:= NrMovedPoints( prodgens );
>     for i in [ 2 .. Length( groups ) ] do
>       gens:= GeneratorsOfGroup( groups[i] );
>       D:= MovedPoints( gens );
>       pi:= MappingPermListList( D, [ deg+1 .. deg+Length( D ) ] );
>       deg:= deg + Length( D );
>       prodgens:= List( [ 1 .. Length( prodgens ) ],
>                   i -> prodgens[i] * gens[i]^pi );
>     od;
>
>     return Group( prodgens );
> end );
```

The following two functions are used to reduce checks of generation to class representatives of maximal order. Note that if $\langle s, g \rangle$ is a proper subgroup of G then also $\langle s^k, g \rangle$ is a proper subgroup of G, so we need not check powers s^k different from s in this situation.

For an ordinary character table `tbl`, the function `RepresentativesMaximallyCyclicSubgroups` returns a list of class positions, containing one class of generators for each class of maximally cyclic subgroups.

```
───────────────────────────── Example ─────────────────────────────
gap> BindGlobal( "RepresentativesMaximallyCyclicSubgroups", function( tbl )
>      local n, result, orders, p, pmap, i, j;
>
>      # Initialize.
>      n:= NrConjugacyClasses( tbl );
>      result:= BlistList( [ 1 .. n ], [ 1 .. n ] );
>
>      # Omit powers of smaller order.
>      orders:= OrdersClassRepresentatives( tbl );
>      for p in PrimeDivisors( Size( tbl ) ) do
>        pmap:= PowerMap( tbl, p );
>        for i in [ 1 .. n ] do
>          if orders[ pmap[i] ] < orders[i] then
>            result[ pmap[i] ]:= false;
>          fi;
>        od;
>      od;
>
>      # Omit Galois conjugates.
>      for i in [ 1 .. n ] do
>        if result[i] then
>          for j in ClassOrbit( tbl, i ) do
>            if i <> j then
>              result[j]:= false;
>            fi;
>          od;
>        fi;
>      od;
>
>      # Return the result.
>      return ListBlist( [ 1 .. n ], result );
> end );
```

Let `G` be a finite group, `tbl` be the ordinary character table of `G`, and `cols` be a list of class positions in `tbl`, for example the list returned by `RepresentativesMaximallyCyclicSubgroups`. The function `ClassesPerhapsCorrespondingToTableColumns` returns the sublist of those conjugacy classes of `G` for which the corresponding column in `tbl` can be contained in `cols`, according to element order and class size.

```
───────────────────────────── Example ─────────────────────────────
gap> BindGlobal( "ClassesPerhapsCorrespondingToTableColumns",
>      function( G, tbl, cols )
>      local orders, classes, invariants;
>
>      orders:= OrdersClassRepresentatives( tbl );
```

```
>       classes:= SizesConjugacyClasses( tbl );
>       invariants:= List( cols, i -> [ orders[i], classes[i] ] );
>
>       return Filtered( ConjugacyClasses( G ),
>           c -> [ Order( Representative( c ) ), Size(c) ] in invariants );
> end );
```

The next function computes, for a finite group G and subgroups $M_1, M_2, \ldots, M_n$ of G, an upper bound for $\max\{\sum_{i=1}^{n} \mu(g, G/M_i); g \in G \setminus Z(G)\}$. So if the M_i are the groups in $\mathbb{M}(G, s)$, for some $s \in G^{\times}$, then we get an upper bound for $\sigma(G, s)$.

The idea is that for $M \le G$ and $g \in G$ of order p, we have

$$\mu(g, G/M) = |g^G \cap M|/|g^G| \le \sum_{h \in C} |h^M|/|g^G| = \sum_{h \in C} |h^M| \cdot |C_G(g)|/|G|,$$

where C is a set of class representatives $h \in M$ of all those classes that satisfy $|h| = p$ and $|C_G(h)| = |C_G(g)|$, and in the case that G is a permutation group additionally that h and g move the same number of points. (Note that it is enough to consider elements of *prime* order.)

For computing the maximum of the rightmost term in this inequality, for $g \in G \setminus Z(G)$, we need not determine the G-conjugacy of class representatives in M. Of course we pay the price that the result may be larger than the leftmost term. However, if the maximal sum is in fact taken only over a single class representative, we are sure that equality holds. Thus we return a list of length two, containing the maximum of the right hand side of the above inequality and a Boolean value indicating whether this is equal to $\max\{\mu(g, G/M); g \in G \setminus Z(G)\}$ or just an upper bound.

The arguments for `UpperBoundFixedPointRatios` are the group `G`, a list `maxesclasses` such that the i-th entry is a list of conjugacy classes of M_i, which covers all classes of prime element order in M_i, and either `true` or `false`, where `true` means that the *exact* value of $\sigma(G, s)$ is computed, not just an upper bound; this can be much more expensive because of the conjugacy tests in G that may be necessary. (We try to reduce the number of conjugacy tests in this case, the second half of the code is not completely straightforward. The special treatment of conjugacy checks for elements with the same sets of fixed points is essential in the computation of $\sigma'(G, s)$ for $G = \mathrm{PGL}(6, 4)$; the critical input line is `ApproxPForOuterClassesInGL( 6, 4 )`, see Section 11.5.7. Currently the standard GAP conjugacy test for an element of order three and its inverse in $G \setminus G'$ requires hours of CPU time, whereas the check for existence of a conjugating element in the stabilizer of the common set of fixed points of the two elements is almost free of charge.)

`UpperBoundFixedPointRatios` can be used to compute $\sigma'(G, s)$ in the case that G is an automorphic extension of a simple group S, with $[G : S] = p$ a prime; if $\mathbb{M}'(G, s) = \{M_1, M_2, \ldots, M_n\}$ then the i-th entry of `maxesclasses` must contain only the classes of element order p in $M_i \setminus (M_i \cap S)$.

```
─────────────────────── Example ───────────────────────
gap> BindGlobal( "UpperBoundFixedPointRatios",
>     function( G, maxesclasses, truetest )
>     local myIsConjugate, invs, info, c, r, o, inv, pos, sums, max, maxpos,
>           maxlen, reps, split, i, found, j;
>
>     myIsConjugate:= function( G, x, y )
>       local movx, movy;
>
>       movx:= MovedPoints( x );
>       movy:= MovedPoints( y );
```

```
>         if movx = movy then
>           G:= Stabilizer( G, movx, OnSets );
>         fi;
>         return IsConjugate( G, x, y );
>       end;
>
>       invs:= [];
>       info:= [];
>
>       # First distribute the classes according to invariants.
>       for c in Concatenation( maxesclasses ) do
>         r:= Representative( c );
>         o:= Order( r );
>         # Take only prime order representatives.
>         if IsPrimeInt( o ) then
>           inv:= [ o, Size( Centralizer( G, r ) ) ];
>           # Omit classes that are central in 'G'.
>           if inv[2] <> Size( G ) then
>             if IsPerm( r ) then
>               Add( inv, NrMovedPoints( r ) );
>             fi;
>             pos:= First( [ 1 .. Length( invs ) ], i -> inv = invs[i] );
>             if pos = fail then
>               # This class is not 'G'-conjugate to any of the previous ones.
>               Add( invs, inv );
>               Add( info, [ [ r, Size( c ) * inv[2] ] ] );
>             else
>               # This class may be conjugate to an earlier one.
>               Add( info[ pos ], [ r, Size( c ) * inv[2] ] );
>             fi;
>           fi;
>         fi;
>       od;
>
>       if info = [] then
>         return [ 0, true ];
>       fi;
>
>       repeat
>         # Compute the contributions of the classes with the same invariants.
>         sums:= List( info, x -> Sum( List( x, y -> y[2] ) ) );
>         max:= Maximum( sums );
>         maxpos:= Filtered( [ 1 .. Length( info ) ], i -> sums[i] = max );
>         maxlen:= List( maxpos, i -> Length( info[i] ) );
>
>         # Split the sets with the same invariants if necessary
>         # and if we want to compute the exact value.
>         if truetest and not 1 in maxlen then
>           # Make one conjugacy test.
>           pos:= Position( maxlen, Minimum( maxlen ) );
>           reps:= info[ maxpos[ pos ] ];
>           if myIsConjugate( G, reps[1][1], reps[2][1] ) then
```

```
>                # Fuse the two classes.
>                reps[1][2]:= reps[1][2] + reps[2][2];
>                reps[2]:= reps[ Length( reps ) ];
>                Unbind( reps[ Length( reps ) ] );
>            else
>                # Split the list. This may require additional conjugacy tests.
>                Unbind( info[ maxpos[ pos ] ] );
>                split:= [ reps[1], reps[2] ];
>                for i in [ 3 .. Length( reps ) ] do
>                  found:= false;
>                  for j in split do
>                    if myIsConjugate( G, reps[i][1], j[1] ) then
>                       j[2]:= reps[i][2] + j[2];
>                       found:= true;
>                       break;
>                    fi;
>                  od;
>                  if not found then
>                    Add( split, reps[i] );
>                  fi;
>                od;
>
>                info:= Compacted( Concatenation( info,
>                                        List( split, x -> [ x ] ) ) );
>            fi;
>          fi;
>      until 1 in maxlen or not truetest;
>
>      return [ max / Size( G ), 1 in maxlen ];
> end );
```

Suppose that C_1, C_2, C_3 are conjugacy classes in G, and that we have to prove, for each $(x_1, x_2, x_3) \in C_1 \times C_2 \times C_3$, the existence of an element s in a prescribed class C of G such that $\langle x_1, s \rangle = \langle x_2, s \rangle = \langle x_2, s \rangle = G$ holds.

We have to check only representatives under the conjugation action of G on $C_1 \times C_2 \times C_3$. For each representative, we try a prescribed number of random elements in C. If this is successful then we are done. The following two functions implement this idea.

For a group G and a list $[g_1, g_2, \ldots, g_n]$ of elements in G, `OrbitRepresentativesProductOfClasses` returns a list $R(G, g_1, g_2, \ldots, g_n)$ of representatives of G-orbits on the Cartesian product $g_1^G \times g_2^G \times \cdots \times g_n^G$.

The idea behind this function is to choose $R(G, g_1) = \{(g_1)\}$ in the case $n = 1$, and, for $n > 1$,

$$R(G, g_1, g_2, \ldots, g_n) = \{(h_1, h_2, \ldots, h_n) \mid (h_1, h_2, \ldots, h_{n-1}) \in R(G, g_1, g_2, \ldots, g_{n-1}), h_n = g_n^d, \text{ for } d \in D\},$$

where D is a set of representatives of double cosets $C_G(g_n) \setminus G / \cap_{i=1}^{n-1} C_G(h_i)$.

```
─────────────────────── Example ───────────────────────
gap> BindGlobal( "OrbitRepresentativesProductOfClasses",
>     function( G, classreps )
>      local cents, n, orbreps;
>
>      cents:= List( classreps, x -> Centralizer( G, x ) );
```

```
>     n:= Length( classreps );
>
>     orbreps:= function( reps, intersect, pos )
>       if pos > n then
>         return [ reps ];
>       fi;
>       return Concatenation( List(
>           DoubleCosetRepsAndSizes( G, cents[ pos ], intersect ),
>             r -> orbreps( Concatenation( reps, [ classreps[ pos ]^r[1] ] ),
>                 Intersection( intersect, cents[ pos ]^r[1] ), pos+1 ) ) );
>     end;
>
>     return orbreps( [ classreps[1] ], cents[1], 2 );
> end );
```

The function `RandomCheckUniformSpread` takes a transitive permutation group G, a list of class representatives $g_i \in G$, an element $s \in G$, and a positive integer N. The return value is `true` if for each representative of G-orbits on the product of the classes g_i^G, a good conjugate of s is found in at most N random tests.

```
─────────── Example ───────────
gap> BindGlobal( "RandomCheckUniformSpread", function( G, classreps, s, try )
>     local elms, found, i, conj;
>
>     if not IsTransitive( G, MovedPoints( G ) ) then
>       Error( "<G> must be transitive on its moved points" );
>     fi;
>
>     # Compute orbit representatives of G on the direct product,
>     # and try to find a good conjugate of s for each representative.
>     for elms in OrbitRepresentativesProductOfClasses( G, classreps ) do
>       found:= false;
>       for i in [ 1 .. try ] do
>         conj:= s^Random( G );
>         if ForAll( elms,
>               x -> IsGeneratorsOfTransPermGroup( G, [ x, conj ] ) ) then
>           found:= true;
>           break;
>         fi;
>       od;
>       if not found then
>         return elms;
>       fi;
>     od;
>
>     return true;
> end );
```

Of course this approach is not suitable for *dis*proving the existence of s, but it is much cheaper than an exhaustive search in the class C. (Typically, $|C|$ is large whereas the $|C_i|$ are small.)

The following function can be used to verify that a given n-tuple $(x_1, x_2, \ldots, x_n)$ of elements in a group G has the property that for all elements $g \in G$, at least one x_i satisfies $\langle x_i, g \rangle$. The arguments

are a transitive permutation group *G*, a list of class representatives in *G*, and the *n*-tuple in question. The return value is a conjugate *g* of the given representatives that has the property if such an element exists, and `fail` otherwise.

```
─────────────────────────── Example ───────────────────────────
  gap> BindGlobal( "CommonGeneratorWithGivenElements",
  >     function( G, classreps, tuple )
  >      local inter, rep, repcen, pair;
  >
  >      if not IsTransitive( G, MovedPoints( G ) ) then
  >        Error( "<G> must be transitive on its moved points" );
  >      fi;
  >
  >      inter:= Intersection( List( tuple, x -> Centralizer( G, x ) ) );
  >      for rep in classreps do
  >        repcen:= Centralizer( G, rep );
  >        for pair in DoubleCosetRepsAndSizes( G, repcen, inter ) do
  >          if ForAll( tuple,
  >             x -> IsGeneratorsOfTransPermGroup( G, [ x, rep^pair[1] ] ) ) then
  >            return rep;
  >          fi;
  >        od;
  >      od;
  >
  >      return fail;
  > end );
```

11.4   Character-Theoretic Computations

In this section, we apply the functions introduced in Section 11.3.2 to the character tables of simple groups that are available in the GAP Character Table Library.

Our first examples are the sporadic simple groups, in Section 11.4.1, then their automorphism groups are considered in Section 11.4.3.

Then we consider those other simple groups for which GAP provides enough information for automatically computing an upper bound on $\sigma(G, s)$ –see Section 11.4.4– and their automorphic extensions –see Section 11.4.5.

After that, individual groups are considered.

11.4.1   Sporadic Simple Groups

The GAP Character Table Library contains the tables of maximal subgroups of all sporadic simple groups, so the primitive permutation characters of all sporadic simple groups can be computed via the function `PrimitivePermutationCharacters`.

```
─────────────────────────── Example ───────────────────────────
  gap> sporinfo:= [];;
  gap> spornames:= AllCharacterTableNames( IsSporadicSimple, true,
  >                                        IsDuplicateTable, false );;
  gap> for tbl in List( spornames, CharacterTable ) do
  >       info:= ProbGenInfoSimple( tbl );
  >       if info <> fail then
```

```
>          # keep the table columns narrow
>          if info[2] <= 10^-13 then
>            info[2]:= "<= 10^-13";
>          fi;
>          if info[3] >= 10^13 then
>            info[3]:= ">= 10^13";
>          fi;
>          Add( sporinfo, info );
>        fi;
>      od;
```

We show the result as a formatted table.

```
———————————————————————— Example ————————————————————————
gap> PrintFormattedArray( sporinfo );
      B         <= 10^-13        >= 10^13        [ "47A" ]      [ 1 ]
    Co1     421/1545600             3671        [ "35A" ]      [ 4 ]
    Co2           1/270              269        [ "23A" ]      [ 1 ]
    Co3         64/6325               98        [ "21A" ]      [ 4 ]
    F3+ 1/269631216855    269631216854        [ "29A" ]      [ 1 ]
   Fi22          43/585               13        [ "16A" ]      [ 7 ]
   Fi23     2651/2416635              911        [ "23A" ]      [ 2 ]
     HN         4/34375             8593        [ "19A" ]      [ 1 ]
     HS         64/1155               18        [ "15A" ]      [ 2 ]
     He          3/595              198        [ "14C" ]      [ 3 ]
     J1            1/77               76        [ "19A" ]      [ 1 ]
     J2            5/28                5        [ "10C" ]      [ 3 ]
     J3           2/153               76        [ "19A" ]      [ 2 ]
     J4    1/1647124116     1647124115        [ "29A" ]      [ 1 ]
     Ly      1/35049375        35049374        [ "37A" ]      [ 1 ]
      M         <= 10^-13        >= 10^13        [ "59A" ]      [ 1 ]
    M11             1/3                2        [ "11A" ]      [ 1 ]
    M12             1/3                2        [ "10A" ]      [ 3 ]
    M22            1/21               20        [ "11A" ]      [ 1 ]
    M23          1/8064             8063        [ "23A" ]      [ 1 ]
    M24        108/1265               11        [ "21A" ]      [ 2 ]
    McL       317/22275               70 [ "15A", "30A" ] [ 3, 3 ]
     ON        10/30723             3072        [ "31A" ]      [ 2 ]
     Ru          1/2880             2879        [ "29A" ]      [ 1 ]
    Suz        141/5720               40        [ "14A" ]      [ 3 ]
     Th        2/267995           133997 [ "27A", "27B" ] [ 2, 2 ]
```

Here are the exact values for the Baby Monster and the Monster.

```
———————————————————————— Example ————————————————————————
gap> ProbGenInfoSimple( CharacterTable( "B" ) );
[ "B", 1/174702778623598780219392000000,
  174702778623598780219391999999, [ "47A" ], [ 1 ] ]
gap> ProbGenInfoSimple( CharacterTable( "M" ) );
[ "M", 1/56220076312551339782253479235319832248320000000000,
  56220076312551339782253479235319832248319999999999, [ "59A" ], [ 1 ]
 ]
```

We see that in all these cases, $\sigma(G) < 1/2$ and thus $\mathscr{P}(G) \geq 2$, and all sporadic simple groups G except $G = M_{11}$ and $G = M_{12}$ satisfy $\sigma(G) < 1/3$. See 11.5.9 and 11.5.10 for a proof that also these two groups have uniform spread at least three.

The structures and multiplicities of the maximal subgroups containing *s* are as follows.

```
―――――――――――――――――――――――― Example ――――――――――――――――――――――――
gap> for entry in sporinfo do
>        DisplayProbGenMaxesInfo( CharacterTable( entry[1] ), entry[4] );
> od;
B, 47A: 47:23  (1)
Co1, 35A: (A5xJ2):2  (1)
          (A6xU3(3)):2  (2)
          (A7xL2(7)):2  (1)
Co2, 23A: M23  (1)
Co3, 21A: U3(5).3.2  (2)
          L3(4).D12  (1)
          s3xpsl(2,8).3  (1)
F3+, 29A: 29:14  (1)
Fi22, 16A: 2^10:m22  (1)
           (2x2^(1+8)):U4(2):2  (1)
           2F4(2)'  (4)
           2^(5+8):(S3xA6)  (1)
Fi23, 23A: 2..11.m23  (1)
           L2(23)  (1)
HN, 19A: U3(8).3_1  (1)
HS, 15A: A8.2  (1)
         5:4xa5  (1)
He, 14C: 2^1+6.psl(3,2)  (1)
         7^2:2psl(2,7)  (1)
         7^(1+2):(S3x3)  (1)
J1, 19A: 19:6  (1)
J2, 10C: 2^1+4b:a5  (1)
         a5xd10  (1)
         5^2:D12  (1)
J3, 19A: L2(19)  (1)
         J3M3  (1)
J4, 29A: frob  (1)
Ly, 37A: 37:18  (1)
M, 59A: 59:29  (1)
M11, 11A: L2(11)  (1)
M12, 10A: A6.2^2  (1)
          M12M4  (1)
          2xS5  (1)
M22, 11A: L2(11)  (1)
M23, 23A: 23:11  (1)
M24, 21A: L3(4).3.2_2  (1)
          2^6:(psl(3,2)xs3)  (1)
McL, 15A: 3^(1+4):2S5  (1)
          2.A8  (1)
          5^(1+2):3:8  (1)
McL, 30A: 3^(1+4):2S5  (1)
          2.A8  (1)
          5^(1+2):3:8  (1)
```

```
ON, 31A: L2(31)  (1)
          ONM8  (1)
Ru, 29A: L2(29)  (1)
Suz, 14A: J2.2  (2)
          (a4xpsl(3,4)):2  (1)
Th, 27A: ThN3B  (1)
          ThM7  (1)
Th, 27B: ThN3B  (1)
          ThM7  (1)
```

Essentially the same approach is taken in [GM01]. However, there s is restricted to classes of prime order. Thus the results in the above table are better for J_2, HS, M_{24}, McL, He, Suz, Co_3, Fi_{22}, Ly, Th, Co_1, and J_4. Besides that, the value $10\,999$ claimed in [GM01] for $\mathscr{S}(HN)$ is not correct.

11.4.2 No longer necessary computations for the Baby Monster and the Monster

At the time when [BGK08] was written, not all character tables of maximal subgroups of the Baby Monster and the Monster were available, hence it was not straightforward to compute the primitive permutation characters of these two groups.

In this section, we show the original computations for B and a modification of the computations for M.

For the sporadic simple groups B and M, we choose suitable elements s. If $G = B$ and $s \in G$ is of order 47 then, by [Wil99], $\mathbb{M}(G,s) = \{47:23\}$.

```
————————————————————— Example ——————————————————————
 gap> SigmaFromMaxes( CharacterTable( "B" ), "47A",
 >        [ CharacterTable( "47:23" ) ], [ 1 ] );
 1/174702778623598780219392000000
```

If $G = M$ and $s \in G$ is of order 59 then, by [DLPP24], $\mathbb{M}(G,s) = \{59:29\}$.

```
————————————————————— Example ——————————————————————
 gap> SigmaFromMaxes( CharacterTable( "M" ), "59A",
 >        [ CharacterTable( "59:29" ) ], [ 1 ] );
 1/56220076312551339782253479235319832248320000000000
```

Before the publication of [DLPP24], it had been claimed that $\mathbb{M}(G,s) = \{L_2(59)\}$, see [HW04]. Under that assumption, we get a bound that is worse. In this situation, the "permutation character" is not uniquely determined by the character tables, but all possibilities lead to the same value for $\sigma(G)$.

```
————————————————————— Example ——————————————————————
 gap> t:= CharacterTable( "M" );;
 gap> s:= CharacterTable( "L2(59)" );;
 gap> pi:= PossiblePermutationCharacters( s, t );;
 gap> Length( pi );
 5
 gap> spos:= Position( OrdersClassRepresentatives( t ), 59 );
 152
 gap> Set( pi, x -> Maximum( ApproxP( [ x ], spos ) ) );
 [ 1/3385007637938037777290625 ]
```

11.4.3 Automorphism Groups of Sporadic Simple Groups

Next we consider the automorphism groups of the sporadic simple groups. There are exactly 12 cases where nontrivial outer automorphisms exist, and then the simple group *S* has index 2 in its automorphism group *G*.

```
——————————————— Example ———————————————
  gap> sporautnames:= AllCharacterTableNames( IsSporadicSimple, true,
  >                     IsDuplicateTable, false,
  >                     OfThose, AutomorphismGroup );;
  gap> sporautnames:= Difference( sporautnames, spornames );
  [ "F3+.2", "Fi22.2", "HN.2", "HS.2", "He.2", "J2.2", "J3.2", "M12.2",
    "M22.2", "McL.2", "ON.2", "Suz.2" ]
```

First we compute the values $\sigma'(G, s)$, for the same $s \in S$ that were chosen for the simple group *S* in Section 11.4.1.

For six of the groups *G* in question, the character tables of all maximal subgroups are available in the GAP Character Table Library. In these cases, the values $\sigma'(G, s)$ can be computed using `ProbGenInfoAlmostSimple`.

(The above statement can meanwhile be replaced by the statement that the character tables of all maximal subgroups are available for all twelve groups. We show the table results for all these groups but keep the individual computations from the original computations.)

```
——————————————— Example ———————————————
  gap> sporautinfo:= [];;
  gap> fails:= [];;
  gap> for name in sporautnames do
  >       tbl:= CharacterTable( name{ [ 1 .. Position( name, '.' ) - 1 ] } );
  >       tblG:= CharacterTable( name );
  >       info:= ProbGenInfoSimple( tbl );
  >       info:= ProbGenInfoAlmostSimple( tbl, tblG,
  >           List( info[4], x -> Position( AtlasClassNames( tbl ), x ) ) );
  >       if info = fail then
  >         Add( fails, name );
  >       else
  >         Add( sporautinfo, info );
  >       fi;
  >     od;
  gap> PrintFormattedArray( sporautinfo );
     F3+.2        0         [ "29AB" ]      [ 1 ]
    Fi22.2   251/3861       [ "16AB" ]      [ 7 ]
      HN.2     1/6875       [ "19AB" ]      [ 1 ]
      HS.2     36/275        [ "15A" ]      [ 2 ]
      He.2    37/9520       [ "14CD" ]      [ 3 ]
      J2.2       1/15       [ "10CD" ]      [ 3 ]
      J3.2     1/1080       [ "19AB" ]      [ 1 ]
     M12.2       4/99        [ "10A" ]      [ 1 ]
     M22.2       1/21       [ "11AB" ]      [ 1 ]
     McL.2       1/63 [ "15AB", "30AB" ] [ 3, 3 ]
      ON.2    1/84672       [ "31AB" ]      [ 1 ]
     Suz.2 661/46332        [ "14A" ]      [ 3 ]
```

Note that for $S = McL$, the bound $\sigma'(G,s)$ for $G = S.2$ (in the second column) is worse than the bound for the simple group S.

The structures and multiplicities of the maximal subgroups containing s are as follows.

```
―――――――――――――――――――――― Example ――――――――――――――――――――――
 gap> for entry in sporautinfo do
 >        DisplayProbGenMaxesInfo( CharacterTable( entry[1] ), entry[3] );
 > od;
 F3+.2, 29AB: F3+  (1)
              frob  (1)
 Fi22.2, 16AB: Fi22  (1)
               Fi22.2M4  (1)
               (2x2^(1+8)):(U4(2):2x2)  (1)
               2F4(2)'.2  (4)
               2^(5+8):(S3xS6)  (1)
 HN.2, 19AB: HN  (1)
             U3(8).6  (1)
 HS.2, 15A: HS  (1)
            S8x2  (1)
            5:4xS5  (1)
 He.2, 14CD: He  (1)
             2^(1+6)_+.L3(2).2  (1)
             7^2:2.L2(7).2  (1)
             7^(1+2):(S3x6)  (1)
 J2.2, 10CD: J2  (1)
             2^(1+4).S5  (1)
             (A5xD10).2  (1)
             5^2:(4xS3)  (1)
 J3.2, 19AB: J3  (1)
             19:18  (1)
 M12.2, 10A: M12  (1)
             (2^2xA5):2  (1)
 M22.2, 11AB: M22  (1)
              L2(11).2  (1)
 McL.2, 15AB: McL  (1)
              3^(1+4):4S5  (1)
              Isoclinic(2.A8.2)  (1)
              5^(1+2):(24:2)  (1)
 McL.2, 30AB: McL  (1)
              3^(1+4):4S5  (1)
              Isoclinic(2.A8.2)  (1)
              5^(1+2):(24:2)  (1)
 ON.2, 31AB: ON  (1)
             31:30  (1)
 Suz.2, 14A: Suz  (1)
             J2.2x2  (2)
             (A4xL3(4):2_3):2  (1)
```

Note that the maximal subgroups $L_2(19)$ of J_3 do not extend to $J_3.2$ and that a class of maximal subgroups of the type $19:18$ appears in $J_3.2$ whose intersection with J_3 is not maximal in J_3. Similarly, the maximal subgroups $A_6.2^2$ of M_{12} do not extend to $M_{12}.2$.

For the other six groups, we use individual computations.

In the case $S = Fi'_{24}$, the unique maximal subgroup $29 : 14$ that contains an element s of order 29 extends to a group of the type $29 : 28$ in Fi_{24}, which is a nonsplit extension of $29 : 14$.

```
——————————————— Example ———————————————
gap> SigmaFromMaxes( CharacterTable( "Fi24'.2" ), "29AB",
>         [ CharacterTable( "29:28" ) ], [ 1 ], "outer" );
0
```

In the case $S = Fi_{22}$, there are four classes of maximal subgroups that contain s of order 16. They extend to $G = Fi_{22}.2$, and none of the *novelties* in G (i. e., subgroups of G that are maximal in G but whose intersections with S are not maximal in S) contains s, cf. [CCN+85, p. 163].

```
——————————————— Example ———————————————
gap> 16 in OrdersClassRepresentatives( CharacterTable( "U4(2).2" ) );
false
gap> 16 in OrdersClassRepresentatives( CharacterTable( "G2(3).2" ) );
false
```

The character tables of three of the four extensions are available in the GAP Character Table Library. The permutation character on the cosets of the fourth extension can be obtained as the extension of the permutation character of S on the cosets of its maximal subgroup of the type $2^{5+8} : (S_3 \times A_6)$.

```
——————————————— Example ———————————————
gap> t2:= CharacterTable( "Fi22.2" );;
gap> prim:= List( [ "Fi22.2M4", "(2x2^(1+8)):(U4(2):2x2)", "2F4(2)" ],
>         n -> PossiblePermutationCharacters( CharacterTable( n ), t2 ) );;
gap> t:= CharacterTable( "Fi22" );;
gap> pi:= PossiblePermutationCharacters(
>             CharacterTable( "2^(5+8):(S3xA6)" ), t );
[ Character( CharacterTable( "Fi22" ),
  [ 3648645, 56133, 10629, 2245, 567, 729, 405, 81, 549, 165, 133,
      37, 69, 20, 27, 81, 9, 39, 81, 19, 1, 13, 33, 13, 1, 0, 13, 13,
      5, 1, 0, 0, 0, 8, 4, 0, 0, 9, 3, 15, 3, 1, 1, 1, 1, 3, 3, 1, 0,
      0, 0, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2 ] ) ]
gap> torso:= CompositionMaps( pi[1], InverseMap( GetFusionMap( t, t2 ) ) );
[ 3648645, 56133, 10629, 2245, 567, 729, 405, 81, 549, 165, 133, 37,
  69, 20, 27, 81, 9, 39, 81, 19, 1, 13, 33, 13, 1, 0, 13, 13, 5, 1,
  0, 0, 0, 8, 4, 0, 9, 3, 15, 3, 1, 1, 1, 3, 3, 1, 0, 0, 2, 1, 0, 0,
  0, 0, 0, 0, 1, 1, 2 ]
gap> ext:= PermChars( t2, rec( torso:= torso ) );;
gap> Add( prim, ext );
gap> prim:= Concatenation( prim );;  Length( prim );
4
gap> spos:= Position( OrdersClassRepresentatives( t2 ), 16 );;
gap> List( prim, x -> x[ spos ] );
[ 1, 1, 4, 1 ]
gap> sigma:= ApproxP( prim, spos );;
gap> Maximum( sigma{ Difference( PositionsProperty(
>                         OrdersClassRepresentatives( t2 ), IsPrimeInt ),
>                         ClassPositionsOfDerivedSubgroup( t2 ) ) } );
251/3861
```

In the case $S = HN$, the unique maximal subgroup $U_3(8).3$ that contains the fixed element s of order 19 extends to a group of the type $U_3(8).6$ in $HN.2$.

```
─────────────────────────── Example ───────────────────────────
gap> SigmaFromMaxes( CharacterTable( "HN.2" ), "19AB",
>          [ CharacterTable( "U3(8).6" ) ], [ 1 ], "outer" );
1/6875
```

In the case $S = HS$, there are two classes of maximal subgroups that contain s of order 15. They extend to $G = HS.2$, and none of the novelties in G contains s (cf. [CCN+85, p. 80]).

```
─────────────────────────── Example ───────────────────────────
gap> SigmaFromMaxes( CharacterTable( "HS.2" ), "15A",
>        [ CharacterTable( "S8x2" ),
>          CharacterTable( "5:4" ) * CharacterTable( "A5.2" ) ], [ 1, 1 ],
>        "outer" );
36/275
```

In the case $S = He$, there are three classes of maximal subgroups that contain s in the class 14C. They extend to $G = He.2$, and none of the novelties in G contains s (cf. [CCN+85, p. 104]). We compute the extensions of the corresponding primitive permutation characters of S.

```
─────────────────────────── Example ───────────────────────────
gap> t:= CharacterTable( "He" );;
gap> t2:= CharacterTable( "He.2" );;
gap> prim:= PrimitivePermutationCharacters( t );;
gap> spos:= Position( AtlasClassNames( t ), "14C" );;
gap> prim:= Filtered( prim, x -> x[ spos ] <> 0 );;
gap> map:= InverseMap( GetFusionMap( t, t2 ) );;
gap> torso:= List( prim, pi -> CompositionMaps( pi, map ) );
[ [ 187425, 945, 449, 0, 21, 21, 25, 25, 0, 0, 5, 0, 0, 7, 1, 0, 0,
      1, 0, 1, 0, 0, 0, 0, 0, 0 ],
  [ 244800, 0, 64, 0, 84, 0, 0, 16, 0, 0, 4, 24, 45, 3, 4, 0, 0, 0,
      0, 1, 0, 0, 0, 0, 0, 0 ],
  [ 652800, 0, 512, 120, 72, 0, 0, 0, 0, 0, 8, 8, 22, 1, 0, 0, 0, 0,
      0, 1, 0, 0, 1, 1, 2, 0 ] ]
gap> ext:= List( torso, x -> PermChars( t2, rec( torso:= x ) ) );
[ [ Character( CharacterTable( "He.2" ),
      [ 187425, 945, 449, 0, 21, 21, 25, 25, 0, 0, 5, 0, 0, 7, 1, 0,
          0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 315, 15, 0, 0, 3, 7, 7, 3, 0,
          0, 0, 1, 1, 0, 1, 1, 0, 0, 0 ] ) ],
  [ Character( CharacterTable( "He.2" ),
      [ 244800, 0, 64, 0, 84, 0, 0, 16, 0, 0, 4, 24, 45, 3, 4, 0, 0,
          0, 0, 1, 0, 0, 0, 0, 0, 0, 360, 0, 0, 0, 6, 0, 0, 0, 0, 0,
          3, 2, 2, 0, 0, 0, 0, 0, 0 ] ) ],
  [ Character( CharacterTable( "He.2" ),
      [ 652800, 0, 512, 120, 72, 0, 0, 0, 0, 0, 8, 8, 22, 1, 0, 0, 0,
          0, 0, 1, 0, 0, 1, 1, 2, 0, 480, 0, 120, 0, 12, 0, 0, 0, 0,
          0, 4, 0, 0, 0, 0, 0, 0, 1, 1 ] ) ] ]
gap> spos:= Position( AtlasClassNames( t2 ), "14CD" );;
gap> sigma:= ApproxP( Concatenation( ext ), spos );;
gap> Maximum( sigma{ Difference( PositionsProperty(
>                        OrdersClassRepresentatives( t2 ), IsPrimeInt ),
>                        ClassPositionsOfDerivedSubgroup( t2 ) ) } );
37/9520
```

In the case $S = O'N$, the two classes of maximal subgroups of the type $L_2(31)$ do not extend to $G = O'N.2$, and a class of novelties of the structure $31 : 30$ appears (see [CCN$^+$85, p. 132]).

```
────────────────── Example ──────────────────
gap> SigmaFromMaxes( CharacterTable( "ON.2" ), "31AB",
>         [ CharacterTable( "P:Q", [ 31, 30 ] ) ], [ 1 ], "outer" );
1/84672
```

Now we consider also $\sigma(G, \hat{s})$, for suitable $\hat{s} \in G \setminus S$; this yields lower bounds for the spread of the nonsimple groups G. (These results are shown in the last two columns of [BGK08, Table 9].)

As above, we use the known character tables of the maximal subgroups in order to compute the optimal choice for $\hat{s} \in G \setminus S$. (We may use the function ProbGenInfoSimple although the groups are not simple; all we need is that the relevant maximal subgroups are self-normalizing.)

```
────────────────── Example ──────────────────
gap> sporautinfo2:= [];;
gap> for name in List( sporautinfo, x -> x[1] ) do
>       Add( sporautinfo2, ProbGenInfoSimple( CharacterTable( name ) ) );
>    od;
gap> PrintFormattedArray( sporautinfo2 );
   F3+.2     19/5684   299      [ "42E" ]    [ 10 ]
  Fi22.2 1165/20592    17      [ "24G" ]    [ 3 ]
     HN.2      1/1425 1424      [ "24B" ]    [ 4 ]
     HS.2     21/550   26      [ "20C" ]    [ 4 ]
     He.2     33/4165  126      [ "24A" ]    [ 2 ]
     J2.2          1/15   14      [ "14A" ]    [ 1 ]
     J3.2    77/10260  133      [ "34A" ]    [ 1 ]
     M12.2    113/495    4      [ "12B" ]    [ 3 ]
     M22.2       8/33     4      [ "10A" ]    [ 4 ]
     McL.2      1/135  134      [ "22A" ]    [ 1 ]
      ON.2  61/109368 1792 [ "22A", "38A" ] [ 1, 1 ]
     Suz.2      1/351  350      [ "28A" ]    [ 1 ]
gap> for entry in sporautinfo2 do
>       DisplayProbGenMaxesInfo( CharacterTable( entry[1] ), entry[4] );
> od;
F3+.2, 42E: 2^12.M24   (2)
            2^2.U6(2):S3x2   (1)
            2^(3+12).(L3(2)xS6)   (2)
            (S3xS3xG2(3)):2   (1)
            S6xL2(8):3   (1)
            7:6xS7   (1)
            7^(1+2)_+:(6xS3).2   (2)
Fi22.2, 24G: Fi22.2M4   (1)
             2^(5+8):(S3xS6)   (1)
             3^5:(2xU4(2).2)   (1)
HN.2, 24B: 2^(1+8)_+.(A5xA5).2^2   (1)
           5^2.5.5^2.4S5   (2)
           HN.2M13   (1)
HS.2, 20C: (2xA6.2^2).2   (1)
           HS.2N5   (2)
           5:4xS5   (1)
He.2, 24A: 2^(1+6)_+.L3(2).2   (1)
           S4xL3(2).2   (1)
```

```
J2.2, 14A: L3(2).2x2  (1)
J3.2, 34A: L2(17)x2  (1)
M12.2, 12B: L2(11).2  (1)
            2^3.(S4x2)  (1)
            3^(1+2):D8  (1)
M22.2, 10A: M22.2M4  (1)
            A6.2^2  (1)
            L2(11).2  (2)
McL.2, 22A: 2xM11  (1)
ON.2, 22A: J1x2  (1)
ON.2, 38A: J1x2  (1)
Suz.2, 28A: (A4xL3(4):2_3):2  (1)
```

In the other six cases, we do not have the complete lists of primitive permutation characters, so we choose a suitable element $\hat{s}$ for each group. It is sufficient to prescribe $|\hat{s}|$, as follows.

——————————— Example ———————————

```
gap> sporautchoices:= [
>         [ "Fi22",  "Fi22.2",  42 ],
>         [ "Fi24'", "Fi24'.2", 46 ],
>         [ "He",    "He.2",    42 ],
>         [ "HN",    "HN.2",    44 ],
>         [ "HS",    "HS.2",    30 ],
>         [ "ON",    "ON.2",    38 ], ];;
```

First we list the maximal subgroups of the corresponding simple groups that contain the square of $\hat{s}$.

——————————— Example ———————————

```
gap> for triple in sporautchoices do
>       tbl:= CharacterTable( triple[1] );
>       tbl2:= CharacterTable( triple[2] );
>       spos2:= PowerMap( tbl2, 2,
>           Position( OrdersClassRepresentatives( tbl2 ), triple[3] ) );
>       spos:= Position( GetFusionMap( tbl, tbl2 ), spos2 );
>       DisplayProbGenMaxesInfo( tbl, AtlasClassNames( tbl ){ [ spos ] } );
>    od;
Fi22, 21A: O8+(2).3.2  (1)
            S3xU4(3).2_2  (1)
            A10.2  (1)
            A10.2  (1)
F3+, 23A: Fi23  (1)
           F3+M7  (1)
He, 21B: 3.A7.2  (1)
          7^(1+2):(S3x3)  (1)
          7:3xpsl(3,2)  (2)
HN, 22A: 2.HS.2  (1)
HS, 15A: A8.2  (1)
          5:4xa5  (1)
ON, 19B: L3(7).2  (1)
          ONM2  (1)
          J1  (1)
```

According to [CCN+85], exactly the following maximal subgroups of the simple group S in the above list do *not* extend to Aut(S): The two S_{10} type subgroups of Fi_{22} and the two $L_3(7).2$ type subgroups of $O'N$.

Furthermore, the following maximal subgroups of Aut(S) with the property that the intersection with S is not maximal in S have to be considered whether they contain s': $G_2(3).2$ and $3^5 : (2 \times U_4(2).2)$ in $Fi_{22}.2$. (Note that the order of the $7^{1+2}_+ : (3 \times D_{16})$ type subgroup in $O'N.2$ is obviously not divisible by 19.)

```
——————————————— Example ———————————————
gap> 42 in OrdersClassRepresentatives( CharacterTable( "G2(3).2" ) );
false
gap> Size( CharacterTable( "U4(2)" ) ) mod 7 = 0;
false
```

So we take the extensions of the above maximal subgroups, as described in [CCN+85].

```
——————————————— Example ———————————————
gap> SigmaFromMaxes( CharacterTable( "Fi22.2" ), "42A",
>     [ CharacterTable( "O8+(2).3.2" ) * CharacterTable( "Cyclic", 2 ),
>       CharacterTable( "S3" ) * CharacterTable( "U4(3).(2^2)_{122}" ) ],
>     [ 1, 1 ] );
163/1170
gap> SigmaFromMaxes( CharacterTable( "Fi24'.2" ), "46A",
>     [ CharacterTable( "Fi23" ) * CharacterTable( "Cyclic", 2 ),
>       CharacterTable( "2^12.M24" ) ],
>     [ 1, 1 ] );
566/5481
gap> SigmaFromMaxes( CharacterTable( "He.2" ), "42A",
>     [ CharacterTable( "3.A7.2" ) * CharacterTable( "Cyclic", 2 ),
>       CharacterTable( "7^(1+2):(S3x6)" ),
>       CharacterTable( "7:6" ) * CharacterTable( "L3(2)" ) ],
>     [ 1, 1, 1 ] );
1/119
gap> SigmaFromMaxes( CharacterTable( "HN.2" ), "44A",
>     [ CharacterTable( "4.HS.2" ) ],
>     [ 1 ] );
997/192375
gap> SigmaFromMaxes( CharacterTable( "HS.2" ), "30A",
>     [ CharacterTable( "S8" ) * CharacterTable( "C2" ),
>       CharacterTable( "5:4" ) * CharacterTable( "S5" ) ],
>     [ 1, 1 ] );
36/275
gap> SigmaFromMaxes( CharacterTable( "ON.2" ), "38A",
>     [ CharacterTable( "J1" ) * CharacterTable( "C2" ) ],
>     [ 1 ] );
61/109368
```

11.4.4 Other Simple Groups – Easy Cases

We are interested in simple groups G for which `ProbGenInfoSimple` does not guarantee $\mathscr{S}(G) \geq 3$. So we examine the remaining tables of simple groups in the GAP Character Table Library, and distinguish the following three cases: Either `ProbGenInfoSimple` yields the lower bound at least

three, or a smaller bound, or the computation of a lower bound fails because not enough information is available to compute the primitive permutation characters.

```
――――――――――――――――― Example ―――――――――――――――――
  gap> names:= AllCharacterTableNames( IsSimple, true, IsAbelian, false,
  >                                    IsDuplicateTable, false );;
  gap> names:= Difference( names, spornames );;
  gap> fails:= [];;
  gap> lessthan3:= [];;
  gap> atleast3:= [];;
  gap> for name in names do
  >        tbl:= CharacterTable( name );
  >        info:= ProbGenInfoSimple( tbl );
  >        if info = fail then
  >          Add( fails, name );
  >        elif info[3] < 3 then
  >          Add( lessthan3, info );
  >        else
  >          Add( atleast3, info );
  >        fi;
  >     od;
```

For the following simple groups, (currently) not enough information is available in the GAP Character Table Library and in the GAP Library of Tables of Marks, for computing a lower bound for $\sigma(G)$. Some of these groups will be dealt with in later sections, and for the other groups, the bounds derived with theoretical arguments in [BGK08] are sufficient, so we need no GAP computations for them.

```
――――――――――――――――― Example ―――――――――――――――――
  gap> fails;
  [ "2E6(2)", "2F4(8)", "3D4(3)", "3D4(4)", "A14", "A15", "A16", "A17",
    "A18", "A19", "E6(2)", "F4(3)", "G2(7)", "L4(4)", "L4(5)", "L4(9)",
    "L5(3)", "L8(2)", "O10+(2)", "O10+(3)", "O10-(2)", "O10-(3)",
    "O12+(2)", "O12+(3)", "O12-(2)", "O12-(3)", "O7(5)", "O8+(7)",
    "O8-(3)", "O9(3)", "R(27)", "S10(2)", "S12(2)", "S4(7)", "S4(8)",
    "S4(9)", "S6(4)", "S6(5)", "S8(3)", "U4(4)", "U4(5)", "U5(3)",
    "U5(4)", "U6(4)", "U7(2)" ]
```

The following simple groups appear in [BGK08, Table 1–6]. More detailed computations can be found in the sections 11.5.2, 11.5.3, 11.5.4, 11.5.12, 11.5.13, 11.5.20, 11.5.23, 11.5.24.

```
――――――――――――――――― Example ―――――――――――――――――
  gap> PrintFormattedArray( lessthan3 );
        A5       1/3 2                [ "5A" ]          [ 1 ]
        A6       2/3 1                [ "5A" ]          [ 2 ]
        A7       2/5 2                [ "7A" ]          [ 2 ]
     O7(3)   199/351 1               [ "14A" ]          [ 3 ]
    O8+(2)   334/315 0 [ "15A", "15B", "15C" ] [ 7, 7, 7 ]
    O8+(3)  863/1820 2 [ "20A", "20B", "20C" ] [ 8, 8, 8 ]
     S6(2)       4/7 1                [ "9A" ]          [ 4 ]
     S8(2)      8/15 1               [ "17A" ]          [ 3 ]
     U4(2)     21/40 1               [ "12A" ]          [ 2 ]
     U4(3)    53/135 2                [ "7A" ]          [ 7 ]
```

For the following simple groups G, the inequality $\sigma(G) < 1/3$ follows from the loop above. The columns show the name of G, the values $\sigma(G)$ and $\mathscr{S}(G)$, the class names of s for which these values are attained, and $|\mathbb{M}(G, s)|$.

(We increase the line length for this table. Even with this width, the entry for the group $L_7(2)$ would not fit on one screen line, we show it separately below.)

```
──────────── Example ────────────
gap> oldsize:= SizeScreen();;
gap> SizeScreen( [ 80 ] );;
gap> PrintFormattedArray( Filtered( atleast3, l -> l[1] <> "L7(2)" ) );
  2F4(2)'  118/1755   14                                [ "16A" ]               [ 2 ]
   3D4(2)    1/5292 5291                                [ "13A" ]               [ 1 ]
      A10      3/10    3                                [ "21A" ]               [ 1 ]
      A11     2/105   52                                [ "11A" ]               [ 2 ]
      A12       2/9    4                                [ "35A" ]               [ 1 ]
      A13    4/1155  288                                [ "13A" ]               [ 5 ]
       A8      3/14    4                                [ "15A" ]               [ 1 ]
       A9      9/35    3                         [ "9A", "9B" ]              [ 4, 4 ]
     F4(2)     9/595   66                                [ "13A" ]               [ 5 ]
     G2(3)       1/7    6                                [ "13A" ]               [ 3 ]
     G2(4)      1/21   20                                [ "13A" ]               [ 2 ]
     G2(5)      1/31   30                        [ "7A", "21A" ]             [ 10, 1 ]
   L2(101)     1/101  100                       [ "51A", "17A" ]              [ 1, 1 ]
   L2(103)   53/5253   99                [ "52A", "26A", "13A" ]           [ 1, 1, 1 ]
   L2(107)   55/5671  103 [ "54A", "27A", "18A", "9A", "6A" ] [ 1, 1, 1, 1, 1 ]
   L2(109)     1/109  108                       [ "55A", "11A" ]              [ 1, 1 ]
    L2(11)      7/55    7                                 [ "6A" ]               [ 1 ]
   L2(113)     1/113  112                       [ "57A", "19A" ]              [ 1, 1 ]
   L2(121)     1/121  120                                [ "61A" ]               [ 1 ]
   L2(125)     1/125  124         [ "63A", "21A", "9A", "7A" ]     [ 1, 1, 1, 1 ]
    L2(13)      1/13   12                                 [ "7A" ]               [ 1 ]
    L2(16)      1/15   14                                [ "17A" ]               [ 1 ]
    L2(17)      1/17   16                                 [ "9A" ]               [ 1 ]
    L2(19)    11/171   15                                [ "10A" ]               [ 1 ]
    L2(23)    13/253   19                        [ "6A", "12A" ]             [ 1, 1 ]
    L2(25)      1/25   24                                [ "13A" ]               [ 1 ]
    L2(27)     5/117   23                        [ "7A", "14A" ]             [ 1, 1 ]
    L2(29)      1/29   28                                [ "15A" ]               [ 1 ]
    L2(31)    17/465   27                        [ "8A", "16A" ]             [ 1, 1 ]
    L2(32)      1/31   30                 [ "3A", "11A", "33A" ]           [ 1, 1, 1 ]
    L2(37)      1/37   36                                [ "19A" ]               [ 1 ]
    L2(41)      1/41   40                       [ "21A", "7A" ]             [ 1, 1 ]
    L2(43)    23/903   39                       [ "22A", "11A" ]             [ 1, 1 ]
    L2(47)   25/1081   43         [ "24A", "12A", "8A", "6A" ]     [ 1, 1, 1, 1 ]
    L2(49)      1/49   48                                [ "25A" ]               [ 1 ]
    L2(53)      1/53   52                        [ "27A", "9A" ]             [ 1, 1 ]
    L2(59)   31/1711   55         [ "30A", "15A", "10A", "6A" ]     [ 1, 1, 1, 1 ]
    L2(61)      1/61   60                                [ "31A" ]               [ 1 ]
    L2(64)      1/63   62                       [ "65A", "13A" ]             [ 1, 1 ]
    L2(67)   35/2211   63                       [ "34A", "17A" ]             [ 1, 1 ]
    L2(71)   37/2485   67 [ "36A", "18A", "12A", "9A", "6A" ] [ 1, 1, 1, 1, 1 ]
    L2(73)      1/73   72                                [ "37A" ]               [ 1 ]
    L2(79)   41/3081   75         [ "40A", "20A", "10A", "8A" ]     [ 1, 1, 1, 1 ]
```

```
      L2(8)      1/7       6                        [ "3A", "9A" ]           [ 1, 1 ]
      L2(81)     1/81     80                             [ "41A" ]              [ 1 ]
      L2(83)  43/3403    79 [ "42A", "21A", "14A", "7A", "6A" ] [ 1, 1, 1, 1, 1 ]
      L2(89)     1/89     88            [ "45A", "15A", "9A" ]        [ 1, 1, 1 ]
      L2(97)     1/97     96                       [ "49A", "7A" ]         [ 1, 1 ]
      L3(11)   1/6655   6654                      [ "19A", "133A" ]         [ 1, 1 ]
      L3(2)      1/4       3                              [ "7A" ]              [ 1 ]
      L3(3)      1/24     23                             [ "13A" ]              [ 1 ]
      L3(4)      1/5       4                              [ "7A" ]              [ 3 ]
      L3(5)     1/250    249                             [ "31A" ]              [ 1 ]
      L3(7)    1/1372   1371                             [ "19A" ]              [ 1 ]
      L3(8)    1/1792   1791                             [ "73A" ]              [ 1 ]
      L3(9)    1/2880   2879                             [ "91A" ]              [ 1 ]
      L4(3)   53/1053     19                             [ "20A" ]              [ 1 ]
      L5(2)    1/5376   5375                             [ "31A" ]              [ 1 ]
      L6(2) 365/55552   152                      [ "21A", "63A" ]         [ 2, 2 ]
      O8-(2)     1/63     62                             [ "17A" ]              [ 1 ]
      S4(4)     4/15       3                             [ "17A" ]              [ 2 ]
      S4(5)      1/5       4                             [ "13A" ]              [ 1 ]
      S6(3)    1/117     116                             [ "14A" ]              [ 2 ]
      Sz(32)   1/1271   1270                       [ "5A", "25A" ]         [ 1, 1 ]
      Sz(8)     1/91      90                              [ "5A" ]              [ 1 ]
      U3(11)   1/6655   6654                             [ "37A" ]              [ 1 ]
      U3(3)    16/63       3                       [ "6A", "12A" ]         [ 2, 2 ]
      U3(4)    1/160     159                             [ "13A" ]              [ 1 ]
      U3(5)   46/525      11                             [ "10A" ]              [ 2 ]
      U3(7)    1/1372   1371                             [ "43A" ]              [ 1 ]
      U3(8)    1/1792   1791                             [ "19A" ]              [ 1 ]
      U3(9)    1/3600   3599                             [ "73A" ]              [ 1 ]
      U5(2)     1/54      53                             [ "11A" ]              [ 1 ]
      U6(2)     5/21       4                             [ "11A" ]              [ 4 ]
gap> SizeScreen( oldsize );;
gap> First( atleast3, l -> l[1] = "L7(2)" );
[ "L7(2)", 1/4388290560, 4388290559, [ "127A" ], [ 1 ] ]
```

It should be mentioned that [BW75] states the following lower bounds for the uniform spread of the groups $L_2(q)$.

$$
\begin{array}{c|l}
q-2 & \text{if } 4 \le q \text{ is even,} \\
q-1 & \text{if } 11 \le q \equiv 1 \pmod 4, \\
q-4 & \text{if } 11 \le q \equiv -1 \pmod 4.
\end{array}
$$

These bounds appear in the third column of the above table. Furthermore, [BW75] states that the (uniform) spread of alternating groups of even degree at least 8 is exactly 4.

For the sake of completeness, Table II gives an overview of the sets $\mathbb{M}(G, s)$ for those cases in the above list that are needed in [BGK08] but that do not require a further discussion here. The structure of the maximal subgroups and the order of s in the table refer to the matrix groups not to the simple groups. The number of the subgroups has been shown above, the structure follows from [CCN+85].

G	$\mathbb{M}(G,s)$	$\lvert s\rvert$	see [CCN$^+$85]
$\mathrm{SL}(3,4)=3.L_3(4)$	$3\times L_3(2),3\times L_3(2),3\times L_3(2)$	21	p. 23
$\Omega^-(8,2)=O_8^-(2)$	$\Omega^-(4,4).2=L_2(16).2$	17	p. 89
$\mathrm{Sp}(4,4)=S_4(4)$	$\Omega^-(4,4).2=L_2(16).2,\mathrm{Sp}(2,16).2=L_2(16).2$	17	p. 44
$\mathrm{Sp}(6,3)=2.S_6(3)$	$(4\times U_3(3)).2,\mathrm{Sp}(2,17).3=2.L_2(27).3$	28	p. 113
$\mathrm{SU}(3,3)=U_3(3)$	$3_+^{1+2}:8,\mathrm{GU}(2,3)=4.S_4$	6	p. 14
$\mathrm{SU}(3,5)=3.U_3(5)$	$3\times 5_+^{1+2}:8,\mathrm{GU}(2,5)=3\times 2S_5$	30	p. 34
$\mathrm{SU}(5,2)=U_5(2)$	$L_2(11)$	11	p. 73

Table: Table II: Maximal subgroups/>

11.4.5 Automorphism Groups of other Simple Groups – Easy Cases

We deal with automorphic extensions of those simple groups that are listed in Table I and that have been treated successfully in Section 11.4.4.

For the following groups, `ProbGenInfoAlmostSimple` can be used because GAP can compute their primitive permutation characters.

```
                                    Example
  gap> list:= [
  >    [ "A5", "A5.2" ],
  >    [ "A6", "A6.2_1" ],
  >    [ "A6", "A6.2_2" ],
  >    [ "A6", "A6.2_3" ],
  >    [ "A7", "A7.2" ],
  >    [ "A8", "A8.2" ],
  >    [ "A9", "A9.2" ],
  >    [ "A11", "A11.2" ],
  >    [ "L3(2)", "L3(2).2" ],
  >    [ "L3(3)", "L3(3).2" ],
  >    [ "L3(4)", "L3(4).2_1" ],
  >    [ "L3(4)", "L3(4).2_2" ],
  >    [ "L3(4)", "L3(4).2_3" ],
  >    [ "L3(4)", "L3(4).3" ],
  >    [ "S4(4)", "S4(4).2" ],
  >    [ "U3(3)", "U3(3).2" ],
  >    [ "U3(5)", "U3(5).2" ],
  >    [ "U3(5)", "U3(5).3" ],
  >    [ "U4(2)", "U4(2).2" ],
  >    [ "U4(3)", "U4(3).2_1" ],
  >    [ "U4(3)", "U4(3).2_3" ],
  > ];;
  gap> autinfo:= [];;
  gap> fails:= [];;
  gap> for pair in list do
  >      tbl:= CharacterTable( pair[1] );
  >      tblG:= CharacterTable( pair[2] );
  >      info:= ProbGenInfoSimple( tbl );
  >      spos:= List( info[4], x -> Position( AtlasClassNames( tbl ), x ) );
  >      Add( autinfo, ProbGenInfoAlmostSimple( tbl, tblG, spos ) );
  >    od;
  gap> PrintFormattedArray( autinfo );
```

```
         A5.2       0        [ "5AB" ]    [ 1 ]
        A6.2_1    2/3        [ "5AB" ]    [ 2 ]
        A6.2_2    1/6         [ "5A" ]    [ 1 ]
        A6.2_3      0        [ "5AB" ]    [ 1 ]
         A7.2    1/15        [ "7AB" ]    [ 1 ]
         A8.2   13/28       [ "15AB" ]    [ 1 ]
         A9.2     1/4        [ "9AB" ]    [ 1 ]
        A11.2   1/945       [ "11AB" ]    [ 1 ]
       L3(2).2    1/4        [ "7AB" ]    [ 1 ]
       L3(3).2   1/18       [ "13AB" ]    [ 1 ]
     L3(4).2_1   3/10        [ "7AB" ]    [ 3 ]
     L3(4).2_2  11/60         [ "7A" ]    [ 1 ]
     L3(4).2_3   1/12        [ "7AB" ]    [ 1 ]
       L3(4).3   1/64         [ "7A" ]    [ 1 ]
       S4(4).2      0       [ "17AB" ]    [ 2 ]
       U3(3).2    2/7 [ "6A", "12AB" ] [ 2, 2 ]
       U3(5).2   2/21        [ "10A" ]    [ 2 ]
       U3(5).3 46/525        [ "10A" ]    [ 2 ]
       U4(2).2  16/45       [ "12AB" ]    [ 2 ]
     U4(3).2_1 76/135         [ "7A" ]    [ 3 ]
     U4(3).2_3 31/162        [ "7AB" ]    [ 3 ]
```

We see that from this list, the two groups $A_6.2_1 = S_6$ and $U_4(3).2_1$ require further computations (see Sections 11.5.3 and 11.5.24, respectively) because the bound in the second column is larger than $1/2$.

Also $U_4(2)$ is not done by the above, because in [BGK08, Table 4], an element s of order 9 is chosen for the simple group, see Section 11.5.23.

Finally, we deal with automorphic extensions of the groups $L_4(3)$, $O_8^-(2)$, $S_6(3)$, and $U_5(2)$.

For $S = L_4(3)$ and $s \in S$ of order 20, we have $\mathbb{M}(S, s) = \{(4 \times A_6) : 2\}$, the subgroup has index $2\,106$, see [CCN+85, p. 69].

```
_____ Example _____
 gap> t:= CharacterTable( "L4(3)" );;
 gap> prim:= PrimitivePermutationCharacters( t );;
 gap> spos:= Position( AtlasClassNames( t ), "20A" );;
 gap> prim:= Filtered( prim, x -> x[ spos ] <> 0 );
 [ Character( CharacterTable( "L4(3)" ),
   [ 2106, 106, 42, 0, 27, 27, 0, 46, 6, 6, 1, 7, 7, 0, 3, 3, 0, 0, 0,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 1 ] ) ]
```

For the three automorphic extensions of the structure $G = S.2$, we compute the extensions of the permutation character, and the bounds $\sigma'(G, s)$.

```
_____ Example _____
 gap> for name in [ "L4(3).2_1", "L4(3).2_2", "L4(3).2_3" ] do
 >      t2:= CharacterTable( name );
 >      map:= InverseMap( GetFusionMap( t, t2 ) );
 >      torso:= List( prim, pi -> CompositionMaps( pi, map ) );
 >      ext:= Concatenation( List( torso,
 >                             x -> PermChars( t2, rec( torso:= x ) ) ) );
 >      sigma:= ApproxP( ext, Position( OrdersClassRepresentatives( t2 ), 20 ) );
 >      max:= Maximum( sigma{ Difference( PositionsProperty(
```

```
>                                    OrdersClassRepresentatives( t2 ), IsPrimeInt ),
>                                    ClassPositionsOfDerivedSubgroup( t2 ) ) } );
>        Print( name, ":\n", ext, "\n", max, "\n" );
> od;
L4(3).2_1:
[ Character( CharacterTable( "L4(3).2_1" ),
    [ 2106, 106, 42, 0, 27, 0, 46, 6, 6, 1, 7, 0, 3, 0, 0, 1, 1, 0,
      0, 0, 0, 0, 1, 1, 0, 4, 0, 0, 6, 6, 6, 6, 2, 0, 0, 0, 0, 0, 0,
      1, 1, 1, 1 ] ) ]
0
L4(3).2_2:
[ Character( CharacterTable( "L4(3).2_2" ),
    [ 2106, 106, 42, 0, 27, 27, 0, 46, 6, 6, 1, 7, 7, 0, 3, 3, 0, 0,
      0, 1, 1, 1, 0, 0, 0, 1, 306, 306, 42, 6, 10, 10, 0, 0, 15, 15,
      3, 3, 3, 3, 0, 0, 1, 1, 0, 1, 1, 0, 0 ] ) ]
17/117
L4(3).2_3:
[ Character( CharacterTable( "L4(3).2_3" ),
    [ 2106, 106, 42, 0, 27, 0, 46, 6, 6, 1, 7, 0, 3, 0, 0, 1, 1, 0,
      0, 0, 1, 36, 0, 0, 6, 6, 2, 2, 2, 1, 1, 0, 0, 0 ] ) ]
2/117
```

For $S = O_8^-(2)$ and $s \in S$ of order 17, we have $\mathbb{M}(S,s) = \{L_2(16).2\}$, the subgroup extends to $L_2(16).4$ in $S.2$, see [CCN+85, p. 89]. This is a non-split extension, so $\sigma'(S.2,s) = 0$ holds.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> SigmaFromMaxes( CharacterTable( "O8-(2).2" ), "17AB",
>         [ CharacterTable( "L2(16).4" ) ], [ 1 ], "outer" );
0
```

For $S = S_6(3)$ and $s \in S$ irreducible of order 14, we have $\mathbb{M}(S,s) = \{(2 \times U_3(3)).2, L_2(27).3\}$. In $G = S.2$, the subgroups extend to $(4 \times U_3(3)).2$ and $L_2(27).6$, respectively, see [CCN+85, p. 113]. In order to show that $\sigma'(G,s) = 7/3240$ holds, we compute the primitive permutation characters of S (cf. Section 11.4.4) and the unique extensions to G of those which are nonzero on s.

```
――――――――――――――――――――――― Example ―――――――――――――――――――――――
gap> t:= CharacterTable( "S6(3)" );;
gap> t2:= CharacterTable( "S6(3).2" );;
gap> prim:= PrimitivePermutationCharacters( t );;
gap> spos:= Position( AtlasClassNames( t ), "14A" );;
gap> prim:= Filtered( prim, x -> x[ spos ] <> 0 );;
gap> map:= InverseMap( GetFusionMap( t, t2 ) );;
gap> torso:= List( prim, pi -> CompositionMaps( pi, map ) );;
gap> ext:= List( torso, pi -> PermChars( t2, rec( torso:= pi ) ) );
[ [ Character( CharacterTable( "S6(3).2" ),
      [ 155520, 0, 288, 0, 0, 0, 216, 54, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 6, 1, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 144, 288, 0, 0, 0,
        6, 0, 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
        0 ] ) ],
  [ Character( CharacterTable( "S6(3).2" ),
      [ 189540, 1620, 568, 0, 486, 0, 0, 27, 540, 84, 24, 0, 0, 0, 0,
        0, 54, 0, 0, 10, 0, 7, 1, 6, 6, 0, 0, 0, 0, 0, 0, 18, 0, 0,
```

```
               0, 6, 12, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 234, 64,
               30, 8, 0, 3, 90, 6, 0, 4, 10, 6, 0, 2, 1, 0, 0, 0, 0, 0, 0,
               0, 1, 1, 0, 0 ] ) ] ]
gap> spos:= Position( AtlasClassNames( t2 ), "14A" );;
gap> sigma:= ApproxP( Concatenation( ext ), spos );;
gap> Maximum( sigma{ Difference(
>       PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
>       ClassPositionsOfDerivedSubgroup( t2 ) ) } );
7/3240
```

For $S = U_5(2)$ and $s \in S$ of order 11, we have $\mathbb{M}(S, s) = \{L_2(11)\}$, the subgroup extends to $L_2(11).2$ in $S.2$, see [CCN$^+$85, p. 73].

```
                           ──── Example ────
gap> SigmaFromMaxes( CharacterTable( "U5(2).2" ), "11AB",
>         [ CharacterTable( "L2(11).2" ) ], [ 1 ], "outer" );
1/288
```

Here we clean the workspace for the first time. This may save more than 100 megabytes, due to the fact that the caches for tables of marks and character tables are flushed.

```
                           ──── Example ────
gap> CleanWorkspace();
```

11.4.6 $O_8^-(3)$

We show that $S = O_8^-(3) = \Omega^-(8, 3)$ satisfies the following.

(a)   For $s \in S$ of order 41, $\mathbb{M}(S, s)$ consists of one group of the type $L_2(81).2_1 = \Omega^-(4, 9).2$.

(b) $\sigma(S, s) = 1/567$.

The only maximal subgroups of S containing elements of order 41 have the type $L_2(81).2_1$, and there is one class of these subgroups, see [CCN$^+$85, p. 141].

```
                           ──── Example ────
gap> SigmaFromMaxes( CharacterTable( "O8-(3)" ), "41A",
>     [ CharacterTable( "L2(81).2_1" ) ], [ 1 ] );
1/567
```

11.4.7 $O_{10}^+(2)$

We show that $S = O_{10}^+(2) = \Omega^+(10, 2)$ satisfies the following.

(a)   For $s \in S$ of order 45, $\mathbb{M}(S, s)$ consists of one group of the type $(A_5 \times U_4(2)).2 = (\Omega^-(4, 2) \times \Omega^-(6, 2)).2$.

(b) $\sigma(S, s) = 43/4216$.

(c)   For s as in (a), the maximal subgroup in (a) extends to $S_5 \times U_4(2).2$ in $G = \mathrm{Aut}(S) = S.2$, and $\sigma'(G, s) = 23/248$.

The only maximal subgroups of S containing elements of order 45 are one class of groups $H = (A_5 \times U_4(2)) : 2$, see [CCN$^+$85, p. 146]. (Note that none of the groups $S_8(2)$, $O_8^+(2)$, $L_5(2)$, $O_8^-(2)$, and A_8 contains elements of order 45.) H extends to subgroups of the type $H.2 = S_5 \times U_4(2) : 2$ in G, so we can compute $1_H^S = (1_{H.2}^G)_S$.

```
                              ─── Example ───
  gap> ForAny( [ "S8(2)", "O8+(2)", "L5(2)", "O8-(2)", "A8" ],
  >            x -> 45 in OrdersClassRepresentatives( CharacterTable( x ) ) );
  false
  gap> t:= CharacterTable( "O10+(2)" );;
  gap> t2:= CharacterTable( "O10+(2).2" );;
  gap> s2:= CharacterTable( "A5.2" ) * CharacterTable( "U4(2).2" );
  CharacterTable( "A5.2xU4(2).2" )
  gap> pi:= PossiblePermutationCharacters( s2, t2 );;
  gap> spos:= Position( OrdersClassRepresentatives( t2 ), 45 );;
  gap> approx:= ApproxP( pi, spos );;
  gap> Maximum( approx{ ClassPositionsOfDerivedSubgroup( t2 ) } );
  43/4216
```

Statement (c) follows from considering the outer classes of prime element order.

```
                              ─── Example ───
  gap> Maximum( approx{ Difference(
  >        PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
  >        ClassPositionsOfDerivedSubgroup( t2 ) ) } );
  23/248
```

Alternatively, we can use `SigmaFromMaxes`.

```
                              ─── Example ───
  gap> SigmaFromMaxes( t2, "45AB", [ s2 ], [ 1 ], "outer" );
  23/248
```

11.4.8 $O_{10}^-(2)$

We show that $S = O_{10}^-(2) = \Omega^-(10,2)$ satisfies the following.

(a) For $s \in S$ of order 33, $\mathbb{M}(S,s)$ consists of one group of the type $3 \times U_5(2) = \mathrm{GU}(5,2)$.

(b) $\sigma(S,s) = 1/119$.

(c) For s as in (a), the maximal subgroup in (a) extends to $(3 \times U_5(2)).2$ in G, and $\sigma'(G,s) = 1/595$.

The only maximal subgroups of S containing elements of order 11 have the types A_{12} and $3 \times U_5(2)$, see [CCN$^+$85, p. 147]. So $3 \times U_5(2)$ is the unique class of subgroups containing elements of order 33. This shows statement (a), and statement (b) follows using `SigmaFromMaxes`.

```
                              ─── Example ───
  gap> SigmaFromMaxes( CharacterTable( "O10-(2)" ), "33A",
  >      [ CharacterTable( "Cyclic", 3 ) * CharacterTable( "U5(2)" ) ], [ 1 ] );
  1/119
```

The structure of the maximal subgroup of *G* follows from [CCN+85, p. 147]. We create its character table with a generic construction that is based on the fact that the outer automorphism acts nontrivially on the two direct factors; this determines the character table uniquely. (See [Brec] for details.)

```
────────────────── Example ──────────────────
  gap> tblG:= CharacterTable( "U5(2)" );;
  gap> tblMG:= CharacterTable( "Cyclic", 3 ) * tblG;;
  gap> tblGA:= CharacterTable( "U5(2).2" );;
  gap> acts:= PossibleActionsForTypeMGA( tblMG, tblG, tblGA );;
  gap> poss:= Concatenation( List( acts, pi ->
  >              PossibleCharacterTablesOfTypeMGA( tblMG, tblG, tblGA, pi,
  >                 "(3xU5(2)).2" ) ) );
  [ rec(
        MGfusMGA := [ 1, 2, 3, 4, 4, 5, 5, 6, 7, 8, 9, 10, 11, 12, 12,
            13, 13, 14, 14, 15, 15, 16, 17, 17, 18, 18, 19, 20, 21, 21,
            22, 22, 23, 23, 24, 24, 25, 25, 26, 27, 27, 28, 28, 29, 29,
            30, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
            44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58,
            59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73,
            74, 75, 76, 77, 31, 32, 33, 35, 34, 37, 36, 38, 39, 40, 41,
            42, 43, 45, 44, 47, 46, 49, 48, 51, 50, 52, 54, 53, 56, 55,
            57, 58, 60, 59, 62, 61, 64, 63, 66, 65, 68, 67, 69, 71, 70,
            73, 72, 75, 74, 77, 76 ],
      table := CharacterTable( "(3xU5(2)).2" ) ) ]
```

Now statement (c) follows using `SigmaFromMaxes`.

```
────────────────── Example ──────────────────
  gap> SigmaFromMaxes( CharacterTable( "O10-(2).2" ), "33AB",
  >          [ poss[1].table ], [ 1 ], "outer" );
  1/595
```

11.4.9 $O_{12}^+(2)$

We show that $S = O_{12}^+(2) = \Omega^+(12,2)$ satisfies the following.

(a) For $s \in S$ of the type $4^- \perp 8^-$ (i. e., *s* decomposes the natural 12-dimensional module for $\mathrm{GO}_{12}^+(2) = S.2$ into an orthogonal sum of two irreducible modules of the dimensions 4 and 8, respectively) and of order 85, $\mathbb{M}(S,s)$ consists of one group of the type $G_8 = (\Omega^-(4,2) \times \Omega^-(8,2)).2$ and two groups of the type $L_4(4).2^2 = \Omega^+(6,4).2^2$ that are conjugate in $G = \mathrm{Aut}(S) = S.2 = \mathrm{SO}^+(12,2)$ but *not* conjugate in *S*.

(b) $\sigma(S,s) = 7675/1\,031\,184$.

(c) $\sigma'(G,s) = 73/1\,008$.

The element *s* is a ppd$(12,2;8)$-element in the sense of [GPPS99], so the maximal subgroups of *S* that contain *s* are among the nine cases (2.1)–(2.9) listed in this paper; in the notation of this paper, we have $q = 2$, $d = 12$, $e = 8$, and $r = 17$. Case (2.1) does not occur for orthogonal groups and $q = 2$, according to [KL90]; case (2.2) contributes a unique maximal subgroup, the stabilizer G_8 of the orthogonal decomposition; the cases (2.3), (2.4) (a), (2.5), and (2.6) (a) do not occur because $r \neq e + 1$

in our situation; case (2.4) (b) describes extension field type subgroups that are contained in $\Gamma L(6,4)$, which yields the candidates $GU(6,2).2 \cong 3.U_6(2).S_3$ –but $3.U_6(2).3$ does not contain elements of order 85– and $\Omega^+(6,4).2^2 \cong L_4(4).2^2$ (two classes by [KL90, Prop. 4.3.14]); the cases (2.6) (b)–(c) and (2.8) do not occur because they require $d \leq 8$; case (2.7) does not occur because [GPPS99, Table 5] contains no entry for $r = 2e + 1 = 17$; finally, case (2.9) does not occur because it requires $e \in \{d - 1, d\}$ in the case $r = 2e + 1$.

So we need the permutation characters of the actions on the cosets of $L_4(4).2^2$ (two classes) and G_8. According to [KL90, Prop. 4.1.6], G_8 has the structure $(\Omega^-(4,2) \times \Omega^-(8,2)).2$.

Newer versions of the GAP Character Table Library contain the character table of S, but it is still easier to work with the table of G, which was already available at the times when the first version of these examples was created.

The two classes of $L_4(4).2^2$ type subgroups in S are fused in G. This can be seen from the fact that inducing the trivial character of a subgroup $H_1 = L_4(4).2^2$ of S to G yields a character ψ whose values are not all even; note that if H_1 would extend in G to a subgroup of twice the size of H_1 then ψ would be induced from a degree two character of this subgroup whose values are all even, and induction preserves this property.

```
———————————— Example ————————————
gap> t:= CharacterTable( "O12+(2).2" );;
gap> h1:= CharacterTable( "L4(4).2^2" );;
gap> psi:= PossiblePermutationCharacters( h1, t );;
gap> Length( psi );
1
gap> ForAny( psi[1], IsOddInt );
true
```

The fixed element s of order 85 is contained in a member of each of the two conjugacy classes of the type $L_4(4).2^2$ in S, since S contains only one class of subgroups of the order 85; note that the order of the Sylow 17 centralizer (in both S and G) is not divisible by 25.

```
———————————— Example ————————————
gap> SizesCentralizers( t ){ PositionsProperty(
>          OrdersClassRepresentatives( t ), x -> x = 17 ) } / 25;
[ 408/5, 408/5 ]
```

This implies that the restriction of ψ to S is the sum $\psi_S = \pi_1 + \pi_2$, say, of the first two interesting permutation characters of S.

The subgroup G_8 of S extends to a group of the structure $H_2 = \Omega^-(4,2).2 \times \Omega^-(8,2).2$ in G, inducing the trivial characters of H_2 to G yields a permutation character φ of G whose restriction to S is the third permutation character $\varphi_S = \pi_3$, say.

```
———————————— Example ————————————
gap> h2:= CharacterTable( "S5" ) * CharacterTable( "O8-(2).2" );;
gap> phi:= PossiblePermutationCharacters( h2, t );;
gap> Length( phi );
1
```

We have $\pi_1(1) = \pi_2(1)$ and $\pi_1(s) = \pi_2(s)$, the latter again because S contains only one class of subgroups of order 85.

Now statement (a) follows from the fact that $\pi_i(s) = 1$ holds for $1 \leq i \leq 3$.

```
─────────────────────────── Example ───────────────────────────
gap> prim:= Concatenation( psi, phi );;
gap> spos:= Position( OrdersClassRepresentatives( t ), 85 );
213
gap> List( prim, x -> x[ spos ] );
[ 2, 1 ]
```

For statement (b), we compute $\sigma(S, s)$. Note that we have to consider only classes inside $S = G'$, and that

$$\sigma(g, s) = \sum_{i=1}^{3} \frac{\pi_i(s) \cdot \pi_i(g)}{\pi_i(1)} = \frac{\psi(s) \cdot \psi(g)}{\psi(1)} + \frac{\varphi(s) \cdot \varphi(g)}{\varphi(1)}$$

holds for $g \in S^{\times}$, so the characters ψ and φ are sufficient.

```
─────────────────────────── Example ───────────────────────────
gap> approx:= ApproxP( prim, spos );;
gap> Maximum( approx{ ClassPositionsOfDerivedSubgroup( t ) } );
7675/1031184
```

Statement (c) follows from considering the outer involution classes. Note that by [BGK08, Remark after Proposition 5.14], only the subgroup H_2 need to be considered, no novelties appear.

```
─────────────────────────── Example ───────────────────────────
gap> Maximum( approx{ Difference(
>       PositionsProperty( OrdersClassRepresentatives( t ), IsPrimeInt ),
>       ClassPositionsOfDerivedSubgroup( t ) ) } );
73/1008
```

11.4.10 $O_{12}^{-}(2)$

We show that $S = O_{12}^{-}(2) = \Omega^{-}(12, 2)$ satisfies the following.

(a) For $s \in S$ irreducible of order $2^6 + 1 = 65$, $\mathbb{M}(S, s)$ consists of two groups of the types $U_4(4).2 = \Omega^{-}(6, 4).2$ and $L_2(64).3 = \Omega^{-}(4, 8).3$, respectively.

(b) $\sigma(S, s) = 1/1\,023$.

(c) $\sigma'(\text{Aut}(S), s) = 1/347\,820$.

By [Ber00], $\mathbb{M}(S, s)$ consists of extension field subgroups, which have the structures $U_4(4).2$ and $L_2(64).3$, respectively, and by [KL90, Prop. 4.3.16], there is just one class of each of these types.

Newer versions of the GAP Character Table Library contain the character table of S, but using this table for the computations is not easier than using the table of $G = \text{Aut}(S) = O_{12}^{-}(2).2$, which was already available at the times when the first version of these examples was created. So we compute the permutation characters π_1, π_2 of the extensions of the groups in $\mathbb{M}(S, s)$ to G –these maximal subgroups have the structures $U_4(4).4$ and $L_2(64).6$, respectively– and compute the fixed point ratios of the restrictions to S.

```
─────────────────────────── Example ───────────────────────────
gap> t:= CharacterTable( "O12-(2).2" );;
gap> s1:= CharacterTable( "U4(4).4" );;
gap> pi1:= PossiblePermutationCharacters( s1, t );;
```

```
gap> s2:= CharacterTable( "L2(64).6" );;
gap> pi2:= PossiblePermutationCharacters( s2, t );;
gap> prim:= Concatenation( pi1, pi2 );;  Length( prim );
2
```

Now statement (a) follows from the fact that $\pi_1(s) = \pi_2(s) = 1$ holds.

```
———————————————— Example ————————————————
gap> spos:= Position( OrdersClassRepresentatives( t ), 65 );;
gap> List( prim, x -> x[ spos ] );
[ 1, 1 ]
```

For statement (b), we compute $\sigma(S,s)$; note that we have to consider only classes inside $S = G'$.

```
———————————————— Example ————————————————
gap> approx:= ApproxP( prim, spos );;
gap> Maximum( approx{ ClassPositionsOfDerivedSubgroup( t ) } );
1/1023
```

Statement (c) follows from the values on the outer involution classes.

```
———————————————— Example ————————————————
gap> Maximum( approx{ Difference(
>       PositionsProperty( OrdersClassRepresentatives( t ), IsPrimeInt ),
>       ClassPositionsOfDerivedSubgroup( t ) ) } );
1/347820
```

11.4.11 $S_6(4)$

We show that $S = S_6(4) = \mathrm{Sp}(6,4)$ satisfies the following.

(a)   For $s \in S$ irreducible of order 65, $\mathbb{M}(S,s)$ consists of two groups of the types $U_4(4).2 = \Omega^-(6,4).2$ and $L_2(64).3 = \mathrm{Sp}(2,64).3$, respectively.

(b) $\sigma(S,s) = 16/63$.

(c) $\sigma'(\mathrm{Aut}(S),s) = 0$.

By [Ber00], the element s is contained in maximal subgroups of the given types, and by [KL90, Prop. 4.3.10, 4.8.6], there is exactly one class of these subgroups.

The character tables of these two subgroups are currently not contained in the GAP Character Table Library. We compute the permutation character induced from the first subgroup as the unique character of the right degree that is combinatorially possible (cf. [BP98]).

```
———————————————— Example ————————————————
gap> t:= CharacterTable( "S6(4)" );;
gap> degree:= Size( t ) / ( 2 * Size( CharacterTable( "U4(4)" ) ) );;
gap> pi1:= PermChars( t, rec( torso:= [ degree ] ) );;
gap> Length( pi1 );
1
```

The index of the second subgroup is too large for this simpleminded approach; therefore, we first restrict the set of possible irreducible constituents of the permutation character to those of 1_H^G, where H is the derived subgroup of $L_2(64).3$, for which the character table is available.

```
─────────── Example ───────────
gap> CharacterTable( "L2(64).3" );  CharacterTable( "U4(4).2" );
fail
fail
gap> s:= CharacterTable( "L2(64)" );;
gap> subpi:= PossiblePermutationCharacters( s, t );;
gap> Length( subpi );
1
gap> scp:= MatScalarProducts( t, Irr( t ), subpi );;
gap> nonzero:= PositionsProperty( scp[1], x -> x <> 0 );
[ 1, 11, 13, 14, 17, 18, 32, 33, 56, 58, 59, 73, 74, 77, 78, 79, 80,
  93, 95, 96, 103, 116, 117, 119, 120 ]
gap> const:= RationalizedMat( Irr( t ){ nonzero } );;
gap> degree:= Size( t ) / ( 3 * Size( s ) );
5222400
gap> pi2:= PermChars( t, rec( torso:= [ degree ], chars:= const ) );;
gap> Length( pi2 );
1
gap> prim:= Concatenation( pi1, pi2 );;
```

Now statement (a) follows from the fact that $\pi_1(s) = \pi_2(s) = 1$ holds.

```
─────────── Example ───────────
gap> spos:= Position( OrdersClassRepresentatives( t ), 65 );;
gap> List( prim, x -> x[ spos ] );
[ 1, 1 ]
```

For statement (b), we compute $\sigma(G, s)$.

```
─────────── Example ───────────
gap> Maximum( ApproxP( prim, spos ) );
16/63
```

In order to prove statement (c), we have to consider only the extensions of the above permutation characters of S to $\mathrm{Aut}(S) \cong S.2$ (cf. [BGK08, Section 2.2]).

```
─────────── Example ───────────
gap> t2:= CharacterTable( "S6(4).2" );;
gap> tfust2:= GetFusionMap( t, t2 );;
gap> cand:= List( prim, x -> CompositionMaps( x, InverseMap( tfust2 ) ) );;
gap> ext:= List( cand, pi -> PermChars( t2, rec( torso:= pi ) ) );
[ [ Character( CharacterTable( "S6(4).2" ),
      [ 2016, 512, 96, 128, 32, 120, 0, 6, 16, 40, 24, 0, 8, 136, 1,
        6, 6, 1, 32, 0, 8, 6, 2, 0, 2, 0, 0, 4, 0, 16, 32, 1, 8, 2,
        6, 2, 1, 2, 4, 0, 0, 1, 6, 0, 1, 10, 0, 1, 1, 0, 10, 10, 4,
        0, 1, 0, 2, 0, 2, 1, 2, 2, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0,
        0, 0, 0, 32, 0, 0, 8, 0, 0, 0, 0, 8, 8, 0, 0, 0, 0, 8, 0,
        0, 0, 2, 2, 0, 2, 2, 0, 2, 2, 2, 0, 0 ] ) ],
  [ Character( CharacterTable( "S6(4).2" ),
      [ 5222400, 0, 0, 0, 1280, 0, 960, 120, 0, 0, 0, 0, 0, 0, 1600,
```

```
              0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 1, 0, 0, 15, 0, 0, 0, 0, 0,
              0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0,
              0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0,
              0, 0, 960, 0, 0, 0, 16, 0, 24, 12, 0, 0, 0, 0, 0, 0, 0, 0,
              0, 0, 0, 0, 4, 1, 0, 0, 3, 0, 0, 0, 0, 0 ] ) ] ]
gap> spos2:= Position( OrdersClassRepresentatives( t2 ), 65 );;
gap> sigma:= ApproxP( Concatenation( ext ), spos2 );;
gap> Maximum( approx{ Difference(
>         PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
>         ClassPositionsOfDerivedSubgroup( t2 ) ) } );
0
```

For the simple group, we can *alternatively* consider a reducible element $s : 2 \perp 4$ of order 85, which is a multiple of the primitive prime divisor $r = 17$ of $4^4 - 1$. So we have $e = 4$, $d = 6$, and $q = 4$, in the terminology of [GPPS99]. Then $\mathbb{M}(S, s)$ consists of two groups, of the types $\Omega^+(6,4).2 \cong L_4(4).2_2$ and $\mathrm{Sp}(2,4) \times \mathrm{Sp}(4,4)$. This can be shown by checking [GPPS99, Ex. 2.1–2.9]. Ex. 2.1 yields the candidates $\Omega^\pm(6,4).2$, but only $\Omega^+(6,4).2$ contains elements of order 85. Ex. 2.2 yields the stabilizer of a two-dimensional subspace, which has the structure $\mathrm{Sp}(2,4) \times \mathrm{Sp}(4,4)$, by [KL90]. All other cases except Ex. 2.4 (b) are excluded by the fact that $r = 4e + 1$, and Ex. 2.4 (b) does not apply because $d / \gcd(d,e)$ is odd.

```
                          ── Example ──
gap> SigmaFromMaxes( CharacterTable( "S6(4)" ), "85A",
>     [ CharacterTable( "L4(4).2_2" ),
>       CharacterTable( "A5" ) * CharacterTable( "S4(4)" ) ], [ 1, 1 ] );
142/455
```

This bound is not as good as the one obtained from the irreducible element of order 65 used above.

```
                          ── Example ──
gap> 16/63 < 142/455;
true
```

11.4.12 $* S_6(5)$

We show that $S = S_6(5) = \mathrm{PSp}(6,5)$ satisfies the following.

(a) For $s \in S$ of the type $2 \perp 4$ (i. e., the preimage of s in $\mathrm{Sp}(6,5) = 2.G$ decomposes the natural 6-dimensional module for $\mathrm{Sp}(6,5)$ into an orthogonal sum of two irreducible modules of the dimensions 2 and 4, respectively) and of order 78, $\mathbb{M}(S, s)$ consists of one group of the type $G_2 = 2.(\mathrm{PSp}(2,5) \times \mathrm{PSp}(4,5))$.

(b) $\sigma(S, s) = 9/217$.

The order of s is a multiple of the primitive prime divisor $r = 13$ of $5^4 - 1$, so we have $e = 4$, $d = 6$, and $q = 5$, in the terminology of [GPPS99]. We check [GPPS99, Ex. 2.1–2.9]. Ex. 2.1 does not apply because the classes C_5 and C_8 are empty by [KL90, Table 3.5.C], Ex. 2.2 yields exactly the stabilizer G_2 of a 2-dimensional subspace, Ex. 2.4 (b) does not apply because $d / \gcd(d,e)$ is odd, and all other cases are excluded by the fact that $r = 3e + 1$.

The group G_2 has the structure $2.(\mathrm{PSp}(2,5) \times \mathrm{PSp}(4,5))$, which is a central product of $\mathrm{Sp}(2,5) \cong 2.A_5$ and $\mathrm{Sp}(4,5) = 2.S_4(5)$ (see [KL90, Prop. 4.1.3]). The character table of G_2 can be derived from

that of the direct product of $2.A_5$ and $2.S_4(5)$, by factoring out the diagonal central subgroup of order two.

```
                      ─── Example ───
  gap> t:= CharacterTable( "S6(5)" );;
  gap> s1:= CharacterTable( "2.A5" );;
  gap> s2:= CharacterTable( "2.S4(5)" );;
  gap> dp:= s1 * s2;
  CharacterTable( "2.A5x2.S4(5)" )
  gap> c:= Difference( ClassPositionsOfCentre( dp ), Union(
  >                       GetFusionMap( s1, dp ), GetFusionMap( s2, dp ) ) );
  [ 62 ]
  gap> s:= dp / c;
  CharacterTable( "2.A5x2.S4(5)/[ 1, 62 ]" )
```

Now we compute $\sigma(S, s)$.

```
                      ─── Example ───
  gap> SigmaFromMaxes( t, "78A", [ s ], [ 1 ] );
  9/217
```

11.4.13 $S_8(3)$

We show that $S = S_8(3) = \mathrm{PSp}(8, 3)$ satisfies the following.

(a) For $s \in S$ irreducible of order 41, $\mathbb{M}(S, s)$ consists of one group M of the type $S_4(9).2 = \mathrm{PSp}(4, 9).2$.

(b) $\sigma(S, s) = 1/546$.

(c) The preimage of s in the matrix group $2.S_8(3) = \mathrm{Sp}(8, 3)$ can be chosen of order 82, and the preimage of M is $2.S_4(9).2 = \mathrm{Sp}(4, 9).2$.

By [Ber00], the only maximal subgroups of S that contain irreducible elements of order $(3^4 + 1)/2 = 41$ are of extension field type, and by [KL90, Prop. 4.3.10], these groups have the structure $S_4(9).2$ and there is exactly one class of these groups.

The group $U = S_4(9)$ has three nontrivial outer automorphisms, the character table of the subgroup $U.2$ in question has the identifier `"S4(9).2_1"`, which follows from the fact that the extensions of U by the other two outer automorphisms do not admit a class fusion into S.

```
                      ─── Example ───
  gap> t:= CharacterTable( "S8(3)" );;
  gap> pi:= List( [ "S4(9).2_1", "S4(9).2_2", "S4(9).2_3" ],
  >               name -> PossiblePermutationCharacters(
  >                       CharacterTable( name ), t ) );;
  gap> List( pi, Length );
  [ 1, 0, 0 ]
```

Now statement (a) follows from the fact that $(1_{U.2})^S(s) = 1$ holds.

```
                      ─── Example ───
  gap> spos:= Position( OrdersClassRepresentatives( t ), 41 );;
  gap> pi[1][1][ spos ];
  1
```

Now we compute $\sigma(S, s)$ in order to show statement (b).

```
———— Example ————
gap> Maximum( ApproxP( pi[1], spos ) );
1/546
```

Statement (c) is clear from the description of extension field type subgroups in [KL90].

11.4.14 $U_4(4)$

We show that $S = U_4(4) = \mathrm{SU}(4,4)$ satisfies the following.

(a)   For $s \in S$ of the type $1 \perp 3$ (i. e., s decomposes the natural 4-dimensional module for $\mathrm{SU}(4,4)$ into an orthogonal sum of two irreducible modules of the dimensions 1 and 3, respectively) and of order $4^3 + 1 = 65$, $\mathbb{M}(S, s)$ consists of one group of the type $G_1 = 5 \times U_3(4) = \mathrm{GU}(3,4)$.

(b) $\sigma(S, s) = 209/3\,264$.

By [MSW94], the only maximal subgroups of S that contain s are one class of stabilizers $H \cong 5 \times U_3(4)$ of this decomposition, and clearly there is only one such group containing s.

Note that H has index $3\,264$ in S, since S has two orbits on the 1-dimensional subspaces, of lengths $1\,105$ and $3\,264$, respectively, and elements of order $13 = 65/5$ lie in the stabilizers of points in the latter orbit.

```
———— Example ————
gap> g:= SU(4,4);;
gap> orbs:= OrbitsDomain( g, NormedRowVectors( GF(16)^4 ), OnLines );;
gap> orblen:= List( orbs, Length );
[ 1105, 3264 ]
gap> List( orblen, x -> x mod 13 );
[ 0, 1 ]
```

We compute the permutation character $1_{G_1}^S$; there is exactly one combinatorially possible permutation character of degree $3\,264$ (cf. [BP98]).

```
———— Example ————
gap> t:= CharacterTable( "U4(4)" );;
gap> pi:= PermChars( t, rec( torso:= [ orblen[2] ] ) );;
gap> Length( pi );
1
```

Now we compute $\sigma(S, s)$.

```
———— Example ————
gap> spos:= Position( OrdersClassRepresentatives( t ), 65 );;
gap> Maximum( ApproxP( pi, spos ) );
209/3264
```

11.4.15 $U_6(2)$

We show that $S = U_6(2) = \mathrm{PSU}(6,2)$ satisfies the following.

(a)   For $s \in S$ of order 11, $\mathbb{M}(S,s)$ consists of one group of the type $U_5(2) = \mathrm{SU}(5,2)$ and three groups of the type M_{22}.

(b) $\sigma(S,s) = 5/21$.

(c)   The preimage of s in the matrix group $\mathrm{SU}(6,2) = 3.U_6(2)$ can be chosen of order 33, and the preimages of the groups in $\mathbb{M}(S,s)$ have the structures $3 \times U_5(2) \cong \mathrm{GU}(5,2)$ and $3.M_{22}$, respectively.

(d)   With s as in (a), the automorphic extensions $S.2$, $S.3$ of S satisfy $\sigma'(S.2,s) = 5/96$ and $\sigma'(S.3,s) = 59/224$.

   According to the list of maximal subgroups of S in [CCN$^+$85, p. 115], s is contained exactly in maximal subgroups of the types $U_5(2)$ (one class) and M_{22} (three classes).

   The permutation character of the action on the cosets of $U_5(2)$ type subgroups is uniquely determined by the character tables. We get three possibilities for the permutation character on the cosets of M_{22} type subgroups; they correspond to the three classes of such subgroups, because each of these classes contains elements in exactly one of the conjugacy classes 4C, 4D, and 4E of elements in S, and these classes are fused under the outer automorphism of S of order three.

```
————————————————————— Example —————————————————————
  gap> t:= CharacterTable( "U6(2)" );;
  gap> s1:= CharacterTable( "U5(2)" );;
  gap> pi1:= PossiblePermutationCharacters( s1, t );;
  gap> Length( pi1 );
  1
  gap> s2:= CharacterTable( "M22" );;
  gap> pi2:= PossiblePermutationCharacters( s2, t );
  [ Character( CharacterTable( "U6(2)" ),
    [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, 0, 48, 0, 16, 6, 0, 0, 0, 0,
        0, 0, 6, 0, 2, 0, 0, 0, 4, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0 ] ), Character( CharacterTable( "U6(2)" ),
    [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 0, 48, 0, 0, 16, 6, 0, 0, 0, 0,
        0, 0, 6, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0 ] ), Character( CharacterTable( "U6(2)" ),
    [ 20736, 0, 384, 0, 0, 0, 54, 0, 0, 48, 0, 0, 0, 16, 6, 0, 0, 0, 0,
        0, 0, 6, 0, 2, 0, 4, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0 ] ) ]
  gap> imgs:= Set( pi2, x -> Position( x, 48 ) );
  [ 10, 11, 12 ]
  gap> AtlasClassNames( t ){ imgs };
  [ "4C", "4D", "4E" ]
  gap> GetFusionMap( t, CharacterTable( "U6(2).3" ) ){ imgs };
  [ 10, 10, 10 ]
  gap> prim:= Concatenation( pi1, pi2 );;
```

   Now statement (a) follows from the fact that the permutation characters have the value 1 on s.

```
———————————————— Example ————————————————
gap> spos:= Position( OrdersClassRepresentatives( t ), 11 );;
gap> List( prim, x -> x[ spos ] );
[ 1, 1, 1, 1 ]
```

For statement (b), we compute $\sigma(S, s)$.

```
———————————————— Example ————————————————
gap> Maximum( ApproxP( prim, spos ) );
5/21
```

Statement (c) follows from [CCN$^+$85], plus the information that $3.U_6(2)$ does not contain groups of the structure $3 \times M_{22}$.

```
———————————————— Example ————————————————
gap> PossibleClassFusions(
>         CharacterTable( "Cyclic", 3 ) * CharacterTable( "M22" ),
>         CharacterTable( "3.U6(2)" ) );
[  ]
```

For statement (d), we need that the relevant maximal subgroups of $S.2$ are $U_5(2).2$ and one subgroup $M_{22}.2$, and that the relevant maximal subgroup of $S.3$ is $U_5(2) \times 3$, see [CCN$^+$85, p. 115].

```
———————————————— Example ————————————————
gap> SigmaFromMaxes( CharacterTable( "U6(2).2" ), "11AB",
>         [ CharacterTable( "U5(2).2" ), CharacterTable( "M22.2" ) ],
>         [ 1, 1 ], "outer" );
5/96
gap> SigmaFromMaxes( CharacterTable( "U6(2).3" ), "11A",
>         [ CharacterTable( "U5(2)" ) * CharacterTable( "Cyclic", 3 ) ],
>         [ 1 ], "outer" );
59/224
```

11.5 Computations using Groups

Before we start the computations using groups, we clean the workspace.

```
———————————————— Example ————————————————
gap> CleanWorkspace();
```

11.5.1 A_{2m+1}, $2 \leq m \leq 11$

For alternating groups of odd degree $n = 2m + 1$, we choose s to be an n-cycle. The interesting cases in [BGK08, Proposition 6.7] are $5 \leq n \leq 23$.

In each case, we compute representatives of the maximal subgroups of A_n, consider only those that contain an n-cycle, and then compute the permutation characters. Additionally, we show also the names that are used for the subgroups in the GAP Library of Transitive Groups, see [Hul05] and the documentation of this library in the GAP Reference Manual.

```
                              ____ Example ____
gap> PrimitivesInfoForOddDegreeAlternatingGroup:= function( n )
>       local G, max, cycle, spos, prim, nonz;
>
>       G:= AlternatingGroup( n );
>
>       # Compute representatives of the classes of maximal subgroups.
>       max:= MaximalSubgroupClassReps( G );
>
>       # Omit subgroups that cannot contain an 'n'-cycle.
>       max:= Filtered( max, m -> IsTransitive( m, [ 1 .. n ] ) );
>
>       # Compute the permutation characters.
>       cycle:= [];
>       cycle[ n-1 ]:= 1;
>       spos:= PositionProperty( ConjugacyClasses( CharacterTable( G ) ),
>                 c -> CycleStructurePerm( Representative( c ) ) = cycle );
>       prim:= List( max, m -> TrivialCharacter( m )^G );
>       nonz:= PositionsProperty( prim, x -> x[ spos ] <> 0 );
>
>       # Compute the subgroup names and the multiplicities.
>       return rec( spos := spos,
>                   prim := prim{ nonz },
>                   grps := List( max{ nonz },
>                               m -> TransitiveGroup( n,
>                                       TransitiveIdentification( m ) ) ),
>                   mult := List( prim{ nonz }, x -> x[ spos ] ) );
> end;;
```

The sets $\tilde{\mathbb{M}}(s)$ and the values $\sigma(A_n, s)$ are as follows. For each degree in question, the first list shows names for representatives of the conjugacy classes of maximal subgroups containing a fixed n-cycle, and the second list shows the number of conjugates in each class.

```
                              ____ Example ____
gap> for n in [ 5, 7 .. 23 ] do
>       prim:= PrimitivesInfoForOddDegreeAlternatingGroup( n );
>       bound:= Maximum( ApproxP( prim.prim, prim.spos ) );
>       Print( n, ": ", prim.grps, ", ", prim.mult, ", ", bound, "\n" );
> od;
5: [ D(5) = 5:2 ], [ 1 ], 1/3
7: [ L(7) = L(3,2), L(7) = L(3,2) ], [ 1, 1 ], 2/5
9: [ 1/2[S(3)^3]S(3), L(9):3=P|L(2,8) ], [ 1, 3 ], 9/35
11: [ M(11), M(11) ], [ 1, 1 ], 2/105
13: [ F_78(13)=13:6, L(13)=PSL(3,3), L(13)=PSL(3,3) ], [ 1, 2, 2 ], 4/
1155
15: [ 1/2[S(3)^5]S(5), 1/2[S(5)^3]S(3), L(15)=A_8(15)=PSL(4,2),
  L(15)=A_8(15)=PSL(4,2) ], [ 1, 1, 1, 1 ], 29/273
17: [ L(17):4=PYL(2,16), L(17):4=PYL(2,16) ], [ 1, 1 ], 2/135135
19: [ F_171(19)=19:9 ], [ 1 ], 1/6098892800
21: [ t21n150, t21n161, t21n91 ], [ 1, 1, 2 ], 29/285
23: [ M(23), M(23) ], [ 1, 1 ], 2/130945815
```

In the above output, a subgroup printed as 1/2[S(n_1)^n_2]S(n_2), 1/2[S(n_1)^n_2]S(n_2), where

$n = n_1 n_2$ holds, denotes the intersection of A_n with the wreath product $S_{n_1} \wr S_{n_2} \leq S_n$. (Note that the Atlas denotes the subgroup $1/2[\text{S}(3)^\wedge 3]\text{S}(3)$ of A_9 as $3^3 : S_4$.) The groups printed as $\text{P|L}(2,8)$ and $\text{PYL}(2,16)$ denote $P\Gamma L(2,8)$ and $P\Gamma L(2,16)$, respectively. And the three subgroups of A_{21} have the structures $(S_3 \wr S_7) \cap A_{21}$, $(S_7 \wr S_3) \cap A_{21}$, and $\text{PGL}(3,4)$, respectively.

Note that A_9 contains two conjugacy classes of maximal subgroups of the type $P\Gamma L(2,8) \cong L_2(8) : 3$, and that each 9-cycle in A_9 is contained in exactly three *conjugate* subgroups of this type. For $n \in \{13, 15, 17\}$, A_n contains two conjugacy classes of isomorphic maximal subgroups of linear type, and each n-cycle is contained in subgroups from each class. Finally, A_{21} contains only one class of maximal subgroups of linear type.

For the two groups A_5 and A_7, the values computed above are not sufficient. See Section 11.5.2 and 11.5.4 for a further treatment.

The above computations look like a brute-force approach, but note that the computation of the maximal subgroups of alternating and symmetric groups in GAP uses the classification of these subgroups, and also the conjugacy classes of elements in alternating and symmetric groups can be computed cheaply.

Alternative (character-theoretic) computations for $n \in \{5, 7, 9, 11, 13\}$ were shown in Section 11.4.4. (A hand calculation for the case $n = 19$ can be found in [BW75].)

11.5.2 A_5

We show that $S = A_5$ satisfies the following.

(a) $\sigma(S) = 1/3$, and this value is attained exactly for $\sigma(S, s)$ with s of order 5.

(b) For $s \in S$ of order 5, $\mathbb{M}(S, s)$ consists of one group of the type D_{10}.

(c) $P(S) = 1/3$, and this value is attained exactly for $P(S, s)$ with s of order 5.

(d) Each element in S together with one of $(1,2)(3,4)$, $(1,3)(2,4)$, $(1,4)(2,3)$ generates a proper subgroup of S.

(e) Both the spread and the uniform spread of S is exactly two (see [BW75]), with s of order 5.

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

```
———————————————— Example ————————————————
gap> t:= CharacterTable( "A5" );;
gap> ProbGenInfoSimple( t );
[ "A5", 1/3, 2, [ "5A" ], [ 1 ] ]
```

Statement (b) can be read off from the primitive permutation characters, and the fact that the unique class of maximal subgroups that contain elements of order 5 consists of groups of the structure D_{10}, see [CCN+85, p. 2].

```
———————————————— Example ————————————————
gap> OrdersClassRepresentatives( t );
[ 1, 2, 3, 5, 5 ]
gap> PrimitivePermutationCharacters( t );
[ Character( CharacterTable( "A5" ), [ 5, 1, 2, 0, 0 ] ),
  Character( CharacterTable( "A5" ), [ 6, 2, 0, 1, 1 ] ),
  Character( CharacterTable( "A5" ), [ 10, 2, 1, 0, 0 ] ) ]
```

For statement (c), we compute that for all nonidentity elements $s \in S$ and involutions $g \in S$, $P(g,s) \geq 1/3$ holds, with equality if and only if s has order 5. We actually compute, for class representatives s, the proportion of involutions g such that $\langle g,s \rangle \neq S$ holds.

```
———————————————————————— Example ————————————————————————
  gap> g:= AlternatingGroup( 5 );;
  gap> inv:= g.1^2 * g.2;
  (1,4)(2,5)
  gap> cclreps:= List( ConjugacyClasses( g ), Representative );;
  gap> SortParallel( List( cclreps, Order ), cclreps );
  gap> List( cclreps, Order );
  [ 1, 2, 3, 5, 5 ]
  gap> Size( ConjugacyClass( g, inv ) );
  15
  gap> prop:= List( cclreps,
  >                 r -> RatioOfNongenerationTransPermGroup( g, inv, r ) );
  [ 1, 1, 3/5, 1/3, 1/3 ]
  gap> Minimum( prop );
  1/3
```

Statement (d) follows by explicit computations.

```
———————————————————————— Example ————————————————————————
  gap> triple:= [ (1,2)(3,4), (1,3)(2,4), (1,4)(2,3) ];;
  gap> CommonGeneratorWithGivenElements( g, cclreps, triple );
  fail
```

As for statement (e), we know from (a) that the uniform spread of S is at least two, and from (d) that the spread is less than three.

11.5.3 A_6

We show that $S = A_6$ satisfies the following.

(a) $\sigma(S) = 2/3$, and this value is attained exactly for $\sigma(S,s)$ with s of order 5.

(b)   For s of order 5, $\mathbb{M}(S,s)$ consists of two nonconjugate groups of the type A_5.

(c) $P(S) = 5/9$, and this value is attained exactly for $P(S,s)$ with s of order 5.

(d)   Each element in S together with one of $(1,2)(3,4)$, $(1,3)(2,4)$, $(1,4)(2,3)$ generates a proper subgroup of S.

(e)   Both the spread and the uniform spread of S is exactly two (see [BW75]), with s of order 4.

(f)   For $x, y \in S_6^\times$, there is $s \in S_6$ such that $S \subseteq \langle x,s \rangle \cap \langle y,s \rangle$. It is *not* possible to find $s \in S$ with this property, or s in a prescribed conjugacy class of S_6.

(g) $\sigma(\mathrm{PGL}(2,9)) = 1/6$ and $\sigma(M_{10}) = 1/9$, with s of order 10 and 8, respectively.

(Note that in this example, the optimal choice of s for $P(S)$ cannot be used to obtain the result on the exact spread.)

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

```
                            Example
   gap> t:= CharacterTable( "A6" );;
   gap> ProbGenInfoSimple( t );
   [ "A6", 2/3, 1, [ "5A" ], [ 2 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the two classes of maximal subgroups that contain elements of order 5 consist of groups of the structure A_5, see [CCN+85, p. 4].

```
                            Example
   gap> OrdersClassRepresentatives( t );
   [ 1, 2, 3, 3, 4, 5, 5 ]
   gap> prim:= PrimitivePermutationCharacters( t );
   [ Character( CharacterTable( "A6" ), [ 6, 2, 3, 0, 0, 1, 1 ] ),
     Character( CharacterTable( "A6" ), [ 6, 2, 0, 3, 0, 1, 1 ] ),
     Character( CharacterTable( "A6" ), [ 10, 2, 1, 1, 2, 0, 0 ] ),
     Character( CharacterTable( "A6" ), [ 15, 3, 3, 0, 1, 0, 0 ] ),
     Character( CharacterTable( "A6" ), [ 15, 3, 0, 3, 1, 0, 0 ] ) ]
```

For statement (c), we first compute that for all nonidentity elements $s \in S$ and involutions $g \in S$, $P(g, s) \geq 5/9$ holds, with equality if and only if s has order 5. We actually compute, for class representatives s, the proportion of involutions g such that $\langle g, s \rangle \neq S$ holds.

```
                            Example
   gap> S:= AlternatingGroup( 6 );;
   gap> inv:= (S.1*S.2)^2;
   (1,3)(2,5)
   gap> cclreps:= List( ConjugacyClasses( S ), Representative );;
   gap> SortParallel( List( cclreps, Order ), cclreps );
   gap> List( cclreps, Order );
   [ 1, 2, 3, 3, 4, 5, 5 ]
   gap> C:= ConjugacyClass( S, inv );;
   gap> Size( C );
   45
   gap> prop:= List( cclreps,
   >                 r -> RatioOfNongenerationTransPermGroup( S, inv, r ) );
   [ 1, 1, 1, 1, 29/45, 5/9, 5/9 ]
   gap> Minimum( prop );
   5/9
```

Now statement (c) follows from the fact that for $g \in S$ of order larger than two, $\sigma(S, g) \leq 1/2 < 5/9$ holds.

```
                            Example
   gap> ApproxP( prim, 6 );
   [ 0, 2/3, 1/2, 1/2, 0, 1/3, 1/3 ]
```

Statement (d) follows by explicit computations.

```
                            Example
   gap> triple:= [ (1,2)(3,4), (1,3)(2,4), (1,4)(2,3) ];;
   gap> CommonGeneratorWithGivenElements( S, cclreps, triple );
   fail
```

An alternative triple to that in statement (d) is the one given in [BW75].

```
———————————————————— Example ————————————————————
gap> triple:= [ (1,3)(2,4), (1,5)(2,6), (3,6)(4,5) ];;
gap> CommonGeneratorWithGivenElements( S, cclreps, triple );
fail
```

Of course we can also construct such a triple, as follows.

```
———————————————————— Example ————————————————————
gap> TripleWithProperty( [ [ inv ], C, C ],
>        l -> ForAll( S, elm ->
>    ForAny( l, x -> not IsGeneratorsOfTransPermGroup( S, [ elm, x ] ) ) ) );
[ (1,3)(2,5), (1,3)(2,6), (1,3)(2,4) ]
```

For statement (e), we use the random approach described in Section 11.3.3.

```
———————————————————— Example ————————————————————
gap> s:= (1,2,3,4)(5,6);;
gap> reps:= Filtered( cclreps, x -> Order( x ) > 1 );;
gap> ResetGlobalRandomNumberGenerators();
gap> for pair in UnorderedTuples( reps, 2 ) do
>        if RandomCheckUniformSpread( S, pair, s, 40 ) <> true then
>          Print( "#E  nongeneration!\n" );
>        fi;
>    od;
```

We get no output, so a suitable element of order 4 works in all cases. Note that we cannot use an element of order 5, because it fixes a point in the natural permutation representation, and we may take $x_1 = (1,2,3)$ and $x_2 = (4,5,6)$. With this argument, only elements of order 4 and double 3-cycles are possible choices for s, and the latter are excluded by the fact that an outer automorphism maps the class of double s-cycles in A_6 to the class of 3-cycles. So no element in A_6 of order different from 4 works.

Next we show statement (f). Already in $A_6.2_1 = S_6$, elements s of order 4 do in general not work because they do not generate with transpositions.

```
———————————————————— Example ————————————————————
gap> G:= SymmetricGroup( 6 );;
gap> RatioOfNongenerationTransPermGroup( G, s, (1,2) );
1
```

Also, choosing s from a prescribed conjugacy class of S_6 (that is, also s outside A_6 is allowed) with the property that $A_6 \subseteq \langle x,s \rangle \cap \langle y,s \rangle$ is not possible. Note that only 6-cycles are possible for s if x and y are commuting transpositions, and –applying the outer automorphism– no 6-cycle works for two commuting fixed-point free involutions. (The group is small enough for a brute force test.)

```
———————————————————— Example ————————————————————
gap> goods:= Filtered( Elements( G ),
>        s -> IsGeneratorsOfTransPermGroup( G, [ s, (1,2) ] ) and
>            IsGeneratorsOfTransPermGroup( G, [ s, (3,4) ] ) );;
gap> Collected( List( goods, CycleStructurePerm ) );
[ [ [ ,,,, 1 ], 24 ] ]
gap> goods:= Filtered( Elements( G ),
```

```
>          s -> IsGeneratorsOfTransPermGroup( G, [ s, (1,2)(3,4)(5,6) ] ) and
>             IsGeneratorsOfTransPermGroup( G, [ s, (1,3)(2,4)(5,6) ] ) ) );;
gap> Collected( List( goods, CycleStructurePerm ) );
[ [ [ 1, 1 ], 24 ] ]
```

However, for each pair of nonidentity element x, $y \in S_6$, there is $s \in S_6$ such that $\langle x, s \rangle$ and $\langle y, s \rangle$ both contain A_6. (If s works for the pair (x, y) then s^g works for (x^g, y^g), so it is sufficient to consider only orbit representatives (x, y) under the conjugation action of G on pairs. Thus we check conjugacy class representatives x and, for fixed x, representatives of orbits of $C_G(x)$ on the classes y^G, i. e., representatives of $C_G(y)$-$C_G(x)$-double cosets in G. Moreover, clearly we can restrict the checks to elements x, y of prime order.)

```
──────────────── Example ────────────────
gap> Sgens:= GeneratorsOfGroup( S );;
gap> primord:= Filtered( List( ConjugacyClasses( G ), Representative ),
>                         x -> IsPrimeInt( Order( x ) ) );;
gap> for x in primord do
>       for y in primord do
>         for pair in DoubleCosetRepsAndSizes( G, Centralizer( G, y ),
>                         Centralizer( G, x ) ) do
>           if not ForAny( G, s -> IsSubset( Group( x,s ), S ) and
>                                  IsSubset( Group( y^pair[1], s ), S ) ) then
>             Error( [ x, y ] );
>           fi;
>         od;
>       od;
>    od;
```

In other words, the spread of S_6 is 2 but the uniform spread of S_6 is not 2 but only 1.

We cannot always find $s \in A_6$ with the required property: If x is a transposition then any s with $S \subseteq \langle x, s \rangle$ must be a 5-cycle.

```
──────────────── Example ────────────────
gap> filt:= Filtered( S, s -> IsSubset( Group( (1,2), s ), S ) );;
gap> Collected( List( filt, Order ) );
[ [ 5, 48 ] ]
```

Moreover, clearly such s fixes one of the moved points of x, so we may prescribe a transposition $y \neq x$ that commutes with x, it satisfies $S \not\subseteq \langle y, s \rangle$.

For the other two automorphic extensions $A_6.2_2 = \mathrm{PGL}(2, 9)$ and $A_6.2_3 = M_{10}$, we compute the character-theoretic bounds $\sigma(A_6.2_2) = 1/6$ and $\sigma(A_6.2_3) = 1/9$, which shows statement (g).

```
──────────────── Example ────────────────
gap> ProbGenInfoSimple( CharacterTable( "A6.2_2" ) );
[ "A6.2_2", 1/6, 5, [ "10A" ], [ 1 ] ]
gap> ProbGenInfoSimple( CharacterTable( "A6.2_3" ) );
[ "A6.2_3", 1/9, 8, [ "8C" ], [ 1 ] ]
```

Note that $\sigma'(\mathrm{PGL}(2, 9), s) = 1/6$, with s of order 5, and $\sigma'(M_{10}, s) = 0$ for any $s \in A_6$ since M_{10} is a non-split extension of A_6.

```
                          Example
  gap> t:= CharacterTable( "A6" );;
  gap> t2:= CharacterTable( "A6.2_2" );;
  gap> spos:= PositionsProperty( OrdersClassRepresentatives( t ), x -> x = 5 );;
  gap> ProbGenInfoAlmostSimple( t, t2, spos );
  [ "A6.2_2", 1/6, [ "5A", "5B" ], [ 1, 1 ] ]
```

11.5.4 A_7

We show that $S = A_7$ satisfies the following.

(a) $\sigma(S) = 2/5$, and this value is attained exactly for $\sigma(S,s)$ with s of order 7.

(b) For s of order 7, $\mathbb{M}(S,s)$ consists of two nonconjugate subgroups of the type $L_2(7)$.

(c) $P(S) = 2/5$, and this value is attained exactly for $P(S,s)$ with s of order 7.

(d) The uniform spread of S is exactly three, with s of order 7.

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

```
                          Example
  gap> t:= CharacterTable( "A7" );;
  gap> ProbGenInfoSimple( t );
  [ "A7", 2/5, 2, [ "7A" ], [ 2 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the two classes of maximal subgroups that contain elements of order 7 consist of groups of the structure $L_2(7)$, see [CCN+85, p. 10].

```
                          Example
  gap> OrdersClassRepresentatives( t );
  [ 1, 2, 3, 3, 4, 5, 6, 7, 7 ]
  gap> prim:= PrimitivePermutationCharacters( t );
  [ Character( CharacterTable( "A7" ), [ 7, 3, 4, 1, 1, 2, 0, 0, 0 ] ),
    Character( CharacterTable( "A7" ), [ 15, 3, 0, 3, 1, 0, 0, 1, 1 ] ),
    Character( CharacterTable( "A7" ), [ 15, 3, 0, 3, 1, 0, 0, 1, 1 ] ),
    Character( CharacterTable( "A7" ), [ 21, 5, 6, 0, 1, 1, 2, 0, 0 ] ),
    Character( CharacterTable( "A7" ), [ 35, 7, 5, 2, 1, 0, 1, 0, 0 ] )
  ]
```

For statement (c), we compute that for all nonidentity elements $s \in S$ and involutions $g \in S$, $P(g,s) \geq 2/5$ holds, with equality if and only if s has order 7. We actually compute, for class representatives s, the proportion of involutions g such that $\langle g,s \rangle \neq S$ holds.

```
                          Example
  gap> g:= AlternatingGroup( 7 );;
  gap> inv:= (g.1^3*g.2)^3;
  (2,6)(3,7)
  gap> ccl:= List( ConjugacyClasses( g ), Representative );;
  gap> SortParallel( List( ccl, Order ), ccl );
  gap> List( ccl, Order );
  [ 1, 2, 3, 3, 4, 5, 6, 7, 7 ]
```

```
gap> Size( ConjugacyClass( g, inv ) );
105
gap> prop:= List( ccl, r -> RatioOfNongenerationTransPermGroup( g, inv, r ) );
[ 1, 1, 1, 1, 89/105, 17/21, 19/35, 2/5, 2/5 ]
gap> Minimum( prop );
2/5
```

For statement (d), we use the random approach described in Section 11.3.3. By the charac-ter-theoretic bounds, it suffices to consider triples of elements in the classes 2A or 3B.

```
————————————————————— Example —————————————————————
gap> OrdersClassRepresentatives( t );
[ 1, 2, 3, 3, 4, 5, 6, 7, 7 ]
gap> spos:= Position( OrdersClassRepresentatives( t ), 7 );;
gap> SizesCentralizers( t );
[ 2520, 24, 36, 9, 4, 5, 12, 7, 7 ]
gap> ApproxP( prim, spos );
[ 0, 2/5, 0, 2/5, 2/15, 0, 0, 2/15, 2/15 ]
gap> s:= (1,2,3,4,5,6,7);;
gap> 3B:= (1,2,3)(4,5,6);;
gap> C3B:= ConjugacyClass( g, 3B );;
gap> Size( C3B );
280
gap> ResetGlobalRandomNumberGenerators();
gap> for triple in UnorderedTuples( [ inv, 3B ], 3 ) do
>        if RandomCheckUniformSpread( g, triple, s, 80 ) <> true then
>          Print( "#E  nongeneration!\n" );
>        fi;
>      od;
```

We get no output, so the uniform spread of *S* is at least three.

Alternatively, we can use the lemma from Section 11.2.2; this approach is technically more in-volved but faster. We work with the diagonal product of the two degree 15 representations of *S*, which is constructed from the information stored in the GAP Library of Tables of Marks.

```
————————————————————— Example —————————————————————
gap> tom:= TableOfMarks( "A7" );;
gap> a7:= UnderlyingGroup( tom );;
gap> tommaxes:= MaximalSubgroupsTom( tom );
[ [ 39, 38, 37, 36, 35 ], [ 7, 15, 15, 21, 35 ] ]
gap> index15:= List( tommaxes[1]{ [ 2, 3 ] },
>                    i -> RepresentativeTom( tom, i ) );
[ Group([ (1,3)(2,7), (1,5,7)(3,4,6) ]),
  Group([ (1,4)(2,3), (2,4,6)(3,5,7) ]) ]
gap> deg15:= List( index15, s -> RightTransversal( a7, s ) );;
gap> reps:= List( deg15, l -> Action( a7, l, OnRight ) );
[ Group([ (1,5,7)(2,9,10)(3,11,4)(6,12,8)(13,14,15), (1,8,15,5,12)
        (2,13,11,3,10)(4,14,9,7,6) ]),
  Group([ (1,2,3)(4,6,5)(7,8,9)(10,12,11)(13,15,14), (1,12,3,13,10)
        (2,9,15,4,11)(5,6,14,7,8) ]) ]
gap> g:= DiagonalProductOfPermGroups( reps );;
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g );
```

```
>      until Order( s ) = 7;
gap> NrMovedPoints( s );
28
gap> mpg:= MovedPoints( g );;
gap> fixs:= Difference( mpg, MovedPoints( s ) );;
gap> orb_s:= Orbit( g, fixs, OnSets );;
gap> Length( orb_s );
120
gap> SizesCentralizers( t );
[ 2520, 24, 36, 9, 4, 5, 12, 7, 7 ]
gap> repeat 2a:= Random( g ); until Order( 2a ) = 2;
gap> repeat 3b:= Random( g );
>      until Order( 3b ) = 3 and Size( Centralizer( g, 3b ) ) = 9;
gap> orb2a:= Orbit( g, Difference( mpg, MovedPoints( 2a ) ), OnSets );;
gap> orb3b:= Orbit( g, Difference( mpg, MovedPoints( 3b ) ), OnSets );;
gap> orb2aor3b:= Union( orb2a, orb3b );;
gap> TripleWithProperty( [ [ orb2a[1], orb3b[1] ], orb2aor3b, orb2aor3b ],
>         l -> ForAll( orb_s,
>               f -> not IsEmpty( Intersection( Union( l ), f ) ) ) );
fail
```

It remains to show that for any choice of $s \in S$, a quadruple of elements in $S^{\times}$ exists such that s generates a proper subgroup of S together with at least one of these elements.

First we observe (without using GAP) that there is a pair of 3-cycles whose fixed points cover the seven points of the natural permutation representation. This implies the statement for all elements $s \in S$ that fix a point in this representation. So it remains to consider elements s of the orders six and seven.

For the order seven element, the above setup and the lemma from Section 11.2.2 can be used.

```
─────────────── Example ───────────────
gap> QuadrupleWithProperty( [ [ orb2a[1] ], orb2a, orb2a, orb2a ],
>         l -> ForAll( orb_s,
>               f -> not IsEmpty( Intersection( Union( l ), f ) ) ) );
[ [ 2, 5, 12, 18, 19, 26 ], [ 7, 8, 9, 16, 21, 25 ],
  [ 1, 6, 10, 17, 20, 27 ], [ 13, 14, 15, 28, 29, 30 ] ]
```

For the order six element, we use the diagonal product of the primitive permutation representations of the degrees 21 and 35.

```
─────────────── Example ───────────────
gap> has6A:= List( tommaxes[1]{ [ 4, 5 ] },
>                  i -> RepresentativeTom( tom, i ) );
[ Group([ (1,2)(3,7), (2,6,5,4)(3,7) ]),
  Group([ (2,3)(5,7), (1,2)(4,5,6,7), (2,3)(5,6) ]) ]
gap> trans:= List( has6A, s -> RightTransversal( a7, s ) );;
gap> reps:= List( trans, l -> Action( a7, l, OnRight ) );
[ Group([ (1,16,12)(2,17,13)(3,18,11)(4,19,14)(15,20,21), (1,4,7,9,10)
      (2,5,8,3,6)(11,12,15,14,13)(16,20,19,17,18) ]),
  Group([ (2,16,6)(3,17,7)(4,18,8)(5,19,9)(10,20,26)(11,21,27)
      (12,22,28)(13,23,29)(14,24,30)(15,25,31), (1,2,3,4,5)
      (6,10,13,15,9)(7,11,14,8,12)(16,20,23,25,19)(17,21,24,18,22)
      (26,32,35,31,28)(27,33,29,34,30) ]) ]
```

```
gap> g:= DiagonalProductOfPermGroups( reps );;
gap> repeat s:= Random( g );
>    until Order( s ) = 6;
gap> NrMovedPoints( s );
53
gap> mpg:= MovedPoints( g );;
gap> fixs:= Difference( mpg, MovedPoints( s ) );;
gap> orb_s:= Orbit( g, fixs, OnSets );;
gap> Length( orb_s );
105
gap> repeat 3a:= Random( g );
>    until Order( 3a ) = 3 and Size( Centralizer( g, 3a ) ) = 36;
gap> orb3a:= Orbit( g, Difference( mpg, MovedPoints( 3a ) ), OnSets );;
gap> Length( orb3a );
35
gap> TripleWithProperty( [ [ orb3a[1] ], orb3a, orb3a ],
>        l -> ForAll( orb_s,
>                f -> not IsEmpty( Intersection( Union( l ), f ) ) ) );
[ [ 1, 4, 6, 12, 14, 15, 34, 37, 40, 43, 49 ],
  [ 1, 4, 6, 16, 19, 20, 27, 30, 33, 44, 49 ],
  [ 2, 3, 4, 5, 7, 9, 26, 47, 48, 50, 53 ] ]
```

So we have found not only a quadruple but even a triple of 3-cycles that excludes candidates *s* of order six.

11.5.5 $L_d(q)$

In the treatment of small dimensional linear groups $S = \mathrm{SL}(d,q)$, [BGK08] uses a Singer element *s* of order $(q^d - 1)/(q - 1)$. (So the order of the corresponding element in $\mathrm{PSL}(d,q) = (q^d - 1)/[(q - 1)\gcd(d,q-1)]$.) By [Ber00], $\mathbb{M}(S,s)$ consists of extension field type subgroups, except in the cases $d = 2$, $q \in \{2,5,7,9\}$, and $(d,q) = (3,4)$. These subgroups have the structure $\mathrm{GL}(d/p, q^p) : \alpha_q \cap S$, for prime divisors *p* of *d*, where α_q denotes the Frobenius automorphism that acts on matrices by raising each entry to the *q*-th power. (If *q* is a prime then we have $\mathrm{GL}(d/p, q^p) : \alpha_q = \Gamma\mathrm{L}(d/p, q^p)$.) Since *s* acts irreducibly, it is contained in at most one conjugate of each class of extension field type subgroups (cf. [BGK08, Lemma 2.12]).

First we write a GAP function `RelativeSigmaL` that takes a positive integer *d* and a basis *B* of the field extension of degree *n* over the field with *q* elements, and returns the group $\mathrm{GL}(d, q^n) : \alpha_q$, as a subgroup of $\mathrm{GL}(dn, q)$.

```
──────────────── Example ────────────────
gap> RelativeSigmaL:= function( d, B )
>    local n, F, q, glgens, diag, pi, frob, i;
>
>    n:= Length( B );
>    F:= LeftActingDomain( UnderlyingLeftModule( B ) );
>    q:= Size( F );
>
>    # Create the generating matrices inside the linear subgroup.
>    glgens:= List( GeneratorsOfGroup( SL( d, q^n ) ),
>                m -> BlownUpMat( B, m ) );
>
```

```
>       # Create the matrix of a diagonal part that maps to determinant 1.
>       diag:= IdentityMat( d*n, F );
>       diag{ [ 1 .. n ] }{ [ 1 .. n ] }:= BlownUpMat( B, [ [ Z(q^n)^(q-1) ] ] );
>       Add( glgens, diag );
>
>       # Create the matrix that realizes the Frobenius action,
>       # and adjust the determinant.
>       pi:= List( B, b -> Coefficients( B, b^q ) );
>       frob:= NullMat( d*n, d*n, F );
>       for i in [ 0 .. d-1 ] do
>         frob{ [ 1 .. n ] + i*n }{ [ 1 .. n ] + i*n }:= pi;
>       od;
>       diag:= IdentityMat( d*n, F );
>       diag{ [ 1 .. n ] }{ [ 1 .. n ] }:= BlownUpMat( B, [ [ Z(q^n) ] ] );
>       diag:= diag^LogFFE( Inverse( Determinant( frob ) ), Determinant( diag ) );
>
>       # Return the result.
>       return Group( Concatenation( glgens, [ diag * frob ] ) );
> end;;
```

The next function computes $\sigma(\mathrm{SL}(d,q),s)$, by computing the sum of $\mu(g,S/(\mathrm{GL}(d/p,q^p):\alpha_q \cap S))$, for prime divisors p of d, and taking the maximum over $g \in S^\times$. The computations take place in a permutation representation of $\mathrm{PSL}(d,q)$.

```
───────────────────── Example ─────────────────────
gap> ApproxPForSL:= function( d, q )
>       local G, epi, PG, primes, maxes, names, ccl;
>
>       # Check whether this is an admissible case (see [Be00]).
>       if ( d = 2 and q in [ 2, 5, 7, 9 ] ) or ( d = 3 and q = 4 ) then
>         return fail;
>       fi;
>
>       # Create the group SL(d,q), and the map to PSL(d,q).
>       G:= SL( d, q );
>       epi:= ActionHomomorphism( G, NormedRowVectors( GF(q)^d ), OnLines );
>       PG:= ImagesSource( epi );
>
>       # Create the subgroups corresponding to the prime divisors of 'd'.
>       primes:= PrimeDivisors( d );
>       maxes:= List( primes, p -> RelativeSigmaL( d/p,
>                                   Basis( AsField( GF(q), GF(q^p) ) ) ) );
>       names:= List( primes, p -> Concatenation( "GL(", String( d/p ), ",",
>                                   String( q^p ), ").", String( p ) ) );
>       if 2 < q then
>         names:= List( names, name -> Concatenation( name, " cap G" ) );
>       fi;
>
>       # Compute the conjugacy classes of prime order elements in the maxes.
>       # (In order to avoid computing all conjugacy classes of these subgroups,
>       # we work in Sylow subgroups.)
>       ccl:= List( List( maxes, x -> ImagesSet( epi, x ) ),
>               M -> ClassesOfPrimeOrder( M, PrimeDivisors( Size( M ) ),
```

```
>                                          TrivialSubgroup( M ) ) );
>
>     return [ names, UpperBoundFixedPointRatios( PG, ccl, true )[1] ];
> end;;
```

We apply this function to the cases that are interesting in [BGK08, Section 5.12].

```
_____ Example _____
gap> pairs:= [ [ 3, 2 ], [ 3, 3 ], [ 4, 2 ], [ 4, 3 ], [ 4, 4 ],
>             [ 6, 2 ], [ 6, 3 ], [ 6, 4 ], [ 6, 5 ], [ 8, 2 ], [ 10, 2 ] ];;
gap> array:= [];;
gap> for pair in pairs do
>       d:= pair[1];  q:= pair[2];
>       approx:= ApproxPForSL( d, q );
>       Add( array, [ Concatenation( "SL(", String(d), ",", String(q), ")" ),
>                     (q^d-1)/(q-1),
>                     approx[1], approx[2] ] );
>    od;
gap> oldsize:= SizeScreen();;
gap> SizeScreen( [ 80 ] );;
gap> PrintFormattedArray( array );
   SL(3,2)    7                       [ "GL(1,8).3" ]                   1/4
   SL(3,3)   13                 [ "GL(1,27).3 cap G" ]                 1/24
   SL(4,2)   15                       [ "GL(2,4).2" ]                  3/14
   SL(4,3)   40                 [ "GL(2,9).2 cap G" ]             53/1053
   SL(4,4)   85                [ "GL(2,16).2 cap G" ]                1/108
   SL(6,2)   63          [ "GL(3,4).2", "GL(2,8).3" ]           365/55552
   SL(6,3)  364   [ "GL(3,9).2 cap G", "GL(2,27).3 cap G" ] 22843/123845436
   SL(6,4) 1365  [ "GL(3,16).2 cap G", "GL(2,64).3 cap G" ]           1/85932
   SL(6,5) 3906 [ "GL(3,25).2 cap G", "GL(2,125).3 cap G" ]          1/484220
   SL(8,2)  255                       [ "GL(4,4).2" ]               1/7874
  SL(10,2) 1023           [ "GL(5,4).2", "GL(2,32).5" ]             1/129794
gap> SizeScreen( oldsize );;
```

The only missing case for [BGK08] is $S = L_3(4)$, for which $\mathbb{M}(S,s)$ consists of three groups of the type $L_3(2)$ (see [CCN+85, p. 23]). The group $L_3(4)$ has been considered already in Section 11.4.4, where $\sigma(S,s) = 1/5$ has been proved. Also the cases SL(3,3), SL(4,2) $\cong A_8$, and SL(4,3) have been handled there.

An alternative character-theoretic proof for $S = L_6(2)$ looks as follows. In this case, the subgroups in $\mathbb{M}(S,s)$ have the types $\Gamma L(3,4) \cong GL(3,4).2 \cong 3.L_3(4).3.2_2$ and $\Gamma L(2,8) \cong GL(2,8).3 \cong (7 \times L_2(8)).3$.

```
_____ Example _____
gap> t:= CharacterTable( "L6(2)" );;
gap> s1:= CharacterTable( "3.L3(4).3.2_2" );;
gap> s2:= CharacterTable( "(7xL2(8)).3" );;
gap> SigmaFromMaxes( t, "63A", [ s1, s2 ], [ 1, 1 ] );
365/55552
```

11.5.6 ∗ $L_d(q)$ with prime d

For $S = \mathrm{SL}(d,q)$ with *prime* dimension d, and $s \in S$ a Singer cycle, we have $\mathbb{M}(S,s) = \{M\}$, where $M = N_S(\langle s \rangle) \cong \Gamma\mathrm{L}(1,q^d) \cap S$. So

$$\sigma(g,s) = \mu(g, S/M) = |g^S \cap M|/|g^S| < |M|/|g^S| \le (q^d - 1) \cdot d/|g^S|$$

holds for any $g \in S \setminus Z(S)$, which implies $\sigma(S,s) < \max\{(q^d-1) \cdot d/|g^S|; g \in S \setminus Z(S)\}$. The right hand side of this inequality is returned by the following function. In [BGK08, Lemma 3.8], the global upper bound $1/q^d$ is derived for primes $d \ge 5$.

```
————————————————————————— Example —————————————————————————
gap> UpperBoundForSL:= function( d, q )
>      local G, Msize, ccl;
>
>      if not IsPrimeInt( d ) then
>        Error( "<d> must be a prime" );
>      fi;
>
>      G:= SL( d, q );
>      Msize:= (q^d-1) * d;
>      ccl:= Filtered( ConjugacyClasses( G ),
>                   c ->     Msize mod Order( Representative( c ) ) = 0
>                        and Size( c ) <> 1 );
>
>      return Msize / Minimum( List( ccl, Size ) );
> end;;
```

The interesting values are (d,q) with $d \in \{5,7,11\}$ and $q \in \{2,3,4\}$, and perhaps also $(d,q) \in \{(3,2),(3,3)\}$. (Here we exclude SL(11,4) because writing down the conjugacy classes of this group would exceed the permitted memory.)

```
————————————————————————— Example —————————————————————————
gap> NrConjugacyClasses( SL(11,4) );
1397660
gap> pairs:= [ [ 3, 2 ], [ 3, 3 ], [ 5, 2 ], [ 5, 3 ], [ 5, 4 ],
>              [ 7, 2 ], [ 7, 3 ], [ 7, 4 ],
>              [ 11, 2 ], [ 11, 3 ] ];;
gap> array:= [];;
gap> for pair in pairs do
>      d:= pair[1];  q:= pair[2];
>      approx:= UpperBoundForSL( d, q );
>      Add( array, [ Concatenation( "SL(", String(d), ",", String(q), ")" ),
>                 (q^d-1)/(q-1),
>                 approx ] );
>    od;
gap> PrintFormattedArray( array );
   SL(3,2)     7                                      7/8
   SL(3,3)    13                                      3/4
   SL(5,2)    31                                31/64512
   SL(5,3)   121                                   10/81
   SL(5,4)   341                                  15/256
   SL(7,2)   127                               7/9142272
```

```
    SL(7,3)   1093                                      14/729
    SL(7,4)   5461                                      21/4096
    SL(11,2)  2047  2047/34112245508649716682268134604800
    SL(11,3) 88573                                      22/59049
```

The exact values are clearly better than the above bounds. We compute them for $L_5(2)$ and $L_7(2)$. In the latter case, the class fusion of the $127 : 7$ type subgroup M is not uniquely determined by the character tables; here we use the additional information that the elements of order 7 in M have centralizer order 49 in $L_7(2)$. (See Section 11.4.4 for the examples with $d = 3$.)

```
                                 ─── Example ───
  gap> SigmaFromMaxes( CharacterTable( "L5(2)" ), "31A",
  >          [ CharacterTable( "31:5" ) ], [ 1 ] );
  1/5376
  gap> t:= CharacterTable( "L7(2)" );;
  gap> s:= CharacterTable( "P:Q", [ 127, 7 ] );;
  gap> pi:= PossiblePermutationCharacters( s, t );;
  gap> Length( pi );
  2
  gap> ord7:= PositionsProperty( OrdersClassRepresentatives( t ), x -> x = 7 );
  [ 38, 45, 76, 77, 83 ]
  gap> sizes:= SizesCentralizers( t ){ ord7 };
  [ 141120, 141120, 3528, 3528, 49 ]
  gap> List( pi, x -> x[83] );
  [ 42, 0 ]
  gap> spos:= Position( OrdersClassRepresentatives( t ), 127 );;
  gap> Maximum( ApproxP( pi{ [ 1 ] }, spos ) );
  1/4388290560
```

11.5.7 Automorphic Extensions of $L_d(q)$

For the following values of d and q, automorphic extensions G of $L_d(q)$ had to be checked for [BGK08, Section 5.12].

$$(d,q) \in \{(3,4),(6,2),(6,3),(6,4),(6,5),(10,2)\}$$

The first case has been treated in Section 11.4.5. For the other cases, we compute $\sigma'(G,s)$ below.

In any case, the extension by a *graph* automorphism occurs, which can be described by mapping each matrix in $\mathrm{SL}(d,q)$ to its inverse transpose. If $q > 2$, also extensions by *diagonal* automorphisms occur, which are induced by conjugation with elements in $\mathrm{GL}(d,q)$. If q is nonprime then also extensions by *field* automorphisms occur, which can be described by powering the matrix entries by roots of q. Finally, products (of prime order) of these three kinds of automorphisms have to be considered.

We start with the extension G of $S = \mathrm{SL}(d,q)$ by a graph automorphism. G can be embedded into $\mathrm{GL}(2d,q)$ by representing the matrix $A \in S$ as a block diagonal matrix with diagonal blocks equal to A and A^{-tr}, and representing the graph automorphism by a permutation matrix that interchanges the two blocks. In order to construct the field extension type subgroups of G, we have to choose the basis of the field extension in such a way that the subgroup is normalized by the permutation matrix; a sufficient condition is that the matrices of the $\mathbb{F}_q$-linear mappings induced by the basis elements are symmetric.

(We do not give a function that computes a basis with this property from the parameters d and q. Instead, we only write down the bases that we will need.)

```
                              Example
  gap> SymmetricBasis:= function( q, n )
  >     local vectors, B, issymmetric;
  >
  >     if   q = 2 and n = 2 then
  >       vectors:= [ Z(2)^0, Z(2^2) ];
  >     elif q = 2 and n = 3 then
  >       vectors:= [ Z(2)^0, Z(2^3), Z(2^3)^5 ];
  >     elif q = 2 and n = 5 then
  >       vectors:= [ Z(2)^0, Z(2^5), Z(2^5)^4, Z(2^5)^25, Z(2^5)^26 ];
  >     elif q = 3 and n = 2 then
  >       vectors:= [ Z(3)^0, Z(3^2) ];
  >     elif q = 3 and n = 3 then
  >       vectors:= [ Z(3)^0, Z(3^3)^2, Z(3^3)^7 ];
  >     elif q = 4 and n = 2 then
  >       vectors:= [ Z(2)^0, Z(2^4)^3 ];
  >     elif q = 4 and n = 3 then
  >       vectors:= [ Z(2)^0, Z(2^3), Z(2^3)^5 ];
  >     elif q = 5 and n = 2 then
  >       vectors:= [ Z(5)^0, Z(5^2)^2 ];
  >     elif q = 5 and n = 3 then
  >       vectors:= [ Z(5)^0, Z(5^3)^9, Z(5^3)^27 ];
  >     else
  >       Error( "sorry, no basis for <q> and <n> stored" );
  >     fi;
  >
  >     B:= Basis( AsField( GF(q), GF(q^n) ), vectors );
  >
  >     # Check that the basis really has the required property.
  >     issymmetric:= M -> M = TransposedMat( M );
  >     if not ForAll( B, b -> issymmetric( BlownUpMat( B, [ [ b ] ] ) ) ) then
  >       Error( "wrong basis!" );
  >     fi;
  >
  >     # Return the result.
  >     return B;
  > end;;
```

In later examples, we will need similar embeddings of matrices. Therefore, we provide a more general function `EmbeddedMatrix` that takes a field `F`, a matrix `mat`, and a function `func`, and returns a block diagonal matrix over `F` whose diagonal blocks are `mat` and `func( mat )`.

```
                              Example
  gap> BindGlobal( "EmbeddedMatrix", function( F, mat, func )
  >   local d, result;
  >
  >   d:= Length( mat );
  >   result:= NullMat( 2*d, 2*d, F );
  >   result{ [ 1 .. d ] }{ [ 1 .. d ] }:= mat;
  >   result{ [ d+1 .. 2*d ] }{ [ d+1 .. 2*d ] }:= func( mat );
  >
  >   return result;
  > end );
```

The following function is similar to `ApproxPForSL`, the differences are that the group G in question is not $SL(d,q)$ but the extension of this group by a graph automorphism, and that $\sigma'(G,s)$ is computed not $\sigma(G,s)$.

```
─────────────────────────── Example ───────────────────────────
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut:= function( d, q )
>     local embedG, swap, G, orb, epi, PG, Gprime, primes, maxes, ccl, names;
>
>     # Check whether this is an admissible case (see [Be00],
>     # note that a graph automorphism exists only for 'd > 2').
>     if d = 2 or ( d = 3 and q = 4 ) then
>       return fail;
>     fi;
>
>     # Provide a function that constructs a block diagonal matrix.
>     embedG:= mat -> EmbeddedMatrix( GF( q ), mat,
>                                       M -> TransposedMat( M^-1 ) );
>
>     # Create the matrix that exchanges the two blocks.
>     swap:= NullMat( 2*d, 2*d, GF(q) );
>     swap{ [ 1 .. d ] }{ [ d+1 .. 2*d ] }:= IdentityMat( d, GF(q) );
>     swap{ [ d+1 .. 2*d ] }{ [ 1 .. d ] }:= IdentityMat( d, GF(q) );
>
>     # Create the group SL(d,q).2, and the map to the projective group.
>     G:= ClosureGroupDefault( Group( List( GeneratorsOfGroup( SL( d, q ) ),
>                                       embedG ) ),
>                       swap );
>     orb:= Orbit( G, One( G )[1], OnLines );
>     epi:= ActionHomomorphism( G, orb, OnLines );
>     PG:= ImagesSource( epi );
>     Gprime:= DerivedSubgroup( PG );
>
>     # Create the subgroups corresponding to the prime divisors of 'd'.
>     primes:= PrimeDivisors( d );
>     maxes:= List( primes,
>             p -> ClosureGroupDefault( Group( List( GeneratorsOfGroup(
>                       RelativeSigmaL( d/p, SymmetricBasis( q, p ) ) ) ),
>                       embedG ) ),
>                    swap ) );
>
>     # Compute conjugacy classes of outer involutions in the maxes.
>     # (In order to avoid computing all conjugacy classes of these subgroups,
>     # we work in the Sylow $2$ subgroups.)
>     maxes:= List( maxes, M -> ImagesSet( epi, M ) );
>     ccl:= List( maxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );
>     names:= List( primes, p -> Concatenation( "GL(", String( d/p ), ",",
>                                 String( q^p ), ").", String( p ) ) );
>
>     return [ names, UpperBoundFixedPointRatios( PG, ccl, true )[1] ];
> end;;
```

And these are the results for the groups we are interested in (and others).

```
─────────────────────────── Example ───────────────────────────
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 4, 3 );
[ [ "GL(2,9).2" ], 17/117 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 4, 4 );
[ [ "GL(2,16).2" ], 73/1008 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 6, 2 );
[ [ "GL(3,4).2", "GL(2,8).3" ], 41/1984 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 6, 3 );
[ [ "GL(3,9).2", "GL(2,27).3" ], 541/352836 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 6, 4 );
[ [ "GL(3,16).2", "GL(2,64).3" ], 3265/12570624 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 6, 5 );
[ [ "GL(3,25).2", "GL(2,125).3" ], 13001/195250000 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 8, 2 );
[ [ "GL(4,4).2" ], 367/1007872 ]
gap> ApproxPForOuterClassesInExtensionOfSLByGraphAut( 10, 2 );
[ [ "GL(5,4).2", "GL(2,32).5" ], 609281/476346056704 ]
```

Now we consider diagonal automorphisms. We modify the approach for $SL(d,q)$ by constructing the field extension type subgroups of $GL(d,q)$...

```
─────────────────────────── Example ───────────────────────────
gap> RelativeGammaL:= function( d, B )
>     local n, F, q, diag;
>
>     n:= Length( B );
>     F:= LeftActingDomain( UnderlyingLeftModule( B ) );
>     q:= Size( F );
>     diag:= IdentityMat( d * n, F );
>     diag{[ 1 .. n ]}{[ 1 .. n ]}:= BlownUpMat( B, [ [ Z(q^n) ] ] );
>     return ClosureGroup( RelativeSigmaL( d, B ),  diag );
> end;;
```

... and counting the elements of prime order outside the simple group.

```
─────────────────────────── Example ───────────────────────────
gap> ApproxPForOuterClassesInGL:= function( d, q )
>     local G, epi, PG, Gprime, primes, maxes, names;
>
>     # Check whether this is an admissible case (see [Be00]).
>     if ( d = 2 and q in [ 2, 5, 7, 9 ] ) or ( d = 3 and q = 4 ) then
>       return fail;
>     fi;
>
>     # Create the group GL(d,q), and the map to PGL(d,q).
>     G:= GL( d, q );
>     epi:= ActionHomomorphism( G, NormedRowVectors( GF(q)^d ), OnLines );
>     PG:= ImagesSource( epi );
>     Gprime:= ImagesSet( epi, SL( d, q ) );
>
>     # Create the subgroups corresponding to the prime divisors of 'd'.
>     primes:= PrimeDivisors( d );
>     maxes:= List( primes, p -> RelativeGammaL( d/p,
```

```
>                                      Basis( AsField( GF(q), GF(q^p) ) ) ) );
>       maxes:= List( maxes, M -> ImagesSet( epi, M ) );
>       names:= List( primes, p -> Concatenation( "M(", String( d/p ), ",",
>                                      String( q^p ), ")" ) );
>
>       return [ names,
>               UpperBoundFixedPointRatios( PG, List( maxes,
>                   M -> ClassesOfPrimeOrder( M,
>                           PrimeDivisors( Index( PG, Gprime ) ), Gprime ) ),
>               true )[1] ];
> end;;
```

Here are the required results.

```
───────────────── Example ─────────────────
gap> ApproxPForOuterClassesInGL( 6, 3 );
[ [ "M(3,9)", "M(2,27)" ], 41/882090 ]
gap> ApproxPForOuterClassesInGL( 4, 3 );
[ [ "M(2,9)" ], 0 ]
gap> ApproxPForOuterClassesInGL( 6, 4 );
[ [ "M(3,16)", "M(2,64)" ], 1/87296 ]
gap> ApproxPForOuterClassesInGL( 6, 5 );
[ [ "M(3,25)", "M(2,125)" ], 821563/756593750000 ]
```

(Note that the extension field type subgroup in $\mathrm{PGL}(4,3) = L_4(3).2_1$ is a *non-split* extension of its intersection with $L_4(3)$, hence the zero value.)

Concerning extensions by Frobenius automorphisms, only the case $(d,q) = (6,4)$ is interesting in [BGK08]. In fact, we would not need to compute anything for the extension G of $S = \mathrm{SL}(6,4)$ by the Frobenius map that squares each matrix entry. This is because $\mathbb{M}'(G,s)$ consists of the normalizers of the two subgroups of the types $\mathrm{SL}(3,16)$ and $\mathrm{SL}(2,64)$, and the former maximal subgroup is a *non-split* extension of its intersection with S, so only one maximal subgroup can contribute to $\sigma'(G,s)$, which is thus smaller than $1/2$, by [BGK08, Prop. 2.6].

However, it is easy enough to compute the exact value of $\sigma'(G,s)$. We work with the projective action of S on its natural module, and compute the permutation induced by the Frobenius map as the Frobenius action on the normed row vectors.

```
───────────────── Example ─────────────────
gap> matgrp:= SL(6,4);;
gap> dom:= NormedRowVectors( GF(4)^6 );;
gap> Gprime:= Action( matgrp, dom, OnLines );;
gap> pi:= PermList( List( dom, v -> Position( dom, List( v, x -> x^2 ) ) ) );;
gap> G:= ClosureGroup( Gprime, pi );;
```

Then we compute the maximal subgroups, the classes of outer involutions, and the bound, similar to the situation with graph automorphisms.

```
───────────────── Example ─────────────────
gap> maxes:= List( [ 2, 3 ], p -> Normalizer( G,
>             Action( RelativeSigmaL( 6/p,
>               Basis( AsField( GF(4), GF(4^p) ) ) ), dom, OnLines ) ) );;
gap> ccl:= List( maxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );;
gap> List( ccl, Length );
```

```
    [ 0, 1 ]
    gap> UpperBoundFixedPointRatios( G, ccl, true );
    [ 1/34467840, true ]
```

For $(d,q) = (6,4)$, we have to consider also the extension G of $S = \mathrm{SL}(6,4)$ by the product α of the Frobenius map and the graph automorphism. We use the same approach as for the graph automorphism, i. e., we embed $\mathrm{SL}(6,4)$ into a 12-dimensional group of 6×6 block matrices, where the second block is the image of the first block under α, and describe α by the transposition of the two blocks.

First we construct the projective actions of S and G on an orbit of 1-spaces.

```
                              ──── Example ────
    gap> embedFG:= function( F, mat )
    >        return EmbeddedMatrix( F, mat,
    >                    M -> List( TransposedMat( M^-1 ),
    >                             row -> List( row, x -> x^2 ) ) );
    >        end;;
    gap> d:= 6;;   q:= 4;;
    gap> alpha:= NullMat( 2*d, 2*d, GF(q) );;
    gap> alpha{ [ 1 .. d ] }{ [ d+1 .. 2*d ] }:= IdentityMat( d, GF(q) );;
    gap> alpha{ [ d+1 .. 2*d ] }{ [ 1 .. d ] }:= IdentityMat( d, GF(q) );;
    gap> Gprime:= Group( List( GeneratorsOfGroup( SL(d,q) ),
    >                         mat -> embedFG( GF(q), mat ) ) );;
    gap> G:= ClosureGroupDefault( Gprime, alpha );;
    gap> orb:= Orbit( G, One( G )[1], OnLines );;
    gap> G:= Action( G, orb, OnLines );;
    gap> Gprime:= Action( Gprime, orb, OnLines );;
```

Next we construct the maximal subgroups, the classes of outer involutions, and the bound.

```
                              ──── Example ────
    gap> maxes:= List( PrimeDivisors( d ), p -> Group( List( GeneratorsOfGroup(
    >              RelativeSigmaL( d/p, Basis( AsField( GF(q), GF(q^p) ) ) ) ) ),
    >                mat -> embedFG( GF(q), mat ) ) ) );;
    gap> maxes:= List( maxes, x -> Action( x, orb, OnLines ) );;
    gap> maxes:= List( maxes, x -> Normalizer( G, x ) );;
    gap> ccl:= List( maxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );;
    gap> List( ccl, Length );
    [ 0, 1 ]
    gap> UpperBoundFixedPointRatios( G, ccl, true );
    [ 1/10792960, true ]
```

The only missing cases are the extensions of $\mathrm{SL}(6,3)$ and $\mathrm{SL}(6,5)$ by the involutory outer automorphism that acts as the product of a diagonal and a graph automorphism.

In the case $S = \mathrm{SL}(6,3)$, we can directly write down the extension G.

```
                              ──── Example ────
    gap> d:= 6;;   q:= 3;;
    gap> diag:= IdentityMat( d, GF(q) );;
    gap> diag[1][1]:= Z(q);;
    gap> embedDG:= mat -> EmbeddedMatrix( GF(q), mat,
    >                                    M -> TransposedMat( M^-1 )^diag );;
```

```
gap> Gprime:= Group( List( GeneratorsOfGroup( SL(d,q) ), embedDG ) );;
gap> alpha:= NullMat( 2*d, 2*d, GF(q) );;
gap> alpha{ [ 1 .. d ] }{ [ d+1 .. 2*d ] }:= IdentityMat( d, GF(q) );;
gap> alpha{ [ d+1 .. 2*d ] }{ [ 1 .. d ] }:= IdentityMat( d, GF(q) );;
gap> G:= ClosureGroupDefault( Gprime, alpha );;
```

The maximal subgroups are constructed as the normalizers in *G* of the extension field type subgroups in *S*. We work with a permutation representation of *G*.

```
─────────────── Example ───────────────
gap> maxes:= List( PrimeDivisors( d ), p -> Group( List( GeneratorsOfGroup(
>             RelativeSigmaL( d/p, Basis( AsField( GF(q), GF(q^p) ) ) ) ) ),
>                embedDG ) ) );;
gap> orb:= Orbit( G, One( G )[1], OnLines );;
gap> G:= Action( G, orb, OnLines );;
gap> Gprime:= Action( Gprime, orb, OnLines );;
gap> maxes:= List( maxes, M -> Normalizer( G, Action( M, orb, OnLines ) ) );;
gap> ccl:= List( maxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );;
gap> List( ccl, Length );
[ 1, 1 ]
gap> UpperBoundFixedPointRatios( G, ccl, true );
[ 25/352836, true ]
```

For $S = \mathrm{SL}(6,5)$, this approach does not work because we cannot realize the diagonal involution by an involutory matrix. Instead, we consider the extension of $\mathrm{GL}(6,5) \cong 2.(2 \times L_6(5)).2$ by the graph automorphism α, which can be embedded into $\mathrm{GL}(12,5)$.

```
─────────────── Example ───────────────
gap> d:= 6;;  q:= 5;;
gap> embedG:= mat -> EmbeddedMatrix( GF(q),
>                                  mat, M -> TransposedMat( M^-1 ) );;
gap> Gprime:= Group( List( GeneratorsOfGroup( SL(d,q) ), embedG ) );;
gap> maxes:= List( PrimeDivisors( d ), p -> Group( List( GeneratorsOfGroup(
>             RelativeSigmaL( d/p, Basis( AsField( GF(q), GF(q^p) ) ) ) ) ),
>                embedG ) ) );;
gap> diag:= IdentityMat( d, GF(q) );;
gap> diag[1][1]:= Z(q);;
gap> diag:= embedG( diag );;
gap> alpha:= NullMat( 2*d, 2*d, GF(q) );;
gap> alpha{ [ 1 .. d ] }{ [ d+1 .. 2*d ] }:= IdentityMat( d, GF(q) );;
gap> alpha{ [ d+1 .. 2*d ] }{ [ 1 .. d ] }:= IdentityMat( d, GF(q) );;
gap> G:= ClosureGroupDefault( Gprime, alpha * diag );;
```

Now we switch to the permutation action of this group on the 1-dimensional subspaces, thus factoring out the cyclic normal subgroup of order four. In this action, the involutory diagonal automorphism is represented by an involution, and we can proceed as above.

```
─────────────── Example ───────────────
gap> orb:= Orbit( G, One( G )[1], OnLines );;
gap> Gprime:= Action( Gprime, orb, OnLines );;
gap> G:= Action( G, orb, OnLines );;
gap> maxes:= List( maxes, M -> Action( M, orb, OnLines ) );;
gap> extmaxes:= List( maxes, M -> Normalizer( G, M ) );;
```

```
gap> ccl:= List( extmaxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );;
gap> List( ccl, Length );
[ 2, 1 ]
gap> UpperBoundFixedPointRatios( G, ccl, true );
[ 3863/6052750000, true ]
```

In the same way, we can recheck the values for the extensions of SL(6,5) by the diagonal or by the graph automorphism.

```
                             ——— Example ———
gap> diag:= Permutation( diag, orb, OnLines );;
gap> G:= ClosureGroupDefault( Gprime, diag );;
gap> extmaxes:= List( maxes, M -> Normalizer( G, M ) );;
gap> ccl:= List( extmaxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );;
gap> List( ccl, Length );
[ 3, 1 ]
gap> UpperBoundFixedPointRatios( G, ccl, true );
[ 821563/756593750000, true ]
gap> alpha:= Permutation( alpha, orb, OnLines );;
gap> G:= ClosureGroupDefault( Gprime, alpha );;
gap> extmaxes:= List( maxes, M -> Normalizer( G, M ) );;
gap> ccl:= List( extmaxes, M -> ClassesOfPrimeOrder( M, [ 2 ], Gprime ) );;
gap> List( ccl, Length );
[ 2, 2 ]
gap> UpperBoundFixedPointRatios( G, ccl, true );
[ 13001/195250000, true ]
```

gap> t2:= CharacterTable("L6(2).2");; gap> map:= InverseMap(GetFusionMap(t, t2));; gap> torso:= List(Concatenation(prim), pi -> CompositionMaps(pi, map));; gap> ext:= List(torso, x -> PermChars(t2, rec(torso:= x))); [[Character(CharacterTable("L6(2).2"), [55552, 0, 128, 256, 337, 112, 22, 0, 0, 16, 0, 16, 2, 17, 0, 0, 8, 2, 4, 28, 0, 0, 0, 4, 1, 0, 1, 0, 0, 4, 0, 2, 2, 2, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1120, 192, 32, 0, 0, 40, 13, 0, 4, 6, 0, 4, 4, 4, 0, 2, 8, 5, 0, 2, 0, 0, 0, 0, 1, 0, 1, 0])], [Character(CharacterTable("L6(2).2"), [1904640, 0, 0, 512, 960, 0, 120, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 73, 24, 3, 0, 0, 15, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 960, 960, 0, 0, 0, 24, 0, 12, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 0, 0, 0])]] gap> sigma:= ApproxP(Concatenation(ext), > Position(OrdersClassRepresentatives(t2), 63));; gap> Maximum(sigma{ Difference(PositionsProperty(> OrdersClassRepresentatives(t2), IsPrimeInt), > ClassPositionsOfDerivedSubgroup(t2)) }); 41/1984 -->

11.5.8 $L_3(2)$

We show that $S = L_3(2) = \mathrm{SL}(3,2)$ satisfies the following.

(a) $\sigma(S) = 1/4$, and this value is attained exactly for $\sigma(S,s)$ with s of order 7.

(b)   For s of order 7, $\mathbb{M}(S,s)$ consists of one group of the type $7:3$.

(c) $P(S) = 1/4$, and this value is attained exactly for $P(S,s)$ with s of order 7.

(d)   The uniform spread of S is at exactly three, with s of order 7, and the spread of S is exactly four. (This had been left open in [BW75].)

(Note that in this example, the spread and the uniform spread differ.)

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

```
——————————————— Example ———————————————
gap> t:= CharacterTable( "L3(2)" );;
gap> ProbGenInfoSimple( t );
[ "L3(2)", 1/4, 3, [ "7A" ], [ 1 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the unique class of maximal subgroups that contain elements of order 7 consists of groups of the structure $7 : 3$, see [CCN$^+$85, p. 3].

```
——————————————— Example ———————————————
gap> OrdersClassRepresentatives( t );
[ 1, 2, 3, 4, 7, 7 ]
gap> PrimitivePermutationCharacters( t );
[ Character( CharacterTable( "L3(2)" ), [ 7, 3, 1, 1, 0, 0 ] ),
  Character( CharacterTable( "L3(2)" ), [ 7, 3, 1, 1, 0, 0 ] ),
  Character( CharacterTable( "L3(2)" ), [ 8, 0, 2, 0, 1, 1 ] ) ]
```

For the other statements, we will use the primitive permutation representations on 7 and 8 points of S (computed from the GAP Library of Tables of Marks), and their diagonal products of the degrees 14 and 15.

```
——————————————— Example ———————————————
gap> tom:= TableOfMarks( "L3(2)" );;
gap> g:= UnderlyingGroup( tom );
Group([ (2,4)(5,7), (1,2,3)(4,5,6) ])
gap> mx:= MaximalSubgroupsTom( tom );
[ [ 14, 13, 12 ], [ 7, 7, 8 ] ]
gap> maxes:= List( mx[1], i -> RepresentativeTom( tom, i ) );;
gap> tr:= List( maxes, s -> RightTransversal( g, s ) );;
gap> acts:= List( tr, x -> Action( g, x, OnRight ) );;
gap> g7:= acts[1];
Group([ (3,4)(6,7), (1,3,2)(4,6,5) ])
gap> g8:= acts[3];
Group([ (1,6)(2,5)(3,8)(4,7), (1,7,3)(2,5,8) ])
gap> g14:= DiagonalProductOfPermGroups( acts{ [ 1, 2 ] } );
Group([ (3,4)(6,7)(11,13)(12,14), (1,3,2)(4,6,5)(8,11,9)(10,12,13) ])
gap> g15:= DiagonalProductOfPermGroups( acts{ [ 2, 3 ] } );
Group([ (4,6)(5,7)(8,13)(9,12)(10,15)(11,14), (1,4,2)(3,5,6)(8,14,10)
  (9,12,15) ])
```

First we compute that for all nonidentity elements $s \in S$ and order three elements $g \in S$, $P(g, s) \geq 1/4$ holds, with equality if and only if s has order 7; this implies statement (c). We actually compute, for class representatives s, the proportion of order three elements g such that $\langle g, s \rangle \neq S$ holds.

```
——————————————— Example ———————————————
gap> ccl:= List( ConjugacyClasses( g7 ), Representative );;
gap> SortParallel( List( ccl, Order ), ccl );
gap> List( ccl, Order );
[ 1, 2, 3, 4, 7, 7 ]
gap> Size( ConjugacyClass( g7, ccl[3] ) );
```

```
56
gap> prop:= List( ccl,
>                   r -> RatioOfNongenerationTransPermGroup( g7, ccl[3], r ) );
[ 1, 5/7, 19/28, 2/7, 1/4, 1/4 ]
gap> Minimum( prop );
1/4
```

Now we show that the uniform spread of *S* is less than four. In any of the primitive permutation representations of degree seven, we find three involutions whose sets of fixed points cover the seven points. The elements *s* of order different from 7 in *S* fix a point in this representation, so each such *s* generates a proper subgroup of *S* together with one of the three involutions.

```
——————————— Example ———————————
gap> x:= g7.1;
(3,4)(6,7)
gap> fix:= Difference( MovedPoints( g7 ), MovedPoints( x ) );
[ 1, 2, 5 ]
gap> orb:= Orbit( g7, fix, OnSets );
[ [ 1, 2, 5 ], [ 1, 3, 4 ], [ 2, 3, 6 ], [ 2, 4, 7 ], [ 1, 6, 7 ],
  [ 3, 5, 7 ], [ 4, 5, 6 ] ]
gap> Union( orb{ [ 1, 2, 5 ] } ) = [ 1 .. 7 ];
true
```

So we still have to exclude elements *s* of order 7. In the primitive permutation representation of *S* on eight points, we find four elements of order three whose sets of fixed points cover the set of all points that are moved by *S*, so with each element of order seven in *S*, one of them generates an intransitive group.

```
——————————— Example ———————————
gap> three:= g8.2;
(1,7,3)(2,5,8)
gap> fix:= Difference( MovedPoints( g8 ), MovedPoints( three ) );
[ 4, 6 ]
gap> orb:= Orbit( g8, fix, OnSets );;
gap> QuadrupleWithProperty( [ [ fix ], orb, orb, orb ],
>        list -> Union( list ) = [ 1 .. 8 ] );
[ [ 4, 6 ], [ 1, 7 ], [ 3, 8 ], [ 2, 5 ] ]
```

Together with statement (a), this proves that the uniform spread of *S* is exactly three, with *s* of order seven.

Each element of *S* fixes a point in the permutation representation on 15 points. So for proving that the spread of *S* is less than five, it is sufficient to find a quintuple of elements whose sets of fixed points cover all 15 points. (From the permutation characters it is clear that four of these elements must have order three, and the fifth must be an involution.)

```
——————————— Example ———————————
gap> x:= g15.1;
(4,6)(5,7)(8,13)(9,12)(10,15)(11,14)
gap> fixx:= Difference( MovedPoints( g15 ), MovedPoints( x ) );
[ 1, 2, 3 ]
gap> orbx:= Orbit( g15, fixx, OnSets );
[ [ 1, 2, 3 ], [ 1, 4, 5 ], [ 1, 6, 7 ], [ 2, 4, 6 ], [ 3, 4, 7 ],
```

```
   [ 3, 5, 6 ], [ 2, 5, 7 ] ]
gap> y:= g15.2;
(1,4,2)(3,5,6)(8,14,10)(9,12,15)
gap> fixy:= Difference( MovedPoints( g15 ), MovedPoints( y ) );
[ 7, 11, 13 ]
gap> orby:= Orbit( g15, fixy, OnSets );;
gap> QuadrupleWithProperty( [ [ fixy ], orby, orby, orby ],
>         l -> Difference( [ 1 .. 15 ], Union( l ) ) in orbx );
[ [ 7, 11, 13 ], [ 5, 8, 14 ], [ 1, 10, 15 ], [ 3, 9, 12 ] ]
```

It remains to show that the spread of *S* is (at least) four. By the consideration of permutation characters, we know that we can find a suitable order seven element for all quadruples in question except perhaps quadruples of order three elements. We show that for each such case, we can choose *s* of order four. Since $\mathbb{M}(S,s)$ consists of two subgroups of the type S_4, we work with the representation on 14 points.)

First we compute *s* and the *S*-orbit of its fixed points, and the *S*-orbit of the fixed points of an element *x* of order three. Then we prove that for each quadruple of conjugates of *x*, the union of their fixed points intersects the fixed points of at least one conjugate of *s* trivially.

```
────────────────────────── Example ──────────────────────────
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g14 );
>    until Order( s ) = 4;
gap> s;
(1,3)(2,6,7,5)(9,11,10,12)(13,14)
gap> fixs:= Difference( MovedPoints( g14 ), MovedPoints( s ) );
[ 4, 8 ]
gap> orbs:= Orbit( g14, fixs, OnSets );;
gap> Length( orbs );
21
gap> three:= g14.2;
(1,3,2)(4,6,5)(8,11,9)(10,12,13)
gap> fix:= Difference( MovedPoints( g14 ), MovedPoints( three ) );
[ 7, 14 ]
gap> orb:= Orbit( g14, fix, OnSets );;
gap> Length( orb );
28
gap> QuadrupleWithProperty( [ [ fix ], orb, orb, orb ],
>         l -> ForAll( orbs, o -> not IsEmpty( Intersection( o,
>                       Union( l ) ) ) ) );
fail
```

By the lemma from Section 11.2.2, we are done.

11.5.9　M_{11}

We show that $S = M_{11}$ satisfies the following.

(a)　$\sigma(S) = 1/3$, and this value is attained exactly for $\sigma(S,s)$ with *s* of order 11.

(b)　For *s* of order 11, $\mathbb{M}(S,s)$ consists of one group of the type $L_2(11)$.

(c) $P(S) = 1/3$, and this value is attained exactly for $P(S, s)$ with s of order 11.

(d)   Both the uniform spread and the spread of S is exactly three, with s of order 11.

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.1.

```
———————————————————— Example ————————————————————
gap> t:= CharacterTable( "M11" );;
gap> ProbGenInfoSimple( t );
[ "M11", 1/3, 2, [ "11A" ], [ 1 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the unique class of maximal subgroups that contain elements of order 11 consists of groups of the structure $L_2(11)$, see [CCN$^+$85, p. 18].

```
———————————————————— Example ————————————————————
gap> OrdersClassRepresentatives( t );
[ 1, 2, 3, 4, 5, 6, 8, 8, 11, 11 ]
gap> PrimitivePermutationCharacters( t );
[ 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" ),
  [ 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" ),
  [ 165, 13, 3, 1, 0, 1, 1, 1, 0, 0 ] ) ]
gap> Maxes( t );
[ "A6.2_3", "L2(11)", "3^2:Q8.2", "A5.2", "2.S4" ]
```

For the other statements, we will use the primitive permutation representations of S on 11 and 12 points (which are fetched from the Atlas of Group Representations [WWT$^+$]), and their diagonal product.

```
———————————————————— Example ————————————————————
gap> gens11:= OneAtlasGeneratingSet( "M11", NrMovedPoints, 11 );
rec( charactername := "1a+10a", constituents := [ 1, 2 ],
  contents := "core",
  generators := [ (2,10)(4,11)(5,7)(8,9), (1,4,3,8)(2,5,6,9) ],
  groupname := "M11", id := "",
  identifier := [ "M11", [ "M11G1-p11B0.m1", "M11G1-p11B0.m2" ], 1,
      11 ], isPrimitive := true, maxnr := 1, p := 11, rankAction := 2,
  repname := "M11G1-p11B0", repnr := 1, size := 7920,
  stabilizer := "A6.2_3", standardization := 1, transitivity := 4,
  type := "perm" )
gap> g11:= GroupWithGenerators( gens11.generators );;
gap> gens12:= OneAtlasGeneratingSet( "M11", NrMovedPoints, 12 );;
gap> g12:= GroupWithGenerators( gens12.generators );;
gap> g23:= DiagonalProductOfPermGroups( [ g11, g12 ] );
Group([ (2,10)(4,11)(5,7)(8,9)(12,17)(13,20)(16,18)(19,21), (1,4,3,8)
  (2,5,6,9)(12,17,18,15)(13,19)(14,20)(16,22,23,21) ])
```

First we compute that for all nonidentity elements $s \in S$ and involutions $g \in S$, $P(g,s) \geq 1/3$ holds, with equality if and only if s has order 11; this implies statement (c). We actually compute, for class representatives s, the proportion of involutions g such that $\langle g,s \rangle \neq S$ holds.

```
——————————————— Example ———————————————
gap> inv:= g11.1;
(2,10)(4,11)(5,7)(8,9)
gap> ccl:= List( ConjugacyClasses( g11 ), Representative );;
gap> SortParallel( List( ccl, Order ), ccl );
gap> List( ccl, Order );
[ 1, 2, 3, 4, 5, 6, 8, 8, 11, 11 ]
gap> Size( ConjugacyClass( g11, inv ) );
165
gap> prop:= List( ccl,
>                 r -> RatioOfNongenerationTransPermGroup( g11, inv, r ) );
[ 1, 1, 1, 149/165, 25/33, 31/55, 23/55, 23/55, 1/3, 1/3 ]
gap> Minimum( prop );
1/3
```

For the first part of statement (d), we have to deal only with the case of triples of involutions.

The 11-cycle s is contained in exactly one maximal subgroup of S, of index 12. By Corollary 1 in Section 11.2.2, it is enough to show that in the primitive degree 12 representation of S, the fixed points of no triple (x_1, x_2, x_3) of involutions in S can cover all twelve points; equivalenly (considering complements), we show that there is no triple such that the intersection of the sets of *moved* points is empty.

```
——————————————— Example ———————————————
gap> inv:= g12.1;
(1,6)(2,9)(5,7)(8,10)
gap> moved:= MovedPoints( inv );
[ 1, 2, 5, 6, 7, 8, 9, 10 ]
gap> orb12:= Orbit( g12, moved, OnSets );;
gap> Length( orb12 );
165
gap> TripleWithProperty( [ orb12{[1]}, orb12, orb12 ],
>         list -> IsEmpty( Intersection( list ) ) );
fail
```

This implies that the uniform spread of S is at least three.

Now we show that there is a quadruple consisting of one element of order three and three involutions whose fixed points cover all points in the degree 23 representation constructed above; since the permutation character of this representation is strictly positive, this implies that S does not have spread four, by Corollary 2 in Section 11.2.2, and we have proved statement (d).

```
——————————————— Example ———————————————
gap> inv:= g23.1;
(2,10)(4,11)(5,7)(8,9)(12,17)(13,20)(16,18)(19,21)
gap> moved:= MovedPoints( inv );
[ 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 16, 17, 18, 19, 20, 21 ]
gap> orb23:= Orbit( g23, moved, OnSets );;
gap> three:= ( g23.1*g23.2^2 )^2;
(2,6,10)(4,8,7)(5,9,11)(12,17,23)(15,18,16)(19,21,22)
gap> movedthree:= MovedPoints( three );;
```

```
gap> QuadrupleWithProperty( [ [ movedthree ], orb23, orb23, orb23 ],
>          list -> IsEmpty( Intersection( list ) ) );
[ [ 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 21, 22, 23 ],
  [ 1, 3, 4, 5, 6, 8, 9, 10, 12, 13, 14, 16, 17, 18, 20, 21 ],
  [ 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14, 15, 18, 19, 20, 23 ],
  [ 1, 2, 3, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 20, 22, 23 ] ]
```

11.5.10 M_{12}

We show that $S = M_{12}$ satisfies the following.

(a) $\sigma(S) = 1/3$, and this value is attained exactly for $\sigma(S, s)$ with s of order 10.

(b)   For $s \in S$ of order 10, $\mathbb{M}(S, s)$ consists of two nonconjugate subgroups of the type $A_6.2^2$, and one group of the type $2 \times S_5$.

(c) $P(S) = 31/99$, and this value is attained exactly for $P(S, s)$ with s of order 10.

(d)   The uniform spread of S is at least three, with s of order 10.

(e) $\sigma'(\mathrm{Aut}(S), s) = 4/99$.

     Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.1.

```
                           ——— Example ———
  gap> t:= CharacterTable( "M12" );;
  gap> ProbGenInfoSimple( t );
  [ "M12", 1/3, 2, [ "10A" ], [ 3 ] ]
```

     Statement (b) can be read off from the permutation characters, and the fact that the only classes of maximal subgroups that contain elements of order 10 consist of groups of the structures $A_6.2^2$ (two classes) and $2 \times S_5$ (one class), see [CCN$^+$85, p. 33].

```
                           ——— Example ———
  gap> spos:= Position( OrdersClassRepresentatives( t ), 10 );
  13
  gap> prim:= PrimitivePermutationCharacters( t );;
  gap> List( prim, x -> x{ [ 1, spos ] } );
  [ [ 12, 0 ], [ 12, 0 ], [ 66, 1 ], [ 66, 1 ], [ 144, 0 ], [ 220, 0 ],
    [ 220, 0 ], [ 396, 1 ], [ 495, 0 ], [ 495, 0 ], [ 1320, 0 ] ]
  gap> Maxes( t );
  [ "M11", "M12M2", "A6.2^2", "M12M4", "L2(11)", "3^2.2.S4", "M12M7",
    "2xS5", "M8.S4", "4^2:D12", "A4xS3" ]
```

     For statement (c) (which implies statement (d)), we use the primitive permutation representation on 12 points.

```
                           ——— Example ———
  gap> g:= MathieuGroup( 12 );
  Group([ (1,2,3,4,5,6,7,8,9,10,11), (3,7,11,8)(4,10,5,6), (1,12)(2,11)
    (3,6)(4,8)(5,9)(7,10) ])
```

     First we show that for s of order 10, $P(S, s) = 31/99$ holds.

```
                                ─── Example ───
 gap> approx:= ApproxP( prim, spos );
 [ 0, 3/11, 1/3, 1/11, 1/132, 13/99, 13/99, 13/396, 1/132, 1/33, 1/33,
   1/33, 13/396, 0, 0 ]
 gap> 2B:= g.2^2;
 (3,11)(4,5)(6,10)(7,8)
 gap> Size( ConjugacyClass( g, 2B ) );
 495
 gap> ResetGlobalRandomNumberGenerators();
 gap> repeat s:= Random( g );
 >    until Order( s ) = 10;
 gap> prop:= RatioOfNongenerationTransPermGroup( g, 2B, s );
 31/99
 gap> Filtered( approx, x -> x >= prop );
 [ 1/3 ]
```

Next we show that for *s* of order different from 10, $P(g,s)$ is larger than $31/99$ for suitable $g \in S^{\times}$. Except for *s* in the class 6A (which fixes no point in the degree 12 representation), it suffices to consider *g* in the class 2B (with four fixed points).

```
                                ─── Example ───
 gap> x:= g.2^2;
 (3,11)(4,5)(6,10)(7,8)
 gap> ccl:= List( ConjugacyClasses( g ), Representative );;
 gap> SortParallel( List( ccl, Order ), ccl );
 gap> prop:= List( ccl, r -> RatioOfNongenerationTransPermGroup( g, x, r ) );;
 gap> SortedList( prop );
 [ 7/55, 31/99, 5/9, 5/9, 39/55, 383/495, 383/495, 43/55, 29/33, 1, 1,
   1, 1, 1, 1 ]
 gap> bad:= Filtered( prop, x -> x < 31/99 );
 [ 7/55 ]
 gap> pos:= Position( prop, bad[1] );;
 gap> [ Order( ccl[ pos ] ), NrMovedPoints( ccl[ pos ] ) ];
 [ 6, 12 ]
```

In the remaining case, we choose *g* in the class 2A (which is fixed point free).

```
                                ─── Example ───
 gap> x:= g.3;
 (1,12)(2,11)(3,6)(4,8)(5,9)(7,10)
 gap> s:= ccl[ pos ];;
 gap> prop:= RatioOfNongenerationTransPermGroup( g, x, s );
 17/33
 gap> prop > 31/99;
 true
```

Statement (e) has been shown already in Section 11.4.3.

11.5.11 $O_7(3)$

We show that $S = O_7(3)$ satisfies the following.

(a) $\sigma(S) = 199/351$, and this value is attained exactly for $\sigma(S,s)$ with *s* of order 14.

(b) For $s \in S$ of order 14, $\mathbb{M}(S,s)$ consists of one group of the type $2.U_4(3).2_2 = \Omega^-(6,3).2$ and two nonconjugate groups of the type S_9.

(c) $P(S) = 155/351$, and this value is attained exactly for $P(S,s)$ with s of order 14.

(d) The uniform spread of S is at least three, with s of order 14.

(e) $\sigma'(\mathrm{Aut}(S),s) = 1/3$.

Currently GAP provides neither the table of marks of S nor all character tables of its maximal subgroups. First we compute those primitive permutation characters of S that have the degrees 351 (point stabilizer $2.U_4(3).2_2$), 364 (point stabilizer $3^5 : U_4(2).2$), 378 (point stabilizer $L_4(3).2_2$), 1080 (point stabilizer $G_2(3)$, two classes), 1120 (point stabilizer $3^{3+3} : L_3(3)$), 3159 (point stabilizer $S_6(2)$, two classes), 12636 (point stabilizer S_9, two classes), 22113 (point stabilizer $(2^2 \times U_4(2)).2$, which extends to $D_8 \times U_4(2).2$ in $O_7(3).2$), and 28431 (point stabilizer $2^6 : A_7$).

(So we ignore the primitive permutation characters of the degrees 3640, 265356, and 331695. Note that the orders of the corresponding subgroups are not divisible by 7.)

```
                            ──── Example ────
  gap> t:= CharacterTable( "O7(3)" );;
  gap> someprim:= [];;
  gap> pi:= PossiblePermutationCharacters(
  >              CharacterTable( "2.U4(3).2_2" ), t );;  Length( pi );
  1
  gap> Append( someprim, pi );
  gap> pi:= PermChars( t, rec( torso:= [ 364 ] ) );;  Length( pi );
  1
  gap> Append( someprim, pi );
  gap> pi:= PossiblePermutationCharacters(
  >              CharacterTable( "L4(3).2_2" ), t );;  Length( pi );
  1
  gap> Append( someprim, pi );
  gap> pi:= PossiblePermutationCharacters( CharacterTable( "G2(3)" ), t );
  [ Character( CharacterTable( "O7(3)" ),
    [ 1080, 0, 0, 24, 108, 0, 0, 0, 27, 18, 9, 0, 12, 4, 0, 0, 0, 0, 0,
        0, 0, 0, 12, 0, 0, 0, 0, 0, 3, 6, 0, 3, 2, 2, 2, 0, 0, 0, 3, 0,
        0, 0, 0, 0, 0, 4, 0, 3, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0 ] ),
    Character( CharacterTable( "O7(3)" ),
    [ 1080, 0, 0, 24, 108, 0, 0, 27, 0, 18, 9, 0, 12, 4, 0, 0, 0, 0, 0,
        0, 0, 0, 12, 0, 0, 0, 0, 3, 0, 0, 6, 3, 2, 2, 2, 0, 0, 3, 0, 0,
        0, 0, 0, 0, 0, 4, 3, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0 ] ) ]
  gap> Append( someprim, pi );
  gap> pi:= PermChars( t, rec( torso:= [ 1120 ] ) );;  Length( pi );
  1
  gap> Append( someprim, pi );
  gap> pi:= PossiblePermutationCharacters( CharacterTable( "S6(2)" ), t );
  [ Character( CharacterTable( "O7(3)" ),
    [ 3159, 567, 135, 39, 0, 81, 0, 0, 27, 27, 0, 15, 3, 3, 7, 4, 0,
        27, 0, 0, 0, 0, 0, 9, 3, 0, 9, 0, 3, 9, 3, 0, 2, 1, 1, 0, 0, 0,
        3, 0, 2, 0, 0, 0, 3, 0, 0, 3, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0 ] ),
    Character( CharacterTable( "O7(3)" ),
    [ 3159, 567, 135, 39, 0, 81, 0, 27, 0, 27, 0, 15, 3, 3, 7, 4, 0,
        27, 0, 0, 0, 0, 0, 9, 3, 0, 9, 3, 0, 3, 9, 0, 2, 1, 1, 0, 0, 3,
```

```
        0, 0, 2, 0, 0, 0, 3, 0, 3, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0 ] ) ]
gap> Append( someprim, pi );
gap> pi:= PossiblePermutationCharacters( CharacterTable( "S9" ), t );
[ Character( CharacterTable( "O7(3)" ),
  [ 12636, 1296, 216, 84, 0, 81, 0, 0, 108, 27, 0, 6, 0, 12, 10, 1,
      0, 27, 0, 0, 0, 0, 0, 9, 3, 0, 9, 0, 12, 9, 3, 0, 1, 0, 2, 0,
      0, 0, 3, 1, 1, 0, 0, 0, 3, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0,
      1 ] ), Character( CharacterTable( "O7(3)" ),
  [ 12636, 1296, 216, 84, 0, 81, 0, 108, 0, 27, 0, 6, 0, 12, 10, 1,
      0, 27, 0, 0, 0, 0, 0, 9, 3, 0, 9, 12, 0, 3, 9, 0, 1, 0, 2, 0,
      0, 3, 0, 1, 1, 0, 0, 0, 3, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0,
      1 ] ) ]
gap> Append( someprim, pi );
gap> t2:= CharacterTable( "O7(3).2" );;
gap> s2:= CharacterTable( "Dihedral", 8 ) * CharacterTable( "U4(2).2" );
CharacterTable( "Dihedral(8)xU4(2).2" )
gap> pi:= PossiblePermutationCharacters( s2, t2 );;  Length( pi );
1
gap> pi:= RestrictedClassFunctions( pi, t );;
gap> Append( someprim, pi );
gap> pi:= PossiblePermutationCharacters(
>             CharacterTable( "2^6:A7" ), t );;  Length( pi );
1
gap> Append( someprim, pi );
gap> List( someprim, x -> x[1] );
[ 351, 364, 378, 1080, 1080, 1120, 3159, 3159, 12636, 12636, 22113,
  28431 ]
```

Note that in the three cases where two possible permutation characters were found, there are in fact two classes of subgroups that induce different permutation characters. For the subgroups of the types $G_2(3)$ and $S_6(2)$, this is stated in [CCN+85, p. 109], and for the subgroups of the type S_9, this follows from the fact that each S_9 type subgroup in S contains elements in exactly one of the classes 3D or 3E, and these two classes are fused by the outer automorphism of S.

```
───────────────── Example ─────────────────
gap> cl:= PositionsProperty( AtlasClassNames( t ),
>                            x -> x in [ "3D", "3E" ] );
[ 8, 9 ]
gap> List( Filtered( someprim, x -> x[1] = 12636 ), pi -> pi{ cl } );
[ [ 0, 108 ], [ 108, 0 ] ]
gap> GetFusionMap( t, t2 ){ cl };
[ 8, 8 ]
```

Now we compute the lower bounds for $\sigma(S, s')$ that are given by the sublist `someprim` of the primitive permutation characters.

```
───────────────── Example ─────────────────
gap> spos:= Position( OrdersClassRepresentatives( t ), 14 );
52
gap> Maximum( ApproxP( someprim, spos ) );
199/351
```

This shows that $\sigma(S,s) = 199/351$ holds. For statement (a), we have to show that choosing s' from another class than 14A yields a larger value for $\sigma(S,s')$.

```
_____ Example _____
gap> approx:= List( [ 1 .. NrConjugacyClasses( t ) ],
>       i -> Maximum( ApproxP( someprim, i ) ) );;
gap> PositionsProperty( approx, x -> x <= 199/351 );
[ 52 ]
```

Statement (b) can be read off from the permutation characters.

```
_____ Example _____
gap> pos:= PositionsProperty( someprim, x -> x[ spos ] <> 0 );
[ 1, 9, 10 ]
gap> List( someprim{ pos }, x -> x{ [ 1, spos ] } );
[ [ 351, 1 ], [ 12636, 1 ], [ 12636, 1 ] ]
```

For statement (c), we first compute $P(g,s)$ for g in the class 2A, via explicit computations with the group. For dealing with this case, we first construct a faithful permutation representation of $O_7(3)$ from the natural matrix representation of $SO(7,3)$.

```
_____ Example _____
gap> so73:= SpecialOrthogonalGroup( 7, 3 );;
gap> o73:= DerivedSubgroup( so73 );;
gap> orbs:= OrbitsDomain( o73, Elements( GF(3)^7 ) );;
gap> Set( orbs, Length );
[ 1, 702, 728, 756 ]
gap> g:= Action( o73, First( orbs, x -> Length( x ) = 702 ) );;
gap> Size( g ) = Size( t );
true
```

A 2A element g can be found as the 7-th power of any element of order 14 in S.

```
_____ Example _____
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g );
>     until Order( s ) = 14;
gap> 2A:= s^7;;
gap> bad:= RatioOfNongenerationTransPermGroup( g, 2A, s );
155/351
gap> bad > 1/3;
true
gap> approx:= ApproxP( someprim, spos );;
gap> PositionsProperty( approx, x -> x >= 1/3 );
[ 2 ]
```

This shows that $P(g,s) = 155/351 > 1/3$. Since $\sigma(g,s) < 1/3$ for all nonidentity g not in the class 2A, we have $P(S,s) = 155/351$. For statement (c), it remains to show that $P(S,s')$ is larger than $155/351$ whenever s' is not of order 14. First we compute $P(g,s')$, for g in the class 2A.

```
_____ Example _____
gap> consider:= RepresentativesMaximallyCyclicSubgroups( t );
[ 18, 19, 25, 26, 27, 30, 31, 32, 34, 35, 38, 39, 41, 42, 43, 44, 45,
  46, 47, 48, 49, 50, 52, 53, 54, 56, 57, 58 ]
```

```
gap> Length( consider );
28
gap> consider:= ClassesPerhapsCorrespondingToTableColumns( g, t, consider );;
gap> Length( consider );
31
gap> consider:= List( consider, Representative );;
gap> SortParallel( List( consider, Order ), consider );
gap> app2A:= List( consider, c ->
>       RatioOfNongenerationTransPermGroup( g, 2A, c ) );;
gap> SortedList( app2A );
[ 1/3, 1/3, 155/351, 191/351, 67/117, 23/39, 23/39, 85/117, 10/13,
  10/13, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1 ]
gap> test:= PositionsProperty( app2A, x -> x <= 155/351 );;
gap> List( test, i -> Order( consider[i] ) );
[ 13, 13, 14 ]
```

We see that only for s' in one of the two (algebraically conjugate) classes of element order 13, $P(S,s')$ has a chance to be smaller than 155/351. This possibility is now excluded by counting elements in the class 3A that do not generate S together with s' of order 13.

```
────────────────────────── Example ──────────────────────────
gap> C3A:= First( ConjugacyClasses( g ),
>                 c -> Order( Representative( c ) ) = 3 and Size( c ) = 7280 );;
gap> repeat ss:= Random( g );
>    until Order( ss ) = 13;
gap> bad:= RatioOfNongenerationTransPermGroup( g, Representative( C3A ), ss );
17/35
gap> bad > 155/351;
true
```

Now we show statement (d): For each triple (x_1, x_2, x_3) of nonidentity elements in S, there is an element s in the class 14A such that $\langle x_i, s \rangle = S$ holds for $1 \le i \le 3$. We can read off from the character-theoretic data that only those triples have to be checked for which at least two elements are contained in the class 2A, and the third element lies in one of the classes 2A, 2B, 3B.

```
────────────────────────── Example ──────────────────────────
gap> approx:= ApproxP( someprim, spos );;
gap> max:= Maximum( approx{ [ 3 .. Length( approx ) ] } );
59/351
gap> 155 + 2*59 < 351;
true
gap> third:= PositionsProperty( approx, x -> 2 * 155/351 + x >= 1 );
[ 2, 3, 6 ]
gap> ClassNames( t ){ third };
[ "2a", "2b", "3b" ]
```

We can find elements in the classes 2B and 3B as powers of arbitrary elements of the orders 20 and 15, respectively.

```
────────────────────────── Example ──────────────────────────
gap> ord20:= PositionsProperty( OrdersClassRepresentatives( t ),
>                               x -> x = 20 );
```

```
[ 58 ]
gap> PowerMap( t, 10 ){ ord20 };
[ 3 ]
gap> repeat x:= Random( g );
>     until Order( x ) = 20;
gap> 2B:= x^10;;
gap> C2B:= ConjugacyClass( g, 2B );;
gap> ord15:= PositionsProperty( OrdersClassRepresentatives( t ),
>                               x -> x = 15 );
[ 53 ]
gap> PowerMap( t, 10 ){ ord15 };
[ 6 ]
gap> repeat x:= Random( g );
>     until Order( x ) = 15;
gap> 3B:= x^5;;
gap> C3B:= ConjugacyClass( g, 3B );;
```

The existence of *s* can be shown with the random approach described in Section 11.3.3.

```
———————————————————— Example ————————————————————
gap> repeat s:= Random( g );
>     until Order( s ) = 14;
gap> RandomCheckUniformSpread( g, [ 2A, 2A, 2A ], s, 50 );
true
gap> RandomCheckUniformSpread( g, [ 2B, 2A, 2A ], s, 50 );
true
gap> RandomCheckUniformSpread( g, [ 3B, 2A, 2A ], s, 50 );
true
```

Finally, we show statement (e). Let $G = \mathrm{Aut}(S) = S.2$. By [CCN+85, p. 109], $\mathbb{M}'(G, s)$ consists of the extension of the $2.U_4(3).2_1$ type subgroup. We compute the extension of the permutation character.

```
———————————————————— Example ————————————————————
gap> prim:= someprim{ [ 1 ] };
[ Character( CharacterTable( "O7(3)" ),
  [ 351, 127, 47, 15, 27, 45, 36, 0, 0, 9, 0, 15, 3, 3, 7, 6, 19, 19,
      10, 11, 12, 8, 3, 5, 3, 6, 1, 0, 0, 3, 3, 0, 1, 1, 1, 6, 3, 0,
      0, 2, 2, 0, 3, 0, 3, 3, 0, 0, 1, 0, 0, 1, 0, 4, 4, 1, 2, 0 ] ) ]
gap> spos:= Position( AtlasClassNames( t ), "14A" );;
gap> t2:= CharacterTable( "O7(3).2" );;
gap> map:= InverseMap( GetFusionMap( t, t2 ) );;
gap> torso:= List( prim, pi -> CompositionMaps( pi, map ) );;
gap> ext:= List( torso, x -> PermChars( t2, rec( torso:= x ) ) );
[ [ Character( CharacterTable( "O7(3).2" ),
      [ 351, 127, 47, 15, 27, 45, 36, 0, 9, 0, 15, 3, 3, 7, 6, 19,
          19, 10, 11, 12, 8, 3, 5, 3, 6, 1, 0, 3, 0, 1, 1, 1, 6, 3,
          0, 2, 2, 0, 3, 0, 3, 3, 0, 1, 0, 0, 1, 0, 4, 1, 2, 0, 117,
          37, 21, 45, 1, 13, 5, 1, 9, 9, 18, 15, 1, 7, 9, 6, 4, 0, 3,
          0, 3, 3, 6, 2, 2, 9, 6, 1, 3, 1, 4, 1, 2, 1, 1, 0, 3, 1, 0,
          0, 0, 0, 1, 1, 0, 0 ] ) ] ] ]
gap> approx:= ApproxP( Concatenation( ext ),
>       Position( AtlasClassNames( t2 ), "14A" ) );;
gap> Maximum( approx{ Difference(
```

```
  >          PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
  >          ClassPositionsOfDerivedSubgroup( t2 ) ) } );
  1/3
```

11.5.12 $O_8^+(2)$

We show that $S = O_8^+(2) = \Omega^+(8,2)$ satisfies the following.

(a) $\sigma(S) = 334/315$, and this value is attained exactly for $\sigma(S,s)$ with s of order 15.

(b) For $s \in S$ of order 15, $\mathbb{M}(S,s)$ consists of one group of the type $S_6(2)$, two conjugate groups of the type $2^6 : A_8$, two conjugate groups of the type A_9, and one group of each of the types $(3 \times U_4(2)) : 2 = (3 \times \Omega^-(6,2)) : 2$ and $(A_5 \times A_5) : 2^2 = (\Omega^-(4,2) \times \Omega^-(4,2)) : 2^2$.

(c) $P(S) = 29/42$, and this value is attained exactly for $P(S,s)$ with s of order 15.

(d) Let $x,y \in S$ such that x,y,xy lie in the unique involution class of length $1\,575$ of S. (This is the class 2A.) Then each element in S together with one of x, y, xy generates a proper subgroup of S.

(e) Both the spread and the uniform spread of S is exactly two, with s of order 15.

(f) For each choice of $s \in S$, there is an extension $S.2$ such that for any element g in the (outer) class 2F, $\langle s,g \rangle$ does not contain S.

(g) For an element s of order 15 in S, either S is the only maximal subgroup of $S.2$ that contains s, or the maximal subgroups of $S.2$ that contain s are S and the extensions of the subgroups listed in statement (b); these groups have the structures $S_6(2) \times 2$, $2^6 : S_8$ (twice), S_9 (twice), $S_3 \times U_4(2).2$, and $S_5 \wr 2$.

(h) For $s \in S$ of order 15 and arbitrary $g \in S.3 \setminus S$, we have $\langle s,g \rangle = S.3$.

(i) If x, y are nonidentity elements in $\mathrm{Aut}(S)$ then there is an element s of order 15 in S such that $S \subseteq \langle x,s \rangle \cap \langle y,s \rangle$.

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

```
───────────────────────────── Example ─────────────────────────────
gap> t:= CharacterTable( "O8+(2)" );;
gap> ProbGenInfoSimple( t );
[ "O8+(2)", 334/315, 0, [ "15A", "15B", "15C" ], [ 7, 7, 7 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the only classes of maximal subgroups that contain elements of order 15 consist of groups of the structures as claimed, see [CCN+85, p. 85].

```
───────────────────────────── Example ─────────────────────────────
gap> prim:= PrimitivePermutationCharacters( t );;
gap> spos:= Position( OrdersClassRepresentatives( t ), 15 );;
gap> List( Filtered( prim, x -> x[ spos ] <> 0 ), l -> l{ [ 1, spos ] } );
[ [ 120, 1 ], [ 135, 2 ], [ 960, 2 ], [ 1120, 1 ], [ 12096, 1 ] ]
```

For the remaining statements, we take a primitive permutation representation on 120 points, and assume that the permutation character is 1a+35a+84a. (See [CCN+85, p. 85], note that the three classes of maximal subgroups of index 120 in *S* are conjugate under triality.)

```
─────────────────────── Example ───────────────────────
gap> matgroup:= DerivedSubgroup( GeneralOrthogonalGroup( 1, 8, 2 ) );;
gap> points:= NormedRowVectors( GF(2)^8 );;
gap> orbs:= OrbitsDomain( matgroup, points );;
gap> List( orbs, Length );
[ 135, 120 ]
gap> g:= Action( matgroup, orbs[2] );;
gap> Size( g );
174182400
gap> pi:= Sum( Irr( t ){ [ 1, 3, 7 ] } );
Character( CharacterTable( "O8+(2)" ),
 [ 120, 24, 32, 0, 0, 8, 36, 0, 0, 3, 6, 12, 4, 8, 0, 0, 0, 10, 0, 0,
   12, 0, 0, 8, 0, 0, 3, 6, 0, 0, 2, 0, 0, 2, 1, 2, 2, 3, 0, 0, 2, 0,
   0, 0, 0, 0, 3, 2, 0, 0, 1, 0, 0 ] )
```

In order to show statement (c), we first observe that for *s* in the class 15A and *g not* in one of the classes 2A, 2B, 3A, $\sigma(g,s) < 1/3$ holds, and for the exceptional three classes, we have $\sigma(g,s) > 1/2$.

```
─────────────────────── Example ───────────────────────
gap> approx:= ApproxP( prim, spos );;
gap> testpos:= PositionsProperty( approx, x -> x >= 1/3 );
[ 2, 3, 7 ]
gap> AtlasClassNames( t ){ testpos };
[ "2A", "2B", "3A" ]
gap> approx{ testpos };
[ 254/315, 334/315, 1093/1120 ]
gap> ForAll( approx{ testpos }, x -> x > 1/2 );
true
```

Now we compute the values $P(g,s)$, for *s* in the class 15A and *g* in one of the classes 2A, 2B, 3A.

By our choice of the character of the permutation representation we use, the class 15A is determined as the unique class of element order 15 with one fixed point. (Note that the three classes of element order 15 in *S* are conjugate under triality.) A 2A element can be found as the fourth power of any element of order 8 in *S*, a 3A element can be found as the fifth power of a 15A element, and a 2B element can be found as the sixth power of an element of order 12, with 32 fixed points.

```
─────────────────────── Example ───────────────────────
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g );
>    until Order( s ) = 15 and NrMovedPoints( g ) = 1 + NrMovedPoints( s );
gap> 3A:= s^5;;
gap> repeat x:= Random( g ); until Order( x ) = 8;
gap> 2A:= x^4;;
gap> repeat x:= Random( g ); until Order( x ) = 12 and
>       NrMovedPoints( g ) = 32 + NrMovedPoints( x^6 );
gap> 2B:= x^6;;
gap> prop15A:= List( [ 2A, 2B, 3A ],
>                    x -> RatioOfNongenerationTransPermGroup( g, x, s ) );
[ 23/35, 29/42, 149/224 ]
```

```
gap> Maximum( prop15A );
29/42
```

This means that for *s* in the class 15A, we have $P(S, s) = 29/42$, and the same holds for all *s* of order 15 since the three classes of element order 15 are conjugate under triality. Now we show that for *s* of order different from 15, the value $P(g, s)$ is larger than 29/42, for *g* in one of the classes 2A, 2B, 3A, or their images under triality. This implies statement (c).

```
                              ──── Example ────
gap> test:= List( [ 2A, 2B, 3A ], x -> ConjugacyClass( g, x ) );;
gap> ccl:= ConjugacyClasses( g );;
gap> consider:= Filtered( ccl, c -> Size( c ) in List( test, Size ) );;
gap> Length( consider );
7
gap> filt:= Filtered( ccl, c -> ForAll( consider, cc ->
>        RatioOfNongenerationTransPermGroup( g, Representative( cc ),
>            Representative( c ) ) <= 29/42 ) );;
gap> Length( filt );
3
gap> List( filt, c -> Order( Representative( c ) ) );
[ 15, 15, 15 ]
```

Now we show statement (d). First we observe that all those Klein four groups in *S* whose involutions lie in the class 2A are conjugate in *S*. Note that this is the unique class of length 1575 in *S*, and also the unique class whose elements have 24 fixed points in the degree 120 permutation representation.

For that, we use the character table of *S* to read off that *S* contains exactly 14175 such subgroups, and we use the group to compute one such subgroup and its normalizer of index 14175.

```
                              ──── Example ────
gap> SizesConjugacyClasses( t );
[ 1, 1575, 3780, 3780, 3780, 56700, 2240, 2240, 2240, 89600, 268800,
  37800, 340200, 907200, 907200, 907200, 2721600, 580608, 580608,
  580608, 100800, 100800, 100800, 604800, 604800, 604800, 806400,
  806400, 806400, 806400, 2419200, 2419200, 2419200, 7257600,
  24883200, 5443200, 5443200, 6451200, 6451200, 6451200, 8709120,
  8709120, 8709120, 1209600, 1209600, 1209600, 4838400, 7257600,
  7257600, 7257600, 11612160, 11612160, 11612160 ]
gap> NrPolyhedralSubgroups( t, 2, 2, 2 );
rec( number := 14175, type := "V4" )
gap> repeat x:= Random( g );
>    until    Order( x ) mod 2 = 0
>         and NrMovedPoints( x^( Order(x)/2 ) ) = 120 - 24;
gap> x:= x^( Order(x)/2 );;
gap> repeat y:= x^Random( g );
>    until NrMovedPoints( x*y ) = 120 - 24;
gap> v4:= SubgroupNC( g, [ x, y ] );;
gap> n:= Normalizer( g, v4 );;
gap> Index( g, n );
14175
```

We verify that the triple has the required property.

```
───────────────────────────── Example ─────────────────────────────
gap> maxorder:= RepresentativesMaximallyCyclicSubgroups( t );;
gap> maxorderreps:= List( ClassesPerhapsCorrespondingToTableColumns( g, t,
>        maxorder ), Representative );;
gap> Length( maxorderreps );
28
gap> CommonGeneratorWithGivenElements( g, maxorderreps, [ x, y, x*y ] );
fail
```

For the simple group *S*, it remains to show statement (e). We want to show that for any choice of two nonidentity elements *x*, *y* in *S*, there is an element *s* in the class 15A such that $\langle s, x \rangle = \langle s, y \rangle = S$ holds. Only *x*, *y* in the classes given by the list `testpos` must be considered, by the estimates $\sigma(g, s)$.

We replace the values $\sigma(g, s)$ by the exact values $P(g, s)$, for *g* in one of these three classes. Each of the three classes is determined by its element order and its number of fixed points.

```
───────────────────────────── Example ─────────────────────────────
gap> reps:= List( ccl, Representative );;
gap> bading:= List( testpos, i -> Filtered( reps,
>        r -> Order( r ) = OrdersClassRepresentatives( t )[i] and
>           NrMovedPoints( r ) = 120 - pi[i] ) );;
gap> List( bading, Length );
[ 1, 1, 1 ]
gap> bading:= List( bading, x -> x[1] );;
```

For each pair (C_1, C_2) of classes represented by this list, we have to show that for any choice of elements $x \in C_1, y \in C_2$ there is *s* in the class 15A such that $\langle s, x \rangle = \langle s, y \rangle = S$ holds. This is done with the random approach that is described in Section 11.3.3.

```
───────────────────────────── Example ─────────────────────────────
gap> for pair in UnorderedTuples( bading, 2 ) do
>        test:= RandomCheckUniformSpread( g, pair, s, 80 );
>        if test <> true then
>          Error( test );
>        fi;
>     od;
```

We get no error message, so statement (e) holds.

Now we turn to the automorphic extensions of *S*. First we compute a permutation representation of $\mathrm{SO}^+(8, 2) \cong S.2$ and an element *g* in the class 2F, which is the unique conjugacy class of size 120 in *S*.2.

```
───────────────────────────── Example ─────────────────────────────
gap> matgrp:= SO(1,8,2);;
gap> g2:= Image( IsomorphismPermGroup( matgrp ) );;
gap> IsTransitive( g2, MovedPoints( g2 ) );
true
gap> repeat x:= Random( g2 ); until Order( x ) = 14;
gap> 2F:= x^7;;
gap> Size( ConjugacyClass( g2, 2F ) );
120
```

Only for *s* in six conjugacy classes of *S*, there is a nonzero probability to have $S.2 = \langle g, s \rangle$.

```
───────────────── Example ─────────────────
gap> der:= DerivedSubgroup( g2 );;
gap> cclreps:= List( ConjugacyClasses( der ), Representative );;
gap> nongen:= List( cclreps,
>               x -> RatioOfNongenerationTransPermGroup( g2, 2F, x ) );;
gap> goodpos:= PositionsProperty( nongen, x -> x < 1 );;
gap> invariants:= List( goodpos, i -> [ Order( cclreps[i] ),
>        Size( Centralizer( g2, cclreps[i] ) ), nongen[i] ] );;
gap> SortedList( invariants );
[ [ 10, 20, 1/3 ], [ 10, 20, 1/3 ], [ 12, 24, 2/5 ], [ 12, 24, 2/5 ],
  [ 15, 15, 0 ], [ 15, 15, 0 ] ]
```

S contains three classes of element order 10, which are conjugate in $S.3$. For a fixed extension of the type $S.2$, the element s can be chosen only in two of these three classes, which means that there is another group of the type $S.2$ (more precisely, another subgroup of index three in $S.S_3$) in which this choice of s is not suitable –note that the general aim is to find $s \in S$ uniformly for all automorphic extensions of S. Analogous statements hold for the other possibilities for s, so statement (f) follows.

Statement (g) follows from the list of maximal subgroups in [CCN+85, p. 85].

Statement (h) follows from the fact that S is the only maximal subgroup of $S.3$ that contains elements of order 15, according to the list of maximal subgroups in [CCN+85, p. 85]. Alternatively, if we do not want to assume this information, we can use explicit computations, as follows. All we have to check is that any element in the classes 3F and 3G generates $S.3$ together with a fixed element of order 15 in S.

We compute a permutation representation of $S.3$ as the derived subgroup of a subgroup of the type $S.S_3$ inside the sporadic simple Fischer group Fi_{22}; these subgroups lie in the fourth class of maximal subgroups of Fi_{22}, see [CCN+85, p. 163]. An element in the class 3F of $S.3$ can be found as a power of an order 21 element, and an element in the class 3G can be found as the fourth power of a 12P element.

```
───────────────── Example ─────────────────
gap> aut:= Group( AtlasGenerators( "Fi22", 1, 4 ).generators );;
gap> Size( aut ) = 6 * Size( t );
true
gap> g3:= DerivedSubgroup( aut );;
gap> orbs:= OrbitsDomain( g3, MovedPoints( g3 ) );;
gap> List( orbs, Length );
[ 3150, 360 ]
gap> g3:= Action( g3, orbs[2] );;
gap> repeat s:= Random( g3 ); until Order( s ) = 15;
gap> repeat x:= Random( g3 ); until Order( x ) = 21;
gap> 3F:= x^7;;
gap> RatioOfNongenerationTransPermGroup( g3, 3F, s );
0
gap> repeat x:= Random( g3 );
>    until Order( x ) = 12 and Size( Centralizer( g3, x^4 ) ) = 648;
gap> 3G:= x^4;;
gap> RatioOfNongenerationTransPermGroup( g3, 3G, s );
0
```

Finally, consider statement (i). It implies that [BGK08, Corollary 1.5] holds for $\Omega^+(8, 2)$, with s of order 15. Note that by part (f), *s cannot be chosen in a prescribed conjugacy class* of S that is

independent of the elements *x*, *y*.

If *x* and *y* lie in *S* then statement (i) follows from part (e), and by part (g), the case that *x* or *y* lie in *S*.3 \ *S* is also not a problem. We now show that also *x* or *y* in *S*.2 \ *S* is not a problem. Here we have to deal with the cases that *x* and *y* lie in the same subgroup of index 3 in Aut(*S*) or in different such subgroups. Actually we show that for each index 3 subgroup $H = S.2 <$ Aut(*S*), we can choose *s* from two of the three classes of element order 15 in *S* such that *S* is the only maximal subgroup of *H* that contains *s*, and thus $\langle x, s \rangle$ contains *H*, for any choice of $x \in H \setminus S$.

For that, we note that no novelty in *S*.2 contains elements of order 15, so all maximal subgroups of *S*.2 that contain such elements –besides *S*– have one of the indices 120, 135, 960, 1120, or 12096, and point stabilizers of the types $S_6(2) \times 2$, $2^6 : S_8$, S_9, $S_3 \times U_4(2) : 2$, or $S_5 \wr 2$. We compute the corresponding permutation characters.

```
─────────────────────────── Example ───────────────────────────
  gap> t2:= CharacterTable( "O8+(2).2" );;
  gap> s:= CharacterTable( "S6(2)" ) * CharacterTable( "Cyclic", 2 );;
  gap> pi:= PossiblePermutationCharacters( s, t2 );;
  gap> prim:= pi;;
  gap> pi:= PermChars( t2, rec( torso:= [ 135 ] ) );;
  gap> Append( prim, pi );
  gap> pi:= PossiblePermutationCharacters( CharacterTable( "A9.2" ), t2 );;
  gap> Append( prim, pi );
  gap> s:= CharacterTable( "Dihedral(6)" ) * CharacterTable( "U4(2).2" );;
  gap> pi:= PossiblePermutationCharacters( s, t2 );;
  gap> Append( prim, pi );
  gap> s:= CharacterTableWreathSymmetric( CharacterTable( "S5" ), 2 );;
  gap> pi:= PossiblePermutationCharacters( s, t2 );;
  gap> Append( prim, pi );
  gap> Length( prim );
  5
  gap> ord15:= PositionsProperty( OrdersClassRepresentatives( t2 ),
  >                               x -> x = 15 );
  [ 39, 40 ]
  gap> List( prim, pi -> pi{ ord15 } );
  [ [ 1, 0 ], [ 2, 0 ], [ 2, 0 ], [ 1, 0 ], [ 1, 0 ] ]
  gap> List( ord15, i -> Maximum( ApproxP( prim, i ) ) );
  [ 307/120, 0 ]
```

Here it is appropriate to clean the workspace again.

```
─────────────────────────── Example ───────────────────────────
  gap> CleanWorkspace();
```

11.5.13 $O_8^+(3)$

We show that $S = O_8^+(3)$ satisfies the following.

(a) $\sigma(S) = 863/1820$, and this value is attained exactly for $\sigma(S, s)$ with *s* of order 20.

(b)   For $s \in S$ of order 20, $\mathbb{M}(S, s)$ consists of two nonconjugate groups of the type $O_7(3) = \Omega(7,3)$, two conjugate subgroups of the type $3^6 : L_4(3)$, two nonconjugate subgroups of the type $(A_4 \times U_4(2)) : 2$, and one subgroup of each of the types $2.U_4(3).(2^2)_{122}$ and $(A_6 \times A_6) : 2^2$.

(c) $P(S) = 194/455$, and this value is attained exactly for $P(S, s)$ with s of order 20.

(d) The uniform spread of S is at least three, with s of order 20.

(e) The preimage of s in the matrix group $2.S = \Omega^+(8,3)$ can be chosen of order 40, and then the maximal subgroups of $2.S$ containing s have the structures $2.O_7(3)$, $3^6 : 2.L_4(3)$, $4.U_4(3).2^2 = \mathrm{SU}(4,3).2^2$, $2.(A_4 \times U_4(2)).2 = 2.(\mathrm{PSp}(2,3) \otimes \mathrm{PSp}(4,3)).2$, and $2.(A_6 \times A_6) : 2^2 = 2.(\Omega^-(4,3) \times \Omega^-(4,3)) : 2^2$, respectively.

(f) For $s \in S$ of order 20, we have $P'(S.2_1, s) \in \{83/567, 574/1215\}$, $P'(S.2_2, s) \in \{0, 1\}$ (depending on the choice of s), and $\sigma'(S.3, s) = 0$.

Furthermore, for any choice of $s' \in S$, we have $\sigma'(S.2_2, s') = 1$ for some group $S.2_2$. However, if it is allowed to choose s from an $\mathrm{Aut}(S)$-class of elements of order 20 (and not from a fixed S-class) then we can achieve $\sigma(g, s) = 0$ for any given $g \in S.2_2 \setminus S$.

(g) The maximal subgroups of $S.2_1$ that contain an element of order 20 are either S and the extensions of the subgroups listed in statement (b) or they are S and $L_4(3).2^2$, $3^6 : L_4(3).2$ (twice), $2.U_4(3).(2^2)_{122}.2$, and $(A_6 \times A_6) : 2^2.2$.

In the former case, the groups have the structures $O_7(3) : 2$ (twice), $3^6 : (L_4(3) \times 2)$ (twice), $S_4 \times U_4(2).2$ (twice), $2.U_4(3).(2^2)_{122}.2$, and $(A_6 \times A_6) : 2^2 \times 2$.

Statement (a) follows from inspection of the primitive permutation characters.

```
──────────────────── Example ────────────────────
gap> t:= CharacterTable( "O8+(3)" );;
gap> ProbGenInfoSimple( t );
[ "O8+(3)", 863/1820, 2, [ "20A", "20B", "20C" ], [ 8, 8, 8 ] ]
```

Also statement (b) follows from the information provided by the character table of S (cf. [CCN$^+$85, p. 140]).

```
──────────────────── Example ────────────────────
gap> prim:= PrimitivePermutationCharacters( t );;
gap> ord:= OrdersClassRepresentatives( t );;
gap> spos:= Position( ord, 20 );;
gap> filt:= PositionsProperty( prim, x -> x[ spos ] <> 0 );
[ 1, 2, 7, 15, 18, 19, 24 ]
gap> Maxes( t ){ filt };
[ "O7(3)", "O8+(3)M2", "3^6:L4(3)", "2.U4(3).(2^2)_{122}",
  "(A4xU4(2)):2", "O8+(3)M19", "(A6xA6):2^2" ]
gap> prim{ filt }{ [ 1, spos ] };
[ [ 1080, 1 ], [ 1080, 1 ], [ 1120, 2 ], [ 189540, 1 ],
  [ 7960680, 1 ], [ 7960680, 1 ], [ 9552816, 1 ] ]
```

For statement (c), we first show that $P(S, s) = 194/455$ holds. Since this value is larger than $1/3$, we have to inspect only those classes g^S for which $\sigma(g, s) \geq 1/3$ holds,

```
──────────────────── Example ────────────────────
gap> ord:= OrdersClassRepresentatives( t );;
gap> ord20:= PositionsProperty( ord, x -> x = 20 );;
gap> cand:= [];;
gap> for i in ord20 do
```

```
>        approx:= ApproxP( prim, i );
>        Add( cand, PositionsProperty( approx, x -> x >= 1/3 ) );
>     od;
gap> cand;
[ [ 2, 6, 7, 10 ], [ 3, 6, 8, 11 ], [ 4, 6, 9, 12 ] ]
gap> AtlasClassNames( t ){ cand[1] };
[ "2A", "3A", "3B", "3E" ]
```

The three possibilities form one orbit under the outer automorphism group of *S*.

```
——————————————————— Example ———————————————————
gap> t3:= CharacterTable( "O8+(3).3" );;
gap> tfust3:= GetFusionMap( t, t3 );;
gap> List( cand, x -> tfust3{ x } );
[ [ 2, 4, 5, 6 ], [ 2, 4, 5, 6 ], [ 2, 4, 5, 6 ] ]
```

By symmetry, we may consider only the first possibility, and assume that *s* is in the class 20A.

We work with a permutation representation of degree 1080, and assume that the permutation character is 1a+260a+819a. (Note that all permutation characters of *S* of degree 1080 are conjugate under Aut(*S*).)

```
——————————————————— Example ———————————————————
gap> g:= Action( SO(1,8,3), NormedRowVectors( GF(3)^8 ), OnLines );;
gap> Size( g );
9904359628800
gap> g:= DerivedSubgroup( g );;  Size( g );
4952179814400
gap> orbs:= OrbitsDomain( g, MovedPoints( g ) );;
gap> List( orbs, Length );
[ 1080, 1080, 1120 ]
gap> g:= Action( g, orbs[1] );;
gap> PositionProperty( Irr( t ), chi -> chi[1] = 819 );
9
gap> permchar:= Sum( Irr( t ){ [ 1, 2, 9 ] } );
Character( CharacterTable( "O8+(3)" ),
 [ 1080, 128, 0, 0, 24, 108, 135, 0, 0, 108, 0, 0, 27, 27, 0, 0, 18,
   9, 12, 16, 0, 0, 4, 15, 0, 0, 20, 0, 0, 12, 11, 0, 0, 20, 0, 0, 15,
   0, 0, 12, 0, 0, 2, 0, 0, 3, 3, 0, 0, 6, 6, 0, 0, 3, 2, 2, 2, 18, 0,
   0, 9, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 3, 0, 0, 12, 0, 0, 3, 0, 0, 0,
   0, 0, 4, 3, 3, 0, 0, 1, 0, 0, 4, 0, 0, 1, 1, 2, 0, 0, 0, 0, 0, 3,
   0, 0, 2, 0, 0, 5, 0, 0, 1, 0, 0 ] )
```

Now we show that for *s* in the class 20A (which fixes one point), the proportion of nongenerating elements *g* in one of the classes 2A, 3A, 3B, 3E has the maximum 194/455, which is attained exactly for 3A. (We find a 2A element as a power of *s*, a 3A element as a power of any element of order 18, a 3B and a 3E element as elements with 135 and 108 fixed points, respectively, which occur as powers of suitable elements of order 15.)

```
——————————————————— Example ———————————————————
gap> permchar{ ord20 };
[ 1, 0, 0 ]
gap> AtlasClassNames( t )[ PowerMap( t, 10 )[ ord20[1] ] ];
```

```
 "2A"
gap> ord18:= PositionsProperty( ord, x -> x = 18 );;
gap> Set( AtlasClassNames( t ){ PowerMap( t, 6 ){ ord18 } } );
[ "3A" ]
gap> ord15:= PositionsProperty( ord, x -> x = 15 );;
gap> PowerMap( t, 5 ){ ord15 };
[ 7, 8, 9, 10, 11, 12 ]
gap> AtlasClassNames( t ){ [ 7 .. 12 ] };
[ "3B", "3C", "3D", "3E", "3F", "3G" ]
gap> permchar{ [ 7 .. 12 ] };
[ 135, 0, 0, 108, 0, 0 ]
gap> mp:= NrMovedPoints( g );;
gap> ResetGlobalRandomNumberGenerators();
gap> repeat 20A:= Random( g );
>    until Order( 20A ) = 20 and mp - NrMovedPoints( 20A ) = 1;
gap> 2A:= 20A^10;;
gap> repeat x:= Random( g ); until Order( x ) = 18;
gap> 3A:= x^6;;
gap> repeat x:= Random( g );
>    until Order( x ) = 15 and mp - NrMovedPoints( x^5 ) = 135;
gap> 3B:= x^5;;
gap> repeat x:= Random( g );
>    until Order( x ) = 15 and mp - NrMovedPoints( x^5 ) = 108;
gap> 3E:= x^5;;
gap> nongen:= List( [ 2A, 3A, 3B, 3E ],
>                   c -> RatioOfNongenerationTransPermGroup( g, c, 20A ) );
[ 3901/9477, 194/455, 451/1092, 451/1092 ]
gap> Maximum( nongen );
194/455
```

Next we compute the values $P(g,s)$, for g is in the class 3A and certain elements s. It is enough to consider representatives s of maximally cyclic subgroups in S, but here we can do better, as follows. Since 3A is the unique class of length 72800, it is fixed under $\mathrm{Aut}(S)$, so it is enough to consider one element s from each $\mathrm{Aut}(S)$-orbit on the classes of S. We use the class fusion between the character tables of S and $\mathrm{Aut}(S)$ for computing orbit representatives.

```
                         ── Example ──
gap> maxorder:= RepresentativesMaximallyCyclicSubgroups( t );;
gap> Length( maxorder );
57
gap> autt:= CharacterTable( "O8+(3).S4" );;
gap> fus:= PossibleClassFusions( t, autt );;
gap> orbreps:= Set( fus, map -> Set( ProjectionMap( map ) ) );
[ [ 1, 2, 5, 6, 7, 13, 17, 18, 19, 20, 23, 24, 27, 30, 31, 37, 43,
      46, 50, 54, 55, 56, 57, 58, 64, 68, 72, 75, 78, 84, 85, 89, 95,
      96, 97, 100, 106, 112 ] ]
gap> totest:= Intersection( maxorder, orbreps[1] );
[ 43, 50, 54, 56, 57, 64, 68, 75, 78, 84, 85, 89, 95, 97, 100, 106,
  112 ]
gap> Length( totest );
17
gap> AtlasClassNames( t ){ totest };
```

```
  [ "6Q", "6X", "6B1", "8A", "8B", "9G", "9K", "12A", "12D", "12J",
    "12K", "12O", "13A", "14A", "15A", "18A", "20A" ]
```

This means that we have to test one element of each of the element orders 13, 14, 15, and 18 (note that we know already a bound for elements of order 20), plus certain elements of the orders 6, 8, 9, and 12 which can be identified by their centralizer orders and (for elements of order 6 and 8) perhaps the centralizer orders of some powers.

```
——————————————— Example ———————————————
  gap> elementstotest:= [];;
  gap> for elord in [ 13, 14, 15, 18 ] do
  >        repeat s:= Random( g );
  >        until Order( s ) = elord;
  >        Add( elementstotest, s );
  >    od;
```

The next elements to be tested are in the classes 6B1 (centralizer order 162), in one of 9G–9J (centralizer order 729), in one of 9K–9N (centralizer order 81), in one of 12A–12C (centralizer order 1728), in one of 12D–12I (centralizer order 432), in 12J (centralizer order 192), in one of 12K–12N (centralizer order 108), and in one of 12O–12T (centralizer order 72).

```
——————————————— Example ———————————————
  gap> ordcent:= [ [ 6, 162 ], [ 9, 729 ], [ 9, 81 ], [ 12, 1728 ],
  >                [ 12, 432 ], [ 12, 192 ], [ 12, 108 ], [ 12, 72 ] ];;
  gap> cents:= SizesCentralizers( t );;
  gap> for pair in ordcent do
  >        Print( pair, ": ", AtlasClassNames( t ){
  >            Filtered( [ 1 .. Length( ord ) ],
  >                      i -> ord[i] = pair[1] and cents[i] = pair[2] ) }, "\n" );
  >        repeat s:= Random( g );
  >        until Order( s ) = pair[1] and Size( Centralizer( g, s ) ) = pair[2];
  >        Add( elementstotest, s );
  >    od;
  [ 6, 162 ]: [ "6B1" ]
  [ 9, 729 ]: [ "9G", "9H", "9I", "9J" ]
  [ 9, 81 ]: [ "9K", "9L", "9M", "9N" ]
  [ 12, 1728 ]: [ "12A", "12B", "12C" ]
  [ 12, 432 ]: [ "12D", "12E", "12F", "12G", "12H", "12I" ]
  [ 12, 192 ]: [ "12J" ]
  [ 12, 108 ]: [ "12K", "12L", "12M", "12N" ]
  [ 12, 72 ]: [ "12O", "12P", "12Q", "12R", "12S", "12T" ]
```

The next elements to be tested are in one of the classes 6Q–6S (centralizer order 648).

```
——————————————— Example ———————————————
  gap> AtlasClassNames( t ){ Filtered( [ 1 .. Length( ord ) ],
  >        i -> cents[i] = 648 and cents[ PowerMap( t, 2 )[i] ] = 52488
  >                             and cents[ PowerMap( t, 3 )[i] ] = 26127360 ) };
  [ "6Q", "6R", "6S" ]
  gap> repeat s:= Random( g );
  >    until Order( s ) = 6 and Size( Centralizer( g, s ) ) = 648
  >        and Size( Centralizer( g, s^2 ) ) = 52488
  >        and Size( Centralizer( g, s^3 ) ) = 26127360;
  gap> Add( elementstotest, s );
```

The next elements to be tested are in the class 6X–6A1 (centralizer order 648).

```
————————————— Example —————————————
gap> AtlasClassNames( t ){ Filtered( [ 1 .. Length( ord ) ],
>        i -> cents[i] = 648 and cents[ PowerMap( t, 2 )[i] ] = 52488
>                         and cents[ PowerMap( t, 3 )[i] ] = 331776 ) };
[ "6X", "6Y", "6Z", "6A1" ]
gap> repeat s:= Random( g );
>    until Order( s ) = 6 and Size( Centralizer( g, s ) ) = 648
>      and Size( Centralizer( g, s^2 ) ) = 52488
>      and Size( Centralizer( g, s^3 ) ) = 331776;
gap> Add( elementstotest, s );
```

Finally, we add elements from the classes 8A and 8B.

```
————————————— Example —————————————
gap> AtlasClassNames( t ){ Filtered( [ 1 .. Length( ord ) ],
>        i -> ord[i] = 8 and cents[ PowerMap( t, 2 )[i] ] = 13824 ) };
[ "8A" ]
gap> repeat s:= Random( g );
>    until Order( s ) = 8 and Size( Centralizer( g, s^2 ) ) = 13824;
gap> Add( elementstotest, s );
gap> AtlasClassNames( t ){ Filtered( [ 1 .. Length( ord ) ],
>        i -> ord[i] = 8 and cents[ PowerMap( t, 2 )[i] ] = 1536 ) };
[ "8B" ]
gap> repeat s:= Random( g );
>    until Order( s ) = 8 and Size( Centralizer( g, s^2 ) ) = 1536;
gap> Add( elementstotest, s );
```

Now we compute the ratios. It turns out that from these candidates, only elements s of the orders 14 and 15 satisfy $P(g,s) < 194/455$.

```
————————————— Example —————————————
gap> nongen:= List( elementstotest,
>                   s -> RatioOfNongenerationTransPermGroup( g, 3A, s ) );;
gap> smaller:= PositionsProperty( nongen, x -> x < 194/455 );
[ 2, 3 ]
gap> nongen{ smaller };
[ 127/325, 1453/3640 ]
```

So the only candidates for s that may be better than order 20 elements are elements of order 14 or 15. In order to exclude these two possibilities, we compute $P(g,s)$ for s in the class 14A and $g = s^7$ in the class 2A, and for s in the class 15A and g in the class 2A, which yields values that are larger than $194/455$.

```
————————————— Example —————————————
gap> repeat s:= Random( g );
>    until Order( s ) = 14 and NrMovedPoints( s ) = 1078;
gap> 2A:= s^7;;
gap> nongen:= RatioOfNongenerationTransPermGroup( g, 2A, s );
1573/3645
gap> nongen > 194/455;
true
gap> repeat s:= Random( g );
```

```
>      until Order( s ) = 15 and NrMovedPoints( s ) = 1080 - 3;
gap> nongen:= RatioOfNongenerationTransPermGroup( g, 2A, s );
490/1053
gap> nongen > 194/455;
true
```

For statement (d), we show that for each triple of elements in the union of the classes 2A, 3A, 3B, 3E there is an element in the class 20A that generates S together with each element of the triple.

```
─────────────────── Example ───────────────────
gap> for tup in UnorderedTuples( [ 2A, 3A, 3B, 3E ], 3 ) do
>        cl:= ShallowCopy( tup );
>        test:= RandomCheckUniformSpread( g, cl, 20A, 100 );
>        if test <> true then
>          Error( test );
>        fi;
>    od;
```

We get no error message, so statement (d) is true.

For statement (e), first we show that $2.S = \Omega^+(8,3)$ contains elements of order 40 but S does not.

```
─────────────────── Example ───────────────────
gap> der:= DerivedSubgroup( SO(1,8,3) );;
gap> repeat x:= PseudoRandom( der ); until Order( x ) = 40;
gap> 40 in ord;
false
```

Thus elements of order 40 must arise as preimages of order 20 elements under the natural epimorphism from $2.S$ to S, which means that we may choose an order 40 preimage $\hat{s}$ of s. Then $\mathbb{M}(2.S, \hat{s})$ consists of central extensions of the subgroups listed in statement (b). The perfect subgroups $O_7(3)$, $L_4(3)$, $2.U_4(3)$, and $U_4(2)$ of these groups must lift to their Schur double covers in $2.S$ because otherwise the preimages would not contain elements of order 40.

Next we consider the preimage of the subgroup $U = (A_4 \times U_4(2)).2$ of S. We show that the preimages of the two direct factors A_4 and $U_4(2)$ in $U' = A_4 \times U_4(2)$ are Schur covers. For A_4, this follows from the fact that the preimage of U' must contain elements of order 20, and that $U_4(2)$ does not contain elements of order 10.

```
─────────────────── Example ───────────────────
gap> u42:= CharacterTable( "U4(2)" );;
gap> Filtered( OrdersClassRepresentatives( u42 ), x -> x mod 5 = 0 );
[ 5 ]
```

In order to show that the $U_4(2)$ type subgroup of U' lifts to its double cover in $2.S$, we note that the class 2B of $U_4(2)$ lifts to a class of elements of order four in the double cover $2.U_4(2)$, and that the corresponding class of elements in U is S-conjugate to the class of involutions in the direct factor A_4 (which is the unique class of length three in U).

```
─────────────────── Example ───────────────────
gap> u:= CharacterTable( Maxes( t )[18] );
CharacterTable( "(A4xU4(2)):2" )
gap> 2u42:= CharacterTable( "2.U4(2)" );;
gap> OrdersClassRepresentatives( 2u42 )[4];
```

```
4
gap> GetFusionMap( 2u42, u42 )[4];
3
gap> OrdersClassRepresentatives( u42 )[3];
2
gap> List( PossibleClassFusions( u42, u ), x -> x[3] );
[ 8 ]
gap> PositionsProperty( SizesConjugacyClasses( u ), x -> x = 3 );
[ 2 ]
gap> ForAll( PossibleClassFusions( u, t ), x -> x[2] = x[8] );
true
```

The last subgroup for which the structure of the preimage has to be shown is $U = (A_6 \times A_6) : 2^2$. We claim that each of the A_6 type subgroups in the derived subgroup $U' = A_6 \times A_6$ lifts to its double cover in $2.S$. Since all elements of order 20 in U lie in U', at least one of the two direct factors must lift to its double cover, in order to give rise to an order 40 element in U. In fact both factors lift to the double cover since the two direct factors are interchanged by conjugation in U; the latter follows form tha fact that U has no normal subgroup of type A_6.

```
──────────────────────────────── Example ────────────────────────────────
gap> u:= CharacterTable( Maxes( t )[24] );
CharacterTable( "(A6xA6):2^2" )
gap> ClassPositionsOfDerivedSubgroup( u );
[ 1 .. 22 ]
gap> PositionsProperty( OrdersClassRepresentatives( u ), x -> x = 20 );
[ 8 ]
gap> List( ClassPositionsOfNormalSubgroups( u ),
>          x -> Sum( SizesConjugacyClasses( u ){ x } ) );
[ 1, 129600, 259200, 259200, 259200, 518400 ]
```

So statement (e) holds.

For statement (f), we have to consider the upward extensions $S.2_1$, $S.2_2$, and $S.3$.

First we look at $S.2_1$, an extension by an outer automorphism that acts as a double transposition in the outer automorphism group S_4. Note that the symmetry between the three classes of element oder 20 in S is broken in $S.2_1$, two of these classes have square roots in $S.2_1$, the third has not.

```
──────────────────────────────── Example ────────────────────────────────
gap> t2:= CharacterTable( "O8+(3).2_1" );;
gap> ord20:= PositionsProperty( OrdersClassRepresentatives( t2 ),
>                     x -> x = 20 );;
gap> ord20:= Intersection( ord20, ClassPositionsOfDerivedSubgroup( t2 ) );
[ 84, 85, 86 ]
gap> List( ord20, x -> x in PowerMap( t2, 2 ) );
[ false, true, true ]
```

Changing the viewpoint, we see that for each class of element order 20 in S, there is a group of the type $S.2_1$ in which the elements in this class do not have square roots, and there are groups of this type in which these elements have square roots. So we have to deal with two different cases, and we do this by first collecting the permutation characters induced from *all* maximal subgroups of $S.2_1$ (other than S) that contain elements of order 20 in S, and then considering s in each of these classes of S.

We fix an embedding of *S* into $S.2_1$ in which the elements in the class 20A do not have square roots. This situation is given for the stored class fusion between the tables in the GAP Character Table Library.

```
─────────── Example ───────────
gap> tfust2:= GetFusionMap( t, t2 );;
gap> tfust2{ PositionsProperty( OrdersClassRepresentatives( t ),
>                 x -> x = 20 ) };
[ 84, 85, 86 ]
```

The six different actions of *S* on the cosets of $O_7(3)$ type subgroups induce pairwise different permutation characters that form an orbit under the action of Aut(*S*). Four of these characters cannot extend to $S.2_1$, the other two extend to permutation characters of $S.2_1$ on the cosets of $O_7(3).2$ type subgroups; these subgroups contain 20A elements.

```
─────────── Example ───────────
gap> primt2:= [];;
gap> poss:= PossiblePermutationCharacters( CharacterTable( "O7(3)" ), t );;
gap> invfus:= InverseMap( tfust2 );;
gap> List( poss, pi -> ForAll( CompositionMaps( pi, invfus ), IsInt ) );
[ false, false, false, false, true, true ]
gap> PossiblePermutationCharacters(
>       CharacterTable( "O7(3)" ) * CharacterTable( "Cyclic", 2 ), t2 );
[  ]
gap> ext:= PossiblePermutationCharacters( CharacterTable( "O7(3).2" ), t2 );;
gap> List( ext, pi -> pi{ ord20 } );
[ [ 1, 0, 0 ], [ 1, 0, 0 ] ]
gap> Append( primt2, ext );
```

The novelties in $S.2_1$ that arise from $O_7(3)$ type subgroups of *S* have the structure $L_4(3).2^2$. These subgroups contain elements in the classes 20B and 20C of *S*.

```
─────────── Example ───────────
gap> ext:= PossiblePermutationCharacters( CharacterTable( "L4(3).2^2" ), t2 );;
gap> List( ext, pi -> pi{ ord20 } );
[ [ 0, 0, 1 ], [ 0, 1, 0 ] ]
gap> Append( primt2, ext );
```

Note that from the possible permutation characters of $S.2_1$ on the cosets of $L_4(3) : 2 \times 2$ type subgroups, we see that such subgroups must contain 20A elements, i. e., all such subgroups of $S.2_1$ lie inside $O_7(3).2$ type subgroups. This means that the structure description of these novelties in [CCN$^+$85, p. 140] is not correct. The correct structure is $L_4(3).2^2$.)

```
─────────── Example ───────────
gap> List( PossiblePermutationCharacters( CharacterTable( "L4(3).2_2" ) *
>               CharacterTable( "Cyclic", 2 ), t2 ), pi -> pi{ ord20 } );
[ [ 1, 0, 0 ] ]
```

All $3^6 : L_4(3)$ type subgroups of *S* extend to $S.2_1$. We compute these permutation characters as the possible permutation characters of the right degree.

```
─────────── Example ───────────
gap> ext:= PermChars( t2, rec( torso:= [ 1120 ] ) );;
gap> List( ext, pi -> pi{ ord20 } );
[ [ 2, 0, 0 ], [ 0, 0, 2 ], [ 0, 2, 0 ] ]
gap> Append( primt2, ext );
```

Also all $2.U_4(3).2^2$ type subgroups of S extend to $S.2_1$. We compute the permutation characters as the extensions of the corresponding permutation characters of S.

```
─────────── Example ───────────
gap> filt:= Filtered( prim, x -> x[1] = 189540 );;
gap> cand:= List( filt, x -> CompositionMaps( x, invfus ) );;
gap> ext:= Concatenation( List( cand,
>                pi -> PermChars( t2, rec( torso:= pi ) ) ) );;
gap> List( ext, x -> x{ ord20 } );
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ]
gap> Append( primt2, ext );
```

The extensions of $(A_4 \times U_4(2)) : 2$ type subgroups of S to $S.2_1$ have the type $S_4 \times U_4(2) : 2$, they contain 20A elements.

```
─────────── Example ───────────
gap> ext:= PossiblePermutationCharacters( CharacterTable( "Symmetric", 4 ) *
>                CharacterTable( "U4(2).2" ), t2 );;
gap> List( ext, x -> x{ ord20 } );
[ [ 1, 0, 0 ], [ 1, 0, 0 ] ]
gap> Append( primt2, ext );
```

All $(A_6 \times A_6) : 2^2$ type subgroups of S extend to $S.2_1$. We compute the permutation characters as the extensions of the corresponding permutation characters of S.

```
─────────── Example ───────────
gap> filt:= Filtered( prim, x -> x[1] = 9552816 );;
gap> cand:= List( filt, x -> CompositionMaps( x, InverseMap( tfust2 ) ));;
gap> ext:= Concatenation( List( cand,
>                pi -> PermChars( t2, rec( torso:= pi ) ) ) );;
gap> List( ext, x -> x{ ord20 } );
[ [ 1, 0, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ]
gap> Append( primt2, ext );
```

We have found all relevant permutation characters of $S.2_1$. This together with the list in [CCN+85, p. 140] implies statement (g).

Now we compute the bounds $\sigma'(S.2_1, s)$.

```
─────────── Example ───────────
gap> Length( primt2 );
15
gap> approx:= List( ord20, x -> ApproxP( primt2, x ) );;
gap> outer:= Difference(
>      PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
>      ClassPositionsOfDerivedSubgroup( t2 ) );;
gap> List( approx, l -> Maximum( l{ outer } ) );
[ 574/1215, 83/567, 83/567 ]
```

Next we look at $S.2_2$, an extension by an outer automorphism that acts as a transposition in the outer automorphism group S_4. Similar to the above situation, the symmetry between the three classes of element oder 20 in S is broken also in $S.2_2$: The first is a conjugacy class of $S.2_2$, the other two classes are fused in $S.2_2$,

```
———————————————————————— Example ————————————————————————
  gap> t2:= CharacterTable( "O8+(3).2_2" );;
  gap> ord20:= PositionsProperty( OrdersClassRepresentatives( t2 ),
  >                  x -> x = 20 );;
  gap> ord20:= Intersection( ord20, ClassPositionsOfDerivedSubgroup( t2 ) );
  [ 82, 83 ]
  gap> tfust2:= GetFusionMap( t, t2 );;
  gap> tfust2{ PositionsProperty( OrdersClassRepresentatives( t ),
  >                  x -> x = 20 ) };
  [ 82, 83, 83 ]
```

Like in the case $S.2_1$, we compute the permutation characters induced from *all* maximal subgroups of $S.2_2$ (other than S) that contain elements of order 20 in S.

We fix the embedding of S into $S.2_2$ in which the class 20A of S is a class of $S.2_2$. This situation is given for the stored class fusion between the tables in the GAP Character Table Library.

Exactly two classes of $O_7(3)$ type subgroups in S extend to $S.2_2$, these groups contain 20A elements.

```
———————————————————————— Example ————————————————————————
  gap> primt2:= [];;
  gap> ext:= PermChars( t2, rec( torso:= [ 1080 ] ) );;
  gap> List( ext, pi -> pi{ ord20 } );
  [ [ 1, 0 ], [ 1, 0 ] ]
  gap> Append( primt2, ext );
```

Only one class of $3^6 : L_4(3)$ type subgroups extends to $S.2_2$. (Note that we need not consider the novelties of the type $3^{3+6} : (L_3(3) \times 2)$, because the order of these groups is not divisible by 5.)

```
———————————————————————— Example ————————————————————————
  gap> ext:= PermChars( t2, rec( torso:= [ 1120 ] ) );;
  gap> List( ext, pi -> pi{ ord20 } );
  [ [ 2, 0 ] ]
  gap> Append( primt2, ext );
```

Only one class of $2.U_4(3).2^2$ type subgroups of S extends to $S.2_2$. We compute the permutation character as the extension of the corresponding permutation characters of S.

```
———————————————————————— Example ————————————————————————
  gap> filt:= Filtered( prim, x -> x[1] = 189540 );;
  gap> cand:= List( filt, x -> CompositionMaps( x, InverseMap( tfust2 ) ));;
  gap> ext:= Concatenation( List( cand,
  >             pi -> PermChars( t2, rec( torso:= pi ) ) ) );;
  gap> List( ext, x -> x{ ord20 } );
  [ [ 1, 0 ] ]
  gap> Append( primt2, ext );
```

Two classes of $(A_4 \times U_4(2)) : 2$ type subgroups of S extend to $S.2_2$.

```
                          ─── Example ───
  gap> filt:= Filtered( prim, x -> x[1] = 7960680 );;
  gap> cand:= List( filt, x -> CompositionMaps( x, InverseMap( tfust2 ) ));;
  gap> ext:= Concatenation( List( cand,
  >              pi -> PermChars( t2, rec( torso:= pi ) ) ) ) );;
  gap> List( ext, x -> x{ ord20 } );
  [ [ 1, 0 ], [ 1, 0 ] ]
  gap> Append( primt2, ext );
```

Exactly one class of $(A_6 \times A_6) : 2^2$ type subgroups in S extends to $S.2_2$, and the extensions have the structure $S_6 \wr 2$.

```
                          ─── Example ───
  gap> ext:= PossiblePermutationCharacters( CharacterTableWreathSymmetric(
  >              CharacterTable( "S6" ), 2 ), t2 );;
  gap> List( ext, x -> x{ ord20 } );
  [ [ 1, 0 ] ]
  gap> Append( primt2, ext );
```

We have found all relevant permutation characters of $S.2_2$, and compute the bounds $\sigma'(S.2_2, s)$.

```
                          ─── Example ───
  gap> Length( primt2 );
  7
  gap> approx:= List( ord20, x -> ApproxP( primt2, x ) );;
  gap> outer:= Difference(
  >        PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
  >        ClassPositionsOfDerivedSubgroup( t2 ) );;
  gap> List( approx, l -> Maximum( l{ outer } ) );
  [ 14/9, 0 ]
```

This means that there is an extension of the type $S.2_2$ in which s cannot be chosen such that the bound is less than $1/2$. More precisely, we have $\sigma(g, s) \geq 1/2$ exactly for g in the unique outer involution class of size 1080.

```
                          ─── Example ───
  gap> approx:= ApproxP( primt2, ord20[1] );;
  gap> bad:= Filtered( outer, i -> approx[i] >= 1/2 );
  [ 84 ]
  gap> OrdersClassRepresentatives( t2 ){ bad };
  [ 2 ]
  gap> SizesConjugacyClasses( t2 ){ bad };
  [ 1080 ]
  gap> Number( SizesConjugacyClasses( t2 ), x -> x = 1080 );
  1
```

So we compute the proportion of elements in this class that generate $S.2_2$ together with an element s of order 20 in S. (As above, we have to consider two conjugacy classes.) For that, we first compute a permutation representation of $S.2_2$, using that $S.2_2$ is isomorphic to the two subgroups of index 2 in $\mathrm{PGO}^+(8,3) = O_8^+(3).2_{122}^2$ that are different from $\mathrm{PSO}^+(8,3) = O_8^+(3).2_1$, cf. [CCN$^+$85, p. 140].

```
————————————————— Example ———————————————————
gap> go:= GO(1,8,3);;
gap> so:= SO(1,8,3);;
gap> outerelm:= First( GeneratorsOfGroup( go ), x -> not x in so );;
gap> g2:= ClosureGroup( DerivedSubgroup( so ), outerelm );;
gap> Size( g2 );
19808719257600
gap> dom:= NormedRowVectors( GF(3)^8 );;
gap> orbs:= OrbitsDomain( g2, dom, OnLines );;
gap> List( orbs, Length );
[ 1080, 1080, 1120 ]
gap> act:= Action( g2, orbs[1], OnLines );;
```

An involution *g* can be found as a power of one of the given generators.

```
————————————————— Example ———————————————————
gap> Order( outerelm );
26
gap> g:= Permutation( outerelm^13, orbs[1], OnLines );;
gap> Size( ConjugacyClass( act, g ) );
1080
```

Now we find the candidates for the elements *s*, and compute their ratios of nongeneration.

```
————————————————— Example ———————————————————
gap> ord20;
[ 82, 83 ]
gap> SizesCentralizers( t2 ){ ord20 };
[ 40, 20 ]
gap> der:= DerivedSubgroup( act );;
gap> repeat 20A:= Random( der );
>    until Order( 20A ) = 20 and Size( Centralizer( act, 20A ) ) = 40;
gap> RatioOfNongenerationTransPermGroup( act, g, 20A );
1
gap> repeat 20BC:= Random( der );
>    until Order( 20BC ) = 20 and Size( Centralizer( act, 20BC ) ) = 20;
gap> RatioOfNongenerationTransPermGroup( act, g, 20BC );
0
```

This means that for *s* in one *S*-class of elements of order 20, we have $P'(g,s) = 1$, and *s* in the other two *S*-classes of elements of order 20 generates with any conjugate of *g*.

Concerning $S.2_2$, it remains to show that we cannot find a better element than *s*. For that, we first compute class representatives s' in *S*, w.r.t. conjugacy in $S.2_2$, and then compute $P'(s',g)$. (It would be enough to check representatives of classes of maximal element order, but computing all classes is easy enough.)

```
————————————————— Example ———————————————————
gap> ccl:= ConjugacyClasses( act );;
gap> der:= DerivedSubgroup( act );;
gap> reps:= Filtered( List( ccl, Representative ), x -> x in der );;
gap> Length( reps );
83
gap> ratios:= List( reps,
```

```
>                             s -> RatioOfNongenerationTransPermGroup( act, g, s ) );;
gap> cand:= PositionsProperty( ratios, x -> x < 1 );;
gap> ratios:= ratios{ cand };;
gap> SortParallel( ratios, cand );
gap> ratios;
[ 0, 1/10, 1/10, 16/135, 1/3, 1/3, 11/27, 7/15, 7/15 ]
```

For $S.2_2$, it remains to show that there is no element $s' \in S$ such that $P'(s'^x, g) < 1$ holds for any $x \in \text{Aut}(S)$ and $g \in S.2_2$. So we are done when we can show that each class given by `cand` is conjugate in $S.3$ to a class outside `cand`. The classes can be identified by element orders and centralizer orders.

```
                            ─── Example ───
gap> invs:= List( cand,
>        x -> [ Order( reps[x] ), Size( Centralizer( der, reps[x] ) ) ] );
[ [ 20, 20 ], [ 18, 108 ], [ 18, 108 ], [ 14, 28 ], [ 15, 45 ],
  [ 15, 45 ], [ 10, 40 ], [ 12, 72 ], [ 12, 72 ] ]
```

Namely, `cand` contains no full $S.3$-orbit of classes of the element orders 20, 18, 14, 15, and 10; also, `cand` does not contain full $S.3$-orbits on the classes 120–12T.

Finally, we deal with $S.3$. The fact that no maximal subgroup of S containing an element of order 20 extends to $S.3$ follows either from the list of maximal subgroups of S in [CCN$^+$85, p. 140] or directly from the permutation characters.

```
                            ─── Example ───
gap> t3:= CharacterTable( "O8+(3).3" );;
gap> tfust3:= GetFusionMap( t, t3 );;
gap> inv:= InverseMap( tfust3 );;
gap> filt:= PositionsProperty( prim, x -> x[ spos ] <> 0 );;
gap> ForAll( prim{ filt },
>            pi -> ForAny( CompositionMaps( pi, inv ), IsList ) );
true
```

So we have to consider only the classes of novelties in $S.3$, but the order of none of these groups is divisible by 20 –again see [CCN$^+$85, p. 140]). This means that *any* element in $S.3 \setminus S$ together with an element of order 20 in S generates $S.3$. This is in fact stronger than statement (f), which claims this property only for elements of prime order in $S.3 \setminus S$ (and their roots); note that $S.3 \setminus S$ contains elements of the orders 9 and 27.

```
                            ─── Example ───
gap> outer:= Difference( [ 1 .. NrConjugacyClasses( t3 ) ],
>                 ClassPositionsOfDerivedSubgroup( t3 ) );
[ 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
  70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86,
  87, 88, 89, 90, 91, 92, 93, 94 ]
gap> Set( OrdersClassRepresentatives( t3 ){ outer } );
[ 3, 6, 9, 12, 18, 21, 24, 27, 36, 39 ]
```

Before we turn to the next computations, we clean the workspace.

```
                            ─── Example ───
gap> CleanWorkspace();
```

11.5.14 $O_8^+(4)$

We show that $S = O_8^+(4) = \Omega^+(8,4)$ satisfies the following.

(a)   For suitable $s \in S$ of the type $2^- \perp 6^-$ (i. e., s decomposes the natural 8-dimensional module for S into an orthogonal sum of two irreducible modules of the dimensions 2 and 6, respectively) and of order 65, $\mathbb{M}(S,s)$ consists of exactly three pairwise nonconjugate subgroups of the type $(5 \times O_6^-(4)).2 = (5 \times \Omega^-(6,4)).2$.

(b) $\sigma(S,s) \leq 34817/1645056$.

(c)   In the extensions $S.2_1$ and $S.3$ of S by graph automorphisms, there is at most one maximal subgroup besides S that contains s. For the extension $S.2_2$ of S by a field automorphism, we have $\sigma'(S.2_2,s) = 0$. In the extension $S.2_3$ of S by the product of an involutory graph automorphism and a field automorphism, there is a unique maximal subgroup besides S that contains s.

A safe source for determining $\mathbb{M}(S,s)$ is [Kle87]. By inspection of the result matrix in this paper, we get that the only maximal subgroups of S that contain elements of order 65 occur in the rows 9–14 and 23–25; they have the isomorphism types $S_6(4) = \mathrm{Sp}(6,4) \cong O_7(4) = \Omega(7,4)$ and $(5 \times O_6^-(4)).2 = (5 \times \Omega^-(6,4)).2$, respectively, and for each of these, there are three conjugacy classes of subgroups in S, which are conjugate under the triality graph automorphism of S.

We start with the natural matrix representation of S. For convenience, we compute an isomorphic permutation group on $5\,525$ points.

```
─────────────────────────── Example ───────────────────────────
  gap> q:= 4;;   n:= 8;;
  gap> G:= DerivedSubgroup( SO( 1, n, q ) );;
  gap> points:= NormedRowVectors( GF(q)^n );;
  gap> orbs:= OrbitsDomain( G, points, OnLines );;
  gap> List( orbs, Length );
  [ 5525, 16320 ]
  gap> hom:= ActionHomomorphism( G, orbs[1], OnLines );;
  gap> G:= Image( hom );;
```

The group S contains exactly six conjugacy classes of (cyclic) subgroups of order 65; this follows from the fact that the centralizer of any Sylow 13 subgroup in S has the structure $5 \times 5 \times 13$.

```
─────────────────────────── Example ───────────────────────────
  gap> Collected( Factors( Size( G ) ) );
  [ [ 2, 24 ], [ 3, 5 ], [ 5, 4 ], [ 7, 1 ], [ 13, 1 ], [ 17, 2 ] ]
  gap> ResetGlobalRandomNumberGenerators();
  gap> repeat x:= Random( G );
  >      until Order( x ) mod 13 = 0;
  gap> x:= x^( Order( x ) / 13 );;
  gap> c:= Centralizer( G, x );;
  gap> IsAbelian( c );   AbelianInvariants( c );
  true
  [ 5, 5, 13 ]
```

The group $S_6(4)$ contains exactly one class of subgroups of order 65, since the conjugacy classes of elements of order 65 in $S_6(4)$ are algebraically conjugate.

```
                           ─── Example ───
  gap> t:= CharacterTable( "S6(4)" );;
  gap> ord65:= PositionsProperty( OrdersClassRepresentatives( t ),
  >                                       x -> x = 65 );
  [ 105, 106, 107, 108, 109, 110, 111, 112 ]
  gap> ord65 = ClassOrbit( t, ord65[1] );
  true
```

Thus there are at least three classes of order 65 elements in *S* that are *not* contained in $S_6(4)$ type subgroups of *S*. So we choose such an element *s*, and have to consider only overgroups of the type $(5 \times \Omega^-(6,4)).2$.

The group $\Omega^-(6,4) \cong U_4(4)$ contains exactly one class of subgroups of order 65.

```
                           ─── Example ───
  gap> t:= CharacterTable( "U4(4)" );;
  gap> ords:= OrdersClassRepresentatives( t );;
  gap> ord65:= PositionsProperty( ords, x -> x = 65 );;
  gap> ord65 = ClassOrbit( t, ord65[1] );
  true
```

So $5 \times \Omega^-(6,4)$ contains exactly six such classes. Furthermore, subgroups in different classes are not *S*-conjugate.

```
                           ─── Example ───
  gap> syl5:= SylowSubgroup( c, 5 );;
  gap> elms:= Filtered( Elements( syl5 ), y -> Order( y ) = 5 );;
  gap> reps:= Set( elms, SmallestGeneratorPerm );;  Length( reps );
  6
  gap> reps65:= List( reps, y -> SubgroupNC( G, [ y * x ] ) );;
  gap> pairs:= Combinations( reps65, 2 );;
  gap> ForAny( pairs, p -> IsConjugate( G, p[1], p[2] ) );
  false
```

We consider only subgroups $M \leq S$ in the three *S*-classes of the type $(5 \times \Omega^-(6,4)).2$.

```
                           ─── Example ───
  gap> cand:= List( reps, y -> Normalizer( G, SubgroupNC( G, [ y ] ) ) );;
  gap> cand:= Filtered( cand, y -> Size( y ) = 10 * Size( t ) );;
  gap> Length( cand );
  3
```

(Note that one of the members in $\mathbb{M}(S,s)$ is the stabilizer in *S* of the orthogonal decomposition $2^- \perp 6^-$, the other two members are not reducible.)

By the above, the classes of subgroups of order 65 in each such *M* are in bijection with the corresponding classes in *S*. Since $N_S(\langle g \rangle) \subseteq M$ holds for any $g \in M$ of order 65, also the conjugacy classes of *elements* of order 65 in *M* are in bijection with those in *S*.

```
                           ─── Example ───
  gap> norms:= List( reps65, y -> Normalizer( G, y ) );;
  gap> ForAll( norms, y -> ForAll( cand, M -> IsSubset( M, y ) ) );
  true
```

As a consequence, we have $g^S \cap M = g^M$ and thus $1_M^S(g) = 1$. This implies statement (a).

In order to show statement (b), we want to use the function `UpperBoundFixedPointRatios` introduced in Section 11.3.3. For that, we first compute the conjugacy classes of the three class representatives M. (Since the groups have elementary abelian Sylow 5 subgroups of the order 5^4, computing all conjugacy classes appears to be faster than using `ClassesOfPrimeOrder`.) Then we compute an upper bound for $\sigma(S, s)$.

```
———————————————————————— Example ————————————————————————
  gap> syl5:= SylowSubgroup( cand[1], 5 );;
  gap> Size( syl5 );  IsElementaryAbelian( syl5 );
  625
  true
  gap> UpperBoundFixedPointRatios( G, List( cand, ConjugacyClasses ), false );
  [ 34817/1645056, false ]
```

Remark:

Computing the exact value $\sigma(S, s)$ in the above setup would require to test the S-conjugacy of certain order 5 elements in M. With the current GAP implementation, some of the relevant tests need several hours of CPU time.

An alternative approach would be to compute the permutation action of S on the cosets of M, of degree $6\,580\,224$, and to count the fixed points of conjugacy class representatives of prime order. The currently available GAP library methods are not sufficient for computing this in reasonable time. "Ad-hoc code" for this special case works, but it seemed to be not appropriate to include it here.

In the proof of statement (c), again we consult the result matrix in [Kle87]. For $S.3$, the maximal subgroups are in the rows 4, 15, 22, 26, and 61. Only row 26 yields subgroups that contain elements s of order 65, they have the isomorphism type $(5 \times GU(3,4)).2 \cong (5^2 \times U_3(4)).2$. Note that the conjugacy classes of the members in $\mathbb{M}(S, s)$ are permuted by the outer automorphism of order 3, so none of the subgroups in $\mathbb{M}(S, s)$ extends to $S.3$. By [BGK08, Lemma 2.4 (2)], if there is a maximal subgroup of $S.3$ besides S that contains s then this subgroup is the normalizer in $S.3$ of the intersection of the three members of $\mathbb{M}(S, s)$, i. e., s is contained in at most one such subgroup.

For $S.2_1$, only the rows 9 and 23 yield maximal subgroups containing elements of order 65, and since we had chosen s in such a way that row 9 was excluded already for the simple group, only extensions of the elements in $\mathbb{M}(S, s)$ can appear. Exactly one of these three subgroups of S extends to $S.2_1$, so again we get just one maximal subgroup of $S.2_1$, besides S, that contains s.

All subgroups in $\mathbb{M}(S, s)$ extend to $S.2_2$, see [Kle87]. We compute the extensions of the above subgroups M of S to $S.2_2$, by constructing the action of the field automorphism in the permutation representation we used for S. In other words, we compute the projective action of the Frobenius map.

```
———————————————————————— Example ————————————————————————
  gap> frob:= PermList( List( orbs[1], v -> Position( orbs[1],
  >              List( v, x -> x^2 ) ) ) );;
  gap> G2:= ClosureGroupDefault( G, frob );;
  gap> cand2:= List( cand, M -> Normalizer( G2, M ) );;
  gap> ccl:= List( cand2,
  >              M2 -> PcConjugacyClassReps( SylowSubgroup( M2, 2 ) ) );;
  gap> List( ccl, l -> Number( l, x -> Order( x ) = 2 and not x in G ) );
  [ 0, 0, 0 ]
```

So in each case, the extension of M to its normalizer in $S.2_2$ is non-split. This implies $\sigma'(S.2_2, s) = 0$.

Finally, in the extension of S by the product of a graph automorphism and the field automorphism, exactly that member of $\mathbb{M}(S,s)$ is invariant that is invariant under the graph automorphism, hence statement (c) holds.

It is again time to clean the workspace.

```
―――――――― Example ――――――――
gap> CleanWorkspace();
```

11.5.15 $* O_9(3)$

The group $S = O_9(3) = \Omega_9(3)$ is the first member in the series dealt with in [BGK08, Proposition 5.7], and serves as an example to illustrate this statement.

(a) For $s \in S$ of the type $1 \perp 8^-$ (i. e., s decomposes the natural 9–dimensional module for S into an orthogonal sum of two irreducible modules of the dimensions 1 and 8, respectively) and of order $(3^4 + 1)/2 = 41$, $\mathbb{M}(S,s)$ consists of one group of the type $O_8^-(3).2_1 = \mathrm{PGO}^-(8,3)$.

(b) $\sigma(S,s) = 1/3$.

(c) The uniform spread of S is at least three, with s of order 41.

By [MSW94], the only maximal subgroup of S that contains s is the stabilizer M of the orthogonal decomposition. The group $2 \times O_8^-(3).2_1 = \mathrm{GO}^-(8,3)$ embeds naturally into $\mathrm{SO}(9,3)$, its intersection with S is $\mathrm{PGO}^-(8,3)$. This proves statement (a).

The group M is the stabilizer of a 1–space, it has index 3240 in S.

```
―――――――― Example ――――――――
gap> g:= SO( 9, 3 );;
gap> g:= DerivedSubgroup( g );;
gap> Size( g );
65784756654489600
gap> orbs:= OrbitsDomain( g, NormedRowVectors( GF(3)^9 ), OnLines );;
gap> List( orbs, Length ) / 41;
[ 3240/41, 81, 80 ]
gap> Size( SO( 9, 3 ) ) / Size( GO( -1, 8, 3 ) );
3240
```

So we compute the unique transitive permutation character of S that has degree 3240.

```
―――――――― Example ――――――――
gap> t:= CharacterTable( "O9(3)" );;
gap> pi:= PermChars( t, rec( torso:= [ 3240 ] ) );
[ Character( CharacterTable( "O9(3)" ),
  [ 3240, 1080, 380, 132, 48, 324, 378, 351, 0, 0, 54, 27, 54, 27, 0,
      118, 0, 36, 46, 18, 12, 2, 8, 45, 0, 108, 108, 135, 126, 0, 0,
      56, 0, 0, 36, 47, 38, 27, 39, 36, 24, 12, 18, 18, 15, 24, 2,
      18, 15, 9, 0, 0, 0, 2, 0, 18, 11, 3, 9, 6, 6, 9, 6, 3, 6, 3, 0,
      6, 16, 0, 4, 6, 2, 45, 36, 0, 0, 0, 0, 0, 0, 0, 9, 9, 6, 3, 0,
      0, 15, 13, 0, 5, 7, 36, 0, 10, 0, 10, 19, 6, 15, 0, 0, 0, 0,
      12, 3, 10, 0, 3, 3, 7, 0, 6, 6, 2, 8, 0, 4, 0, 2, 0, 1, 3, 0,
      0, 3, 0, 3, 2, 2, 3, 3, 6, 2, 2, 9, 6, 3, 0, 0, 18, 9, 0, 0,
      12, 0, 0, 8, 0, 6, 9, 5, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 3, 2, 1,
      3, 3, 1, 0, 0, 4, 1, 0, 0, 1, 0, 3, 3, 1, 1, 2, 2, 0, 0, 1, 3,
```

```
         4, 0, 1, 2, 0, 0, 1, 0, 4, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
         1, 1, 1, 1, 0, 0, 1, 1, 1, 0 ] ) ]
gap> spos:= Position( OrdersClassRepresentatives( t ), 41 );
208
gap> approx:= ApproxP( pi, spos );;
gap> Maximum( approx );
1/3
gap> PositionsProperty( approx, x -> x = 1/3 );
[ 2 ]
gap> SizesConjugacyClasses( t )[2];
3321
gap> OrdersClassRepresentatives( t )[2];
2
```

We see that $P(S,s) = \sigma(S,s) = 1/3$ holds, and that $\sigma(g,s)$ attains this maximum only for g in one class of involutions in S; let us call this class 2A. (This class consists of the negatives of a class of *reflections* in GO(9,3).) This shows statement (b).

In order to show that the uniform spread of S is at least three, it suffices to show that for each triple of 2A elements, there is an element s of order 41 in S that generates S with each element of the triple.

We work with the primitive permutation representation of S on 3 240 points. In this representation, s fixes exactly one point, and by statement (a), s generates S with $x \in S$ if and only if x moves this point. Since the number of fixed points of each 2A involution in S is exactly one third of the moved points of S, it suffices to show that we cannot choose three such involutions with mutually disjoint fixed point sets. And this is shown particularly easily because it will turn out that already for any two different 2A involutions, the sets of fixed points of are never disjoint.

First we compute a 2A element, which is determined as an involution with exactly 1 080 fixed points.

```
─────────── Example ───────────
  gap> g:= Action( g, orbs[1], OnLines );;
  gap> repeat
  >        repeat x:= Random( g ); ord:= Order( x ); until ord mod 2 = 0;
  >        y:= x^(ord/2);
  > until NrMovedPoints( y ) = 3240 - 1080;
```

Next we compute the sets of fixed points of the elements in the class 2A, by forming the S-orbit of the set of fixed points of the chosen 2A element.

```
─────────── Example ───────────
  gap> fp:= Difference( MovedPoints( g ), MovedPoints( y ) );;
  gap> orb:= Orbit( g, fp, OnSets );;
```

Finally, we show that for any pair of 2A elements, their sets of fixed points intersect nontrivially. (Of course we can fix one of the two elements.) This proves statement (c).

```
─────────── Example ───────────
  gap> ForAny( orb, l -> IsEmpty( Intersection( l, fp ) ) );
  false
```

11.5.16 $O_{10}^-(3)$

We show that the group $S = O_{10}^-(3) = \mathrm{P}\Omega^-(10,3)$ satisfies the following.

(a) For $s \in S$ irreducible of order $(3^5 + 1)/4 = 61$, $\mathbb{M}(S,s)$ consists of one subgroup of the type $SU(5,3) \cong U_5(3)$.

(b) $\sigma(S,s) = 1/1066$.

By [Ber00], the maximal subgroups of S containing s are of extension field type, and by [KL90, Prop. 4.3.18 and 4.3.20], these groups have the structure $SU(5,3) = U_5(3)$ (which lift to $2 \times U_5(3) <$ $GU(5,3)$ in $\Omega^-(10,3) = 2.S$) or $\Omega(5,9).2$, but the order of the latter group is not divisible by $|s|$. Furthermore, by [BGK08, Lemma 2.12 (b)], s is contained in only one member of the former class.

--- Example ---
```
gap> Size( GO(5,9) ) / 61;
6886425600/61
```

When the first version of these computations was written, the character tables of both S and $U_5(3)$ were not contained in the GAP Character Table Library, so we worked with the groups. Meanwhile the character tables are available, thus we can show also a character theoretic solution.)

--- Example ---
```
gap> t:= CharacterTable( "O10-(3)" );  s:= CharacterTable( "U5(3)" );
CharacterTable( "O10-(3)" )
CharacterTable( "U5(3)" )
gap> SigmaFromMaxes( t, "61A", [ s ], [ 1 ] );
1/1066
```

(Now follow the computations with groups.)
The first step is the construction of the embedding of $M = SU(5,3)$ into the matrix group $2.S$, that is, we write the matrix generators of M as linear mappings on the natural module for $2.S$, and then conjugate them such that the result matrices respect the bilinear form of $2.S$. For convenience, we choose a basis for the field extension $\mathbb{F}_9/\mathbb{F}_3$ such that the $\mathbb{F}_3$-linear mapping given by the invariant form of M is invariant under the $\mathbb{F}_3$-linear mappings given by the generators of M.

--- Example ---
```
gap> m:= SU(5,3);;
gap> so:= SO(-1,10,3);;
gap> omega:= DerivedSubgroup( so );;
gap> om:= InvariantBilinearForm( so ).matrix;;
gap> Display( om );
 . 1 . . . . . . . .
 1 . . . . . . . . .
 . . 1 . . . . . . .
 . . . 2 . . . . . .
 . . . . 2 . . . . .
 . . . . . 2 . . . .
 . . . . . . 2 . . .
 . . . . . . . 2 . .
 . . . . . . . . 2 .
 . . . . . . . . . 2
gap> b:= Basis( GF(9), [ Z(3)^0, Z(3^2)^2 ] );
```

```
   Basis( GF(3^2), [ Z(3)^0, Z(3^2)^2 ] )
   gap> blow:= List( GeneratorsOfGroup( m ), x -> BlownUpMat( b, x ) );;
   gap> form:= BlownUpMat( b, InvariantSesquilinearForm( m ).matrix );;
   gap> ForAll( blow, x -> x * form * TransposedMat( x ) = form );
   true
   gap> Display( form );
    . . . . . . . . 1 .
    . . . . . . . . . 1
    . . . . . . 1 . . .
    . . . . . . . 1 . .
    . . . . 1 . . . . .
    . . . . . 1 . . . .
    . . 1 . . . . . . .
    . . . 1 . . . . . .
    1 . . . . . . . . .
    . 1 . . . . . . . .
```

The matrix om of the invariant bilinear form of 2.*S* is equivalent to the identity matrix *I*. So we compute matrices T1 and T2 that transform om and form, respectively, to $\pm I$.

```
   ─────────────── Example ───────────────
   gap> T1:= IdentityMat( 10, GF(3) );;
   gap> T1{[1..3]}{[1..3]}:= [[1,1,0],[1,-1,1],[1,-1,-1]]*Z(3)^0;;
   gap> pi:= PermutationMat( (1,10)(3,8), 10, GF(3) );;
   gap> tr:= NullMat( 10,10,GF(3) );;
   gap> tr{[1, 2]}{[1, 2]}:= [[1,1],[1,-1]]*Z(3)^0;;
   gap> tr{[3, 4]}{[3, 4]}:= [[1,1],[1,-1]]*Z(3)^0;;
   gap> tr{[7, 8]}{[7, 8]}:= [[1,1],[1,-1]]*Z(3)^0;;
   gap> tr{[9,10]}{[9,10]}:= [[1,1],[1,-1]]*Z(3)^0;;
   gap> tr{[5, 6]}{[5, 6]}:= [[1,0],[0,1]]*Z(3)^0;;
   gap> tr2:= IdentityMat( 10,GF(3) );;
   gap> tr2{[1,3]}{[1,3]}:= [[-1,1],[1,1]]*Z(3)^0;;
   gap> tr2{[7,9]}{[7,9]}:= [[-1,1],[1,1]]*Z(3)^0;;
   gap> T2:= tr2 * tr * pi;;
   gap> D:= T1^-1 * T2;;
   gap> tblow:= List( blow, x -> D * x * D^-1 );;
   gap> IsSubset( omega, tblow );
   true
```

Now we switch to a permutation representation of *S*, and use the embedding of *M* into 2.*S* to obtain the corresponding subgroup of type *M* in *S*. Then we compute an upper bound for $\max\{\mu(g, S/M); g \in S^\times\}$.

```
   ─────────────── Example ───────────────
   gap> orbs:= OrbitsDomain( omega, NormedRowVectors( GF(3)^10 ), OnLines );;
   gap> List( orbs, Length );
   [ 9882, 9882, 9760 ]
   gap> permgrp:= Action( omega, orbs[3], OnLines );;
   gap> M:= SubgroupNC( permgrp,
   >             List( tblow, x -> Permutation( x, orbs[3], OnLines ) ) );;
   gap> ccl:= ClassesOfPrimeOrder( M, PrimeDivisors( Size( M ) ),
   >                               TrivialSubgroup( M ) );;
```

```
gap> UpperBoundFixedPointRatios( permgrp, [ ccl ], false );
[ 1/1066, true ]
```

The entry `true` in the second position of the result indicates that in fact the *exact* value for the maximum of $\mu(g, S/M)$ has been computed. This implies statement (b).

We clean the workspace.

```
——————————— Example ———————————
gap> CleanWorkspace();
```

11.5.17 $O_{14}^-(2)$

We show that the group $S = O_{14}^-(2) = \Omega^-(14,2)$ satisfies the following.

(a)   For $s \in S$ irreducible of order $2^7 + 1 = 129$, $\mathbb{M}(S,s)$ consists of one subgroup M of the type $\mathrm{GU}(7,2) \cong 3 \times U_7(2)$.

(b) $\sigma(S,s) = 1/2\,015$.

By [Ber00], any maximal subgroup of S containing s is of extension field type, and by [KL90, Table 3.5.F, Prop. 4.3.18], these groups have the type $\mathrm{GU}(7,2)$, and there is exactly one class of subgroups of this type. Furthermore, by [BGK08, Lemma 2.12 (a)], s is contained in only one member of this class.

We embed $U_7(2)$ into S, by first replacing each element in $\mathbb{F}_4$ by the 2×2 matrix of the induced $\mathbb{F}_2$-linear mapping w.r.t. a suitable basis, and then conjugating the images of the generators such that the invariant quadratic form of S is respected.

```
——————————— Example ———————————
gap> o:= SO(-1,14,2);;
gap> g:= SU(7,2);;
gap> b:= Basis( GF(4) );;
gap> blow:= List( GeneratorsOfGroup( g ), x -> BlownUpMat( b, x ) );;
gap> form:= NullMat( 14, 14, GF(2) );;
gap> for i in [ 1 .. 14 ] do form[i][ 15-i ]:= Z(2); od;
gap> ForAll( blow, x -> x * form * TransposedMat( x ) = form );
true
gap> pi:= PermutationMat( (1,13)(3,11)(5,9), 14, GF(2) );;
gap> pi * form * TransposedMat( pi ) = InvariantBilinearForm( o ).matrix;
true
gap> pi2:= PermutationMat( (7,3)(8,4), 14, GF(2) );;
gap> D:= pi2 * pi;;
gap> tblow:= List( blow, x -> D * x * D^-1 );;
gap> IsSubset( o, tblow );
true
```

Note that the central subgroup of order three in $\mathrm{GU}(7,2)$ consists of scalar matrices.

```
——————————— Example ———————————
gap> omega:= DerivedSubgroup( o );;
gap> IsSubset( omega, tblow );
true
gap> z:= Z(4) * One( g );;
```

```
gap> tz:= D * BlownUpMat( b, z ) * D^-1;;
gap> tz in omega;
true
```

Now we switch to a permutation representation of *S*, and compute the conjugacy classes of prime element order in the subgroup *M*. The latter is done in two steps, first class representatives of the simple subgroup $U_7(2)$ of *M* are computed, and then they are multiplied with the scalars in *M*.

```
────────────── Example ──────────────
gap> orbs:= OrbitsDomain( omega, NormedRowVectors( GF(2)^14 ), OnLines );;
gap> List( orbs, Length );
[ 8127, 8256 ]
gap> omega:= Action( omega, orbs[1], OnLines );;
gap> gens:= List( GeneratorsOfGroup( g ),
>              x -> Permutation( D * BlownUpMat( b, x ) * D^-1, orbs[1] ) );;
gap> g:= Group( gens );;
gap> ccl:= ClassesOfPrimeOrder( g, PrimeDivisors( Size( g ) ),
>                                  TrivialSubgroup( g ) );;
gap> tz:= Permutation( tz, orbs[1] );;
gap> primereps:= List( ccl, Representative );;
gap> Add( primereps, () );
gap> reps:= Concatenation( List( primereps,
>              x -> List( [ 0 .. 2 ], i -> x * tz^i ) ) );;
gap> primereps:= Filtered( reps, x -> IsPrimeInt( Order( x ) ) );;
gap> Length( primereps );
48
```

Finally, we apply `UpperBoundFixedPointRatios` (see Section 11.3.3) to compute an upper bound for $\mu(g, S/M)$, for $g \in S^\times$.

```
────────────── Example ──────────────
gap> M:= ClosureGroup( g, tz );;
gap> bccl:= List( primereps, x -> ConjugacyClass( M, x ) );;
gap> UpperBoundFixedPointRatios( omega, [ bccl ], false );
[ 1/2015, true ]
```

Although some of the classes of *M* in the list `bccl` may be *S*-conjugate, the entry `true` in the second position of the result indicates that in fact the *exact* value for the maximum of $\mu(g, S/M)$, for $g \in S^\times$, has been computed. This implies statement (b).

We clean the workspace.

```
────────────── Example ──────────────
gap> CleanWorkspace();
```

11.5.18 $O_{12}^+(3)$

We show that the group $S = O_{12}^+(3) = \mathrm{P}\Omega^+(12,3)$ satisfies the following.

(a)   *S* has a maximal subgroup *M* of the type $N_S(\mathrm{P}\Omega^+(6,9))$, which has the structure $\mathrm{P}\Omega^+(6,9).[4]$.

(b) $\mu(g, S/M) \le 2/88\,209$ holds for all $g \in S^\times$.

(This result is used in the proof of [BGK08, Proposition 5.14], where it is shown that for $s \in S$ of order 205, $\mathbb{M}(S, s)$ consists of one reducible subgroup G_8 and at most two extension field type subgroups of the type $N_S(\mathrm{P}\Omega^+(6, 9))$. By [GK00, Proposition 3.16], $\mu(g, S/G_8) \leq 19/3^5$ holds for all $g \in S^\times$. This implies $P(g, s) \leq 19/3^5 + 2 \cdot 2/88\,209 = 6\,901/88\,209 < 1/3$.)

Statement (a) follows from [KL90, Prop. 4.3.14].

For statement (b), we embed $\mathrm{GO}^+(6, 9) \cong \Omega^+(6, 9).2^2$ into $\mathrm{SO}^+(12, 3) = 2.S.2$, by replacing each element in $\mathbb{F}_9$ by the 2×2 matrix of the induced $\mathbb{F}_3$-linear mapping w.r.t. a suitable basis (b_1, b_2). We choose a basis with the property $b_1 = 1$ and $b_2^2 = 1 + b_2$, because then the image of a symmetric matrix is again symmetric (so the image of the invariant form is an invariant form for the image of the group), and apply an appropriate transformation to the images of the generators.

```
———————————————————————— Example ————————————————————————
gap> so:= SO(+1,12,3);;
gap> Display( InvariantBilinearForm( so ).matrix );
 . 1 . . . . . . . . . .
 1 . . . . . . . . . . .
 . . 1 . . . . . . . . .
 . . . 2 . . . . . . . .
 . . . . 2 . . . . . . .
 . . . . . 2 . . . . . .
 . . . . . . 2 . . . . .
 . . . . . . . 2 . . . .
 . . . . . . . . 2 . . .
 . . . . . . . . . 2 . .
 . . . . . . . . . . 2 .
 . . . . . . . . . . . 2
gap> g:= GO(+1,6,9);;
gap> Z(9)^2 = Z(3)^0 + Z(9);
true
gap> b:= Basis( GF(9), [ Z(3)^0, Z(9) ] );
Basis( GF(3^2), [ Z(3)^0, Z(3^2) ] )
gap> blow:= List( GeneratorsOfGroup( g ), x -> BlownUpMat( b, x ) );;
gap> m:= BlownUpMat( b, InvariantBilinearForm( g ).matrix );;
gap> Display( m );
 . . 1 . . . . . . . . .
 . . . 1 . . . . . . . .
 1 . . . . . . . . . . .
 . 1 . . . . . . . . . .
 . . . . 2 . . . . . . .
 . . . . . 2 . . . . . .
 . . . . . . 2 . . . . .
 . . . . . . . 2 . . . .
 . . . . . . . . 2 . . .
 . . . . . . . . . 2 . .
 . . . . . . . . . . 2 .
 . . . . . . . . . . . 2
gap> pi:= PermutationMat( (2,3), 12, GF(3) );;
gap> tr:= IdentityMat( 12, GF(3) );;
gap> tr{[3,4]}{[3,4]}:= [[1,-1],[1,1]]*Z(3)^0;;
gap> D:= tr * pi;;
gap> D * m * TransposedMat( D ) = InvariantBilinearForm( so ).matrix;
true
gap> tblow:= List( blow, x -> D * x * D^-1 );;
```

```
gap> IsSubset( so, tblow );
true
```

The image of $\mathrm{GO}^+(6,9)$ under the embedding into $\mathrm{SO}^+(12,3)$ does not lie in $\Omega^+(12,3) = 2.S$, so a factor of two is missing in $\mathrm{GO}^+(6,9) \cap 2.S$ for getting (the preimage $2.M$ of) the required maximal subgroup M of S. Because of this, and also because currently it is time consuming to compute the derived subgroup of $\mathrm{SO}^+(12,3)$, we work with the upward extension $\mathrm{PSO}^+(12,3) = S.2$. Note that M extends to a maximal subgroup of $S.2$.

First we factor out the centre of $\mathrm{SO}^+(12,3)$, and switch to a permutation representation of $S.2$.

```
———————————————— Example ————————————————
gap> orbs:= OrbitsDomain( so, NormedRowVectors( GF(3)^12 ), OnLines );;
gap> List( orbs, Length );
[ 88452, 88452, 88816 ]
gap> act:= Action( so, orbs[1], OnLines );;
gap> SetSize( act, Size( so ) / 2 );
```

Next we rewrite the matrix generators for $\mathrm{GO}^+(6,9)$ accordingly, and compute the normalizer in $S.2$ of the subgroup they generate; this is the maximal subgroup $M.2$ we need.

```
———————————————— Example ————————————————
gap> u:= SubgroupNC( act,
>          List( tblow, x -> Permutation( x, orbs[1], OnLines ) ) );;
gap> n:= Normalizer( act, u );;
gap> Size( n ) / Size( u );
2
```

Now we compute class representatives of prime order in $M.2$, in a smaller faithful permutation representation, and then the desired upper bound for $\mu(g, S/M)$.

```
———————————————— Example ————————————————
gap> norbs:= OrbitsDomain( n, MovedPoints( n ) );;
gap> List( norbs, Length );
[ 58968, 29484 ]
gap> hom:= ActionHomomorphism( n, norbs[2] );;
gap> nact:= Image( hom );;
gap> Size( nact ) = Size( n );
true
gap> ccl:= ClassesOfPrimeOrder( nact, PrimeDivisors( Size( nact ) ),
>                                TrivialSubgroup( nact ) );;
gap> Length( ccl );
26
gap> preim:= List( ccl,
>        x -> PreImagesRepresentative( hom, Representative( x ) ) );;
gap> pccl:= List( preim, x -> ConjugacyClass( n, x ) );;
gap> for i in [ 1 .. Length( pccl ) ] do
>      SetSize( pccl[i], Size( ccl[i] ) );
>    od;
gap> UpperBoundFixedPointRatios( act, [ pccl ], false );
[ 2/88209, true ]
```

Note that we have computed $\max\{\mu(g, S.2/M.2), g \in S.2^\times\} \geq \max\{\mu(g, S.2/M.2), g \in S^\times\} = \max\{\mu(g, S/M), g \in S^\times\}$.

11.5.19 $\ast S_4(8)$

We show that the group $S = S_4(8) = \mathrm{Sp}(4,8)$ satisfies the following.

(a) For $s \in S$ irreducible of order 65, $\mathbb{M}(S,s)$ consists of two nonconjugate subgroups of the type
$S_2(64).2 = \mathrm{Sp}(2,64).2 \cong L_2(64).2 \cong O_4^-(8).2 = \Omega^-(4,8).2$.

(b) $\sigma(S,s) = 8/63$.

By [Ber00], the only maximal subgroups of S that contain s are $O_4^-(8).2 = \mathrm{SO}^-(4,8)$ or of extension field type. By [KL90, Prop. 4.3.10, 4.8.6], there is one class of each of these subgroups (which happen to be isomorphic).

These classes of subgroups induce different permutation characters. One argument to see this is that the involutions in the outer half of extension field type subgroup $S_2(64).2 < S_4(8)$ have a two-dimensional fixed space, whereas the outer involutions in $\mathrm{SO}^-(4,8)$ have a three-dimensional fixed space.

The former statement can be seen by using a normal basis of the field extension $\mathbb{F}_{64}/\mathbb{F}_8$, such that the action of the Frobenius automorphism (which yields a suitable outer involution) is just a double transposition on the basis vectors of the natural module for S.

```
———————————————————— Example ————————————————————
gap> sp:= SP(4,8);;
gap> Display( InvariantBilinearForm( sp ).matrix );
 . . . 1
 . . 1 .
 . 1 . .
 1 . . .
gap> z:= Z(64);;
gap> f:= AsField( GF(8), GF(64) );;
gap> repeat
>      b:= Basis( f, [ z, z^8 ] );
>      z:= z * Z(64);
> until b <> fail;
gap> sub:= SP(2,64);;
gap> Display( InvariantBilinearForm( sub ).matrix );
 . 1
 1 .
gap> ext:= Group( List( GeneratorsOfGroup( sub ),
>                       x -> BlownUpMat( b, x ) ) );;
gap> tr:= PermutationMat( (3,4), 4, GF(2) );;
gap> conj:= ConjugateGroup( ext, tr );;
gap> IsSubset( sp, conj );
true
gap> inv:= [[0,1,0,0],[1,0,0,0],[0,0,0,1],[0,0,1,0]] * Z(2);;
gap> inv in sp;
true
gap> inv in conj;
false
gap> Length( NullspaceMat( inv - inv^0 ) );
2
```

The latter statement can be shown by looking at an outer involution in $\mathrm{SO}^-(4,8)$.

```
                        ─── Example ───
  gap> so:= SO(-1,4,8);;
  gap> der:= DerivedSubgroup( so );;
  gap> x:= First( GeneratorsOfGroup( so ), x -> not x in der );;
  gap> x:= x^( Order(x)/2 );;
  gap> Length( NullspaceMat( x - x^0 ) );
  3
```

The character table of $L_2(64).2$ is currently not available in the GAP Character Table Library, so we compute the possible permutation characters with a combinatorial approach, and show statement (a).

```
                        ─── Example ───
  gap> CharacterTable( "L2(64).2" );
  fail
  gap> t:= CharacterTable( "S4(8)" );;
  gap> degree:= Size( t ) / ( 2 * Size( SL(2,64) ) );;
  gap> pi:= PermChars( t, rec( torso:= [ degree ] ) );
  [ Character( CharacterTable( "S4(8)" ),
    [ 2016, 0, 256, 32, 0, 36, 0, 8, 1, 0, 4, 0, 0, 0, 28, 28, 28, 0,
        0, 0, 0, 0, 0, 36, 36, 36, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,
        4, 4, 4, 0, 0, 0, 4, 4, 4, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1 ] ), Character( CharacterTable( "S4(8)" ),
    [ 2016, 256, 0, 32, 36, 0, 0, 8, 1, 4, 0, 28, 28, 28, 0, 0, 0, 0,
        0, 0, 36, 36, 36, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 4, 4, 4,
        0, 0, 0, 4, 4, 4, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1 ] ) ]
  gap> spos:= Position( OrdersClassRepresentatives( t ), 65 );;
  gap> List( pi, x -> x[ spos ] );
  [ 1, 1 ]
```

Now we compute $\sigma(S,s)$, which yields statement (b).

```
                        ─── Example ───
  gap> Maximum( ApproxP( pi, spos ) );
  8/63
```

We clean the workspace.

```
                        ─── Example ───
  gap> CleanWorkspace();
```

11.5.20 $S_6(2)$

We show that the group $S = S_6(2) = \mathrm{Sp}(6,2)$ satisfies the following.

(a) $\sigma(S) = 4/7$, and this value is attained exactly for $\sigma(S,s)$ with s of order 9.

(b)   For $s \in S$ of order 9, $\mathbb{M}(S,s)$ consists of one subgroup of the type $U_4(2).2 = \Omega^-(6,2).2$ and three conjugate subgroups of the type $L_2(8).3 = \mathrm{Sp}(2,8).3$.

(c) For $s \in S$ of order 9, and $g \in S^{\times}$, we have $P(g,s) < 1/3$, except if g is in one of the classes 2A (the transvection class) or 3A.

(d) For $s \in S$ of order 15, and $g \in S^{\times}$, we have $P(g,s) < 1/3$, except if g is in one of the classes 2A or 2B.

(e) $P(S) = 11/21$, and this value is attained exactly for $P(S,s)$ with s of order 15.

(f) For all $s' \in S$, we have $P(g,s') > 1/3$ for g in at least two classes.

(g) The uniform spread of S is at least two, with s of order 9.

(Note that in this example, the optimal choice of s w.r.t. $\sigma(S,s)$ is not optimal w.r.t. $P(S,s)$.)

Statement (a) follows from the inspection of the primitive permutation characters, cf. Section 11.4.4.

```
─────────────────────────── Example ───────────────────────────
 gap> t:= CharacterTable( "S6(2)" );;
 gap> ProbGenInfoSimple( t );
 [ "S6(2)", 4/7, 1, [ "9A" ], [ 4 ] ]
```

Also statement (b) follows from the information provided by the character table of S (cf. [CCN^{+}85, p. 46]).

```
─────────────────────────── Example ───────────────────────────
 gap> prim:= PrimitivePermutationCharacters( t );;
 gap> ord:= OrdersClassRepresentatives( t );;
 gap> spos:= Position( ord, 9 );;
 gap> filt:= PositionsProperty( prim, x -> x[ spos ] <> 0 );
 [ 1, 8 ]
 gap> Maxes( t ){ filt };
 [ "U4(2).2", "L2(8).3" ]
 gap> List( prim{ filt }, x -> x[ spos ] );
 [ 1, 3 ]
```

Now we consider statement (c). For s of order 9 and g in one of the classes 2A, 3A, we observe that $P(g,s) = \sigma(g,s)$ holds. This is because exactly one maximal subgroup of S contains both s and g. For all other elements g, we have even $\sigma(g,s) < 1/3$.

```
─────────────────────────── Example ───────────────────────────
 gap> prim:= PrimitivePermutationCharacters( t );;
 gap> spos9:= Position( ord, 9 );;
 gap> approx9:= ApproxP( prim, spos9 );;
 gap> filt9:= PositionsProperty( approx9, x -> x >= 1/3 );
 [ 2, 6 ]
 gap> AtlasClassNames( t ){ filt9 };
 [ "2A", "3A" ]
 gap> approx9{ filt9 };
 [ 4/7, 5/14 ]
 gap> List( Filtered( prim, x -> x[ spos9 ] <> 0 ), x -> x{ filt9 } );
 [ [ 16, 10 ], [ 0, 0 ] ]
```

Similarly, statement (d) follows. For *s* of order 15 and *g* in one of the classes 2A, 2B, already the degree 36 permutation character yields $P(g,s) \geq 1/3$. And for all other elements *g*, again we have $\sigma(g,s) < 1/3$.

```
─────────────────── Example ───────────────────
gap> spos15:= Position( ord, 15 );;
gap> approx15:= ApproxP( prim, spos15 );;
gap> filt15:= PositionsProperty( approx15, x -> x >= 1/3 );
[ 2, 3 ]
gap> PositionsProperty( ApproxP( prim{ [ 2 ] }, spos15 ), x -> x >= 1/3 );
[ 2, 3 ]
gap> AtlasClassNames( t ){ filt15 };
[ "2A", "2B" ]
gap> approx15{ filt15 };
[ 46/63, 8/21 ]
```

For the remaining statements, we use explicit computations with *S*, in the transitive degree 63 permutation representation. We start with a function that computes a transvection in $S_d(2)$; note that the invariant bilinear form used for symplectic groups in GAP is described by a matrix with nonzero entries exactly in the positions $(i, d+1-i)$, for $1 \leq i \leq d$.

```
─────────────────── Example ───────────────────
gap> transvection:= function( d )
>       local mat;
>       mat:= IdentityMat( d, Z(2) );
>       mat{ [ 1, d ] }{ [ 1, d ] }:= [ [ 0, 1 ], [ 1, 0 ] ] * Z(2);
>       return mat;
> end;;
```

First we compute, for statement (d), the exact values $P(g,s)$ for *g* in one of the classes 2A or 2B, and *s* of order 15. Note that the classes 2A, 2B are the unique classes of the lengths 63 and 315, respectively.

```
─────────────────── Example ───────────────────
gap> PositionsProperty( SizesConjugacyClasses( t ), x -> x in [ 63, 315 ] );
[ 2, 3 ]
gap> d:= 6;;
gap> matgrp:= Sp(d,2);;
gap> hom:= ActionHomomorphism( matgrp, NormedRowVectors( GF(2)^d ) );;
gap> g:= Image( hom, matgrp );;
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s15:= Random( g );
>      until Order( s15 ) = 15;
gap> 2A:= Image( hom, transvection( d ) );;
gap> Size( ConjugacyClass( g, 2A ) );
63
gap> IsTransitive( g, MovedPoints( g ) );
true
gap> RatioOfNongenerationTransPermGroup( g, 2A, s15 );
11/21
gap> repeat 12C:= Random( g );
>      until Order( 12C ) = 12 and Size( Centralizer( g, 12C ) ) = 12;
gap> 2B:= 12C^6;;
```

```
gap> Size( ConjugacyClass( g, 2B ) );
315
gap> RatioOfNongenerationTransPermGroup( g, 2B, s15 );
8/21
```

For statement (e), we compute $P(g, s')$, for a transvection g and class representatives s' of S. It turns out that the minimum is $11/21$, and it is attained for exactly one s'; by the above, this element has order 15.

```
─────────────────────── Example ───────────────────────
gap> ccl:= ConjugacyClasses( g );;
gap> reps:= List( ccl, Representative );;
gap> nongen2A:= List( reps,
>        x -> RatioOfNongenerationTransPermGroup( g, 2A, x ) );;
gap> min:= Minimum( nongen2A );
11/21
gap> Number( nongen2A, x -> x = min );
1
```

For statement (f), we show that for any choice of s', at least two of the values $P(g, s')$, with g in the classes 2A, 2B, or 3A, are larger than $1/3$.

```
─────────────────────── Example ───────────────────────
gap> nongen2B:= List( reps,
>        x -> RatioOfNongenerationTransPermGroup( g, 2B, x ) );;
gap> 3A:= s15^5;;
gap> nongen3A:= List( reps,
>        x -> RatioOfNongenerationTransPermGroup( g, 3A, x ) );;
gap> bad:= List( [ 1 .. NrConjugacyClasses( t ) ],
>               i -> Number( [ nongen2A, nongen2B, nongen3A ],
>                            x -> x[i] > 1/3 ) );;
gap> Minimum( bad );
2
```

Finally, for statement (g), we have to consider only the case that the two elements x, y are transvections.

```
─────────────────────── Example ───────────────────────
gap> PositionsProperty( approx9, x -> x + approx9[2] >= 1 );
[ 2 ]
```

We use the random approach described in Section 11.3.3.

```
─────────────────────── Example ───────────────────────
gap> repeat s9:= Random( g );
>    until Order( s9 ) = 9;
gap> RandomCheckUniformSpread( g, [ 2A, 2A ], s9, 20 );
true
```

11.5.21 $S_8(2)$

We show that the group $S = S_8(2)$ satisfies the following.

(a) For $s \in S$ of order 17, $\mathbb{M}(S,s)$ consists of one subgroup of each of the types $O_8^-(2).2 = \Omega^-(8,2).2$, $S_4(4).2 = \mathrm{Sp}(4,4).2$, and $L_2(17) = \mathrm{PSL}(2,17)$.

(b) For $s \in S$ of order 17, and $g \in S^\times$, we have $P(g,s) < 1/3$, except if g is a transvection.

(c) The uniform spread of S is at least two, with s of order 17.

Statement (a) follows from the list of maximal subgroups of S in [CCN$^+$85, p. 123], and the fact that $1_H^S(s) = 1$ holds for each $H \in \mathbb{M}(S,s)$. Note that 17 divides the indices of the maximal subgroups of the types $O_8^+(2).2$ and $2^7 : S_6(2)$ in S, and obviously 17 does not divide the orders of the remaining maximal subgroups.

The permutation characters induced from the first two subgroups are uniquely determined by the ordinary character tables. The permutation character induced from the last subgroup is uniquely determined if one considers also the corresponding Brauer tables; the correct class fusion is stored in the GAP Character Table Library, see [Brea].

```
———————————————————————— Example ————————————————————————
  gap> t:= CharacterTable( "S8(2)" );;
  gap> pi1:= PossiblePermutationCharacters( CharacterTable( "O8-(2).2" ), t );;
  gap> pi2:= PossiblePermutationCharacters( CharacterTable( "S4(4).2" ), t );;
  gap> pi3:= [ TrivialCharacter( CharacterTable( "L2(17)" ) )^t ];;
  gap> prim:= Concatenation( pi1, pi2, pi3 );;
  gap> Length( prim );
  3
  gap> spos:= Position( OrdersClassRepresentatives( t ), 17 );;
  gap> List( prim, x -> x[ spos ] );
  [ 1, 1, 1 ]
```

For statement (b), we observe that $\sigma(g,s) < 1/3$ if g is not a transvection, and that $P(g,s) = \sigma(g,s)$ for transvections g because exactly one of the three permutation characters is nonzero on both s and the class of transvections.

```
———————————————————————— Example ————————————————————————
  gap> approx:= ApproxP( prim, spos );;
  gap> PositionsProperty( approx, x -> x >= 1/3 );
  [ 2 ]
  gap> Number( prim, pi -> pi[2] <> 0 and pi[ spos ] <> 0 );
  1
  gap> approx[2];
  8/15
```

In statement (c), we have to consider only the case that the two elements x, y are transvections.

```
———————————————————————— Example ————————————————————————
  gap> PositionsProperty( approx, x -> x + approx[2] >= 1 );
  [ 2 ]
```

We use the random approach described in Section 11.3.3.

```
———————————————————————— Example ————————————————————————
  gap> d:= 8;;
  gap> matgrp:= Sp(d,2);;
  gap> hom:= ActionHomomorphism( matgrp, NormedRowVectors( GF(2)^d ) );;
```

```
gap> x:= Image( hom, transvection( d ) );;
gap> g:= Image( hom, matgrp );;
gap> C:= ConjugacyClass( g, x );;  Size( C );
255
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g );
>    until Order( s ) = 17;
gap> RandomCheckUniformSpread( g, [ x, x ], s, 20 );
true
```

11.5.22 $* S_{10}(2)$

We show that the group $S = S_{10}(2)$ satisfies the following.

(a)   For $s \in S$ of order 33, $\mathbb{M}(S,s)$ consists of one subgroup of each of the types $\Omega^-(10,2).2$ and $L_2(32).5 = \mathrm{Sp}(2,32).5$.

(b)   For $s \in S$ of order 33, and $g \in S^\times$, we have $P(g,s) < 1/3$, except if g is a transvection.

(c)   The uniform spread of S is at least two, with s of order 33.

By [Ber00], the only maximal subgroups of S that contain s have the types stated in (a), and by [KL90, Prop. 4.3.10 and 4.8.6], there is exactly one class of each of these subgroups.

We compute the values $\sigma(g,s)$, for all $g \in S^\times$.

```
―――――――――――――――――― Example ――――――――――――――――――
gap> t:= CharacterTable( "S10(2)" );;
gap> pi1:= PossiblePermutationCharacters( CharacterTable( "O10-(2).2" ), t );;
gap> pi2:= PossiblePermutationCharacters( CharacterTable( "L2(32).5" ), t );;
gap> prim:= Concatenation( pi1, pi2 );;  Length( prim );
2
gap> spos:= Position( OrdersClassRepresentatives( t ), 33 );;
gap> approx:= ApproxP( prim, spos );;
```

For statement (b), we observe that $\sigma(g,s) < 1/3$ if g is not a transvection, and that $P(g,s) = \sigma(g,s)$ for transvections g because exactly one of the two permutation characters is nonzero on both s and the class of transvections.

```
―――――――――――――――――― Example ――――――――――――――――――
gap> PositionsProperty( approx, x -> x >= 1/3 );
[ 2 ]
gap> Number( prim, pi -> pi[2] <> 0 and pi[ spos ] <> 0 );
1
gap> approx[2];
16/31
```

In statement (c), we have to consider only the case that the two elements x, y are transvections. We use the random approach described in Section 11.3.3.

```
―――――――――――――――――― Example ――――――――――――――――――
gap> d:= 10;;
gap> matgrp:= Sp(d,2);;
gap> hom:= ActionHomomorphism( matgrp, NormedRowVectors( GF(2)^d ) );;
```

```
gap> x:= Image( hom, transvection( d ) );;
gap> g:= Image( hom, matgrp );;
gap> C:= ConjugacyClass( g, x );;  Size( C );
1023
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g );
>    until Order( s ) = 33;
gap> RandomCheckUniformSpread( g, [ x, x ], s, 20 );
true
```

11.5.23 $U_4(2)$

We show that $S = U_4(2) = \mathrm{SU}(4,2) \cong S_4(3) = \mathrm{PSp}(4,3)$ satisfies the following.

(a) $\sigma(S) = 21/40$, and this value is attained exactly for $\sigma(S,s)$ with s of order 12.

(b) For $s \in S$ of order 9, $\mathbb{M}(S,s)$ consists of two groups, of the types $3^{1+2}_+ {:} 2A_4 = \mathrm{GU}(3,2)$ and $3^3 {:} S_4$, respectively.

(c) $P(S) = 2/5$, and this value is attained exactly for $P(S,s)$ with s of order 9.

(d) The uniform spread of S is at least three, with s of order 9.

(e) $\sigma'(\mathrm{Aut}(S),s) = 7/20$.

(Note that in this example, the optimal choice of s w.r.t. $\sigma(S,s)$ is not optimal w.r.t. $P(S,s)$.)
Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

``` 
———————————————————————— Example ————————————————————————
gap> t:= CharacterTable( "U4(2)" );;
gap> ProbGenInfoSimple( t );
[ "U4(2)", 21/40, 1, [ "12A" ], [ 2 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the only classes of maximal subgroups that contain elements of order 9 consist of groups of the structures $3^{1+2}_+ : 2A_4$ and $3^3 : S_4$, see [CCN+85, p. 26].

```
———————————————————————— Example ————————————————————————
gap> OrdersClassRepresentatives( t );
[ 1, 2, 2, 3, 3, 3, 3, 4, 4, 5, 6, 6, 6, 6, 6, 6, 9, 9, 12, 12 ]
gap> prim:= PrimitivePermutationCharacters( t );
[ Character( CharacterTable( "U4(2)" ),
  [ 27, 3, 7, 0, 0, 9, 0, 3, 1, 2, 0, 0, 3, 3, 0, 1, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "U4(2)" ),
  [ 36, 12, 8, 0, 0, 6, 3, 0, 2, 1, 0, 0, 0, 3, 2, 0, 0, 0, 0, 0 ] ),
  Character( CharacterTable( "U4(2)" ),
  [ 40, 8, 0, 13, 13, 4, 4, 4, 0, 0, 5, 5, 2, 2, 2, 0, 1, 1, 1, 1 ] ),
  Character( CharacterTable( "U4(2)" ),
  [ 40, 16, 4, 4, 4, 1, 7, 0, 2, 0, 4, 4, 1, 1, 1, 1, 1, 1, 0, 0 ] ),
  Character( CharacterTable( "U4(2)" ),
  [ 45, 13, 5, 9, 9, 6, 3, 1, 1, 0, 1, 1, 4, 4, 1, 2, 0, 0, 1, 1 ] ) ]
```

For statement (c), we use a primitive permutation representation on 40 points that occurs in the natural action of SU(4,2).

```
——————————————————— Example ———————————————————
gap> g:= SU(4,2);;
gap> orbs:= OrbitsDomain( g, NormedRowVectors( GF(4)^4 ), OnLines );;
gap> List( orbs, Length );
[ 45, 40 ]
gap> g:= Action( g, orbs[2], OnLines );;
```

First we show that for s of order 9, $P(S,s) = 2/5$ holds. For that, we have to consider only $P(g,s)$, with g in one of the classes 2A (of length 45) and 3A (of length 40); since the class 3B contains the inverses of the elements in the class 3A, we need not test it.

```
——————————————————— Example ———————————————————
gap> spos:= Position( OrdersClassRepresentatives( t ), 9 );
17
gap> approx:= ApproxP( prim, spos );
[ 0, 3/5, 1/10, 17/40, 17/40, 1/8, 11/40, 1/10, 1/20, 0, 9/40, 9/40,
  3/40, 3/40, 3/40, 1/40, 1/20, 1/20, 1/40, 1/40 ]
gap> badpos:= PositionsProperty( approx, x -> x >= 2/5 );
[ 2, 4, 5 ]
gap> PowerMap( t, 2 )[4];
5
gap> OrdersClassRepresentatives( t );
[ 1, 2, 2, 3, 3, 3, 3, 4, 4, 5, 6, 6, 6, 6, 6, 6, 9, 9, 12, 12 ]
gap> SizesConjugacyClasses( t );
[ 1, 45, 270, 40, 40, 240, 480, 540, 3240, 5184, 360, 360, 720, 720,
  1440, 2160, 2880, 2880, 2160, 2160 ]
```

A representative g of a class of length 40 can be found as the third power of any order 9 element.

```
——————————————————— Example ———————————————————
gap> PowerMap( t, 3 )[ spos ];
4
gap> ResetGlobalRandomNumberGenerators();
gap> repeat s:= Random( g );
>    until Order( s ) = 9;
gap> Size( ConjugacyClass( g, s^3 ) );
40
gap> prop:= RatioOfNongenerationTransPermGroup( g, s^3, s );
13/40
```

Next we examine g in the class 2A.

```
——————————————————— Example ———————————————————
gap> repeat x:= Random( g ); until Order( x ) = 12;
gap> Size( ConjugacyClass( g, x^6 ) );
45
gap> prop:= RatioOfNongenerationTransPermGroup( g, x^6, s );
2/5
```

Finally, we compute that for s of order different from 9 and g in the class 2A, $P(g,s)$ is larger than $2/5$.

```
  _____ Example _____
 gap> ccl:= List( ConjugacyClasses( g ), Representative );;
 gap> SortParallel( List( ccl, Order ), ccl );
 gap> List( ccl, Order );
 [ 1, 2, 2, 3, 3, 3, 3, 4, 4, 5, 6, 6, 6, 6, 6, 6, 9, 9, 12, 12 ]
 gap> prop:= List( ccl, r -> RatioOfNongenerationTransPermGroup( g, x^6, r ) );
 [ 1, 1, 1, 1, 1, 1, 1, 1, 1, 5/9, 1, 1, 1, 1, 1, 1, 2/5, 2/5, 7/15,
   7/15 ]
 gap> Minimum( prop );
 2/5
```

In order to show statement (d), we have to consider triples (x_1, x_2, x_3) with x_i of prime order and $\sum_{i=1}^{3} P(x_i, s) \geq 1$. This means that it suffices to check x in the class 2A, y in 2A∪3A, and z in 2A∪3A∪3D.

```
  _____ Example _____
 gap> approx[2]:= 2/5;;
 gap> approx[4]:= 13/40;;
 gap> primeord:= PositionsProperty( OrdersClassRepresentatives( t ),
 >                                  IsPrimeInt );
 [ 2, 3, 4, 5, 6, 7, 10 ]
 gap> RemoveSet( primeord, 5 );
 gap> primeord;
 [ 2, 3, 4, 6, 7, 10 ]
 gap> approx{ primeord };
 [ 2/5, 1/10, 13/40, 1/8, 11/40, 0 ]
 gap> AtlasClassNames( t ){ primeord };
 [ "2A", "2B", "3A", "3C", "3D", "5A" ]
 gap> triples:= Filtered( UnorderedTuples( primeord, 3 ),
 >                t -> Sum( approx{ t } ) >= 1 );
 [ [ 2, 2, 2 ], [ 2, 2, 4 ], [ 2, 2, 7 ], [ 2, 4, 4 ], [ 2, 4, 7 ] ]
```

We use the random approach described in Section 11.3.3.

```
  _____ Example _____
 gap> repeat 6E:= Random( g );
 >    until Order( 6E ) = 6 and Size( Centralizer( g, 6E ) ) = 18;
 gap> 2A:= 6E^3;;
 gap> 3A:= s^3;;
 gap> 3D:= 6E^2;;
 gap> RandomCheckUniformSpread( g, [ 2A, 2A, 2A ], s, 50 );
 true
 gap> RandomCheckUniformSpread( g, [ 2A, 2A, 3A ], s, 50 );
 true
 gap> RandomCheckUniformSpread( g, [ 3D, 2A, 2A ], s, 50 );
 true
 gap> RandomCheckUniformSpread( g, [ 2A, 3A, 3A ], s, 50 );
 true
 gap> RandomCheckUniformSpread( g, [ 3D, 3A, 2A ], s, 50 );
 true
```

Statement (e) can be proved using `ProbGenInfoAlmostSimple`, cf. Section 11.4.5.

```
——————————————————————————— Example ———————————————————————————
  gap> t:= CharacterTable( "U4(2)" );;
  gap> t2:= CharacterTable( "U4(2).2" );;
  gap> spos:= PositionsProperty( OrdersClassRepresentatives( t ), x -> x = 9 );;
  gap> ProbGenInfoAlmostSimple( t, t2, spos );
  [ "U4(2).2", 7/20, [ "9AB" ], [ 2 ] ]
```

11.5.24 $U_4(3)$

We show that $S = U_4(3) = \mathrm{PSU}(4,3)$ satisfies the following.

(a) $\sigma(S) = 53/153$, and this value is attained exactly for $\sigma(S,s)$ with s of order 7.

(b)   For $s \in S$ of order 7, $\mathbb{M}(S,s)$ consists of two nonconjugate groups of the type $L_3(4)$, one group of the type $U_3(3)$, and four pairwise nonconjugate groups of the type A_7.

(c) $P(S) = 43/135$, and this value is attained exactly for $P(S,s)$ with s of order 7.

(d)   The uniform spread of S is at least three, with s of order 7.

(e)   The preimage of s in the matrix group $\mathrm{SU}(4,3) \cong 4.U_4(3)$ has order 28, the preimages of the groups in $\mathbb{M}(S,s)$ have the structures $4_2.L_3(4)$, $4 \times U_3(3) \cong \mathrm{GU}(3,3)$, and $4.A_7$ (the latter being a central product of a cyclic group of order four and $2.A_7$).

(f) $P'(S.2_1,s) = 13/27$, $\sigma'(S.2_2) = 1/3$, and $\sigma'(S.2_3) = 31/162$, with s of order 7 in each case.

Statement (a) follows from inspection of the primitive permutation characters, cf. Section 11.4.4.

```
——————————————————————————— Example ———————————————————————————
  gap> t:= CharacterTable( "U4(3)" );;
  gap> ProbGenInfoSimple( t );
  [ "U4(3)", 53/135, 2, [ "7A" ], [ 7 ] ]
```

Statement (b) can be read off from the permutation characters, and the fact that the only classes of maximal subgroups that contain elements of order 7 consist of groups of the structures as claimed, see [CCN+85, p. 52].

```
——————————————————————————— Example ———————————————————————————
  gap> prim:= PrimitivePermutationCharacters( t );;
  gap> spos:= Position( OrdersClassRepresentatives( t ), 7 );
  13
  gap> List( Filtered( prim, x -> x[ spos ] <> 0 ), l -> l{ [ 1, spos ] } );
  [ [ 162, 1 ], [ 162, 1 ], [ 540, 1 ], [ 1296, 1 ], [ 1296, 1 ],
    [ 1296, 1 ], [ 1296, 1 ] ]
```

In order to show statement (c) (which then implies statement (d)), we use a permutation representation on 112 points. It corresponds to an orbit of one-dimensional subspaces in the natural module of $\Omega^-(6,3) \cong S$.

```
——————————————————————————— Example ———————————————————————————
  gap> matgrp:= DerivedSubgroup( SO( -1, 6, 3 ) );;
  gap> orbs:= OrbitsDomain( matgrp, NormedRowVectors( GF(3)^6 ), OnLines );;
  gap> List( orbs, Length );
  [ 126, 126, 112 ]
  gap> G:= Action( matgrp, orbs[3], OnLines );;
```

It is sufficient to compute $P(g,s)$, for involutions $g \in S$.

```
                          ── Example ──────────
 gap> approx:= ApproxP( prim, spos );
 [ 0, 53/135, 1/10, 1/24, 1/24, 7/45, 4/45, 1/27, 1/36, 1/90, 1/216,
   1/216, 7/405, 7/405, 1/270, 0, 0, 0, 0, 1/270 ]
 gap> Filtered( approx, x -> x >= 43/135 );
 [ 53/135 ]
 gap> OrdersClassRepresentatives( t );
 [ 1, 2, 3, 3, 3, 3, 4, 4, 5, 6, 6, 6, 7, 7, 8, 9, 9, 9, 9, 12 ]
 gap> ResetGlobalRandomNumberGenerators();
 gap> repeat g:= Random( G ); until Order(g) = 2;
 gap> repeat s:= Random( G );
 >    until Order(s) = 7;
 gap> bad:= RatioOfNongenerationTransPermGroup( G, g, s );
 43/135
 gap> bad < 1/3;
 true
```

Statement (e) can be shown easily with character-theoretic methods, as follows. Since $\mathrm{SU}(4,3)$ is a Schur cover of S and the groups in $\mathbb{M}(S,s)$ are simple, only very few possibilities have to be checked. The Schur multiplier of $U_3(3)$ is trivial (see, e. g., [CCN+85, p. 14]), so the preimage in $\mathrm{SU}(4,3)$ is a direct product of $U_3(3)$ and the centre of $\mathrm{SU}(4,3)$. Neither $L_3(4)$ nor its double cover $2.L_3(4)$ can be a subgroup of $\mathrm{SU}(4,3)$, so the preimage of $L_3(4)$ must be a Schur cover of $L_3(4)$, i. e., it must have either the type $4_1.L_3(4)$ or $4_2.L_3(4)$ (see [CCN+85, p. 23]); only the type $4_2.L_3(4)$ turns out to be possible.

```
                          ── Example ──────────
 gap> 4t:= CharacterTable( "4.U4(3)" );;
 gap> Length( PossibleClassFusions( CharacterTable( "L3(4)" ), 4t ) );
 0
 gap> Length( PossibleClassFusions( CharacterTable( "2.L3(4)" ), 4t ) );
 0
 gap> Length( PossibleClassFusions( CharacterTable( "4_1.L3(4)" ), 4t ) );
 0
 gap> Length( PossibleClassFusions( CharacterTable( "4_2.L3(4)" ), 4t ) );
 4
```

As for the preimage of the A_7 type subgroups, we first observe that the double cover of A_7 cannot be a subgroup of the double cover of S, so the preimage of A_7 in the double cover of $U_4(3)$ is a direct product $2 \times A_7$. The group $\mathrm{SU}(4,3)$ does not contain A_7 type subgroups, thus the A_7 type subgroups in $2.U_4(3)$ lift to double covers of A_7 in $\mathrm{SU}(4,3)$. This proves the claimed structure.

```
                          ── Example ──────────
 gap> 2t:= CharacterTable( "2.U4(3)" );;
 gap> Length( PossibleClassFusions( CharacterTable( "2.A7" ), 2t ) );
 0
 gap> Length( PossibleClassFusions( CharacterTable( "A7" ), 4t ) );
 0
```

For statement (f), we consider automorphic extensions of S. The bound for $S.2_3$ has been computed in Section 11.4.5. That for $S.2_2$ can be computed form the fact that the classes of maximal subgroups of $S.2_2$ containing s of order 7 are S, one class of $U_3(3).2$ type subgroups, and two classes of S_7 type subgroups which induce the same permutation character (see [CCN+85, p. 52]).

```
                        ─────── Example ───────
  gap> t2:= CharacterTable( "U4(3).2_2" );;
  gap> pi1:= PossiblePermutationCharacters( CharacterTable( "U3(3).2" ), t2 );
  [ Character( CharacterTable( "U4(3).2_2" ),
    [ 540, 12, 54, 0, 0, 9, 8, 0, 0, 6, 0, 0, 1, 2, 0, 0, 0, 2, 0, 24,
        4, 0, 0, 0, 0, 0, 0, 3, 2, 0, 4, 0, 0, 0 ] ) ]
  gap> pi2:= PossiblePermutationCharacters( CharacterTable( "A7.2" ), t2 );
  [ Character( CharacterTable( "U4(3).2_2" ),
    [ 1296, 48, 0, 27, 0, 9, 0, 4, 1, 0, 3, 0, 1, 0, 0, 0, 0, 0, 216,
        24, 0, 4, 0, 0, 0, 9, 0, 3, 0, 1, 0, 1, 0, 0 ] ) ]
  gap> prim:= Concatenation( pi1, pi2, pi2 );;
  gap> outer:= Difference(
  >       PositionsProperty( OrdersClassRepresentatives( t2 ), IsPrimeInt ),
  >       ClassPositionsOfDerivedSubgroup( t2 ) );;
  gap> spos:= Position( OrdersClassRepresentatives( t2 ), 7 );;
  gap> Maximum( ApproxP( prim, spos ){ outer } );
  1/3
```

Finally, Section 11.4.5 shows that the character tables are not sufficient for what we need, so we compute the exact proportion of nongeneration for $U_4(3).2_1 \cong SO^-(6,3)$.

```
                        ─────── Example ───────
  gap> matgrp:= SO( -1, 6, 3 );
  SO(-1,6,3)
  gap> orbs:= OrbitsDomain( matgrp, NormedRowVectors( GF(3)^6 ), OnLines );;
  gap> List( orbs, Length );
  [ 126, 126, 112 ]
  gap> G:= Action( matgrp, orbs[3], OnLines );;
  gap> repeat s:= Random( G );
  >    until Order( s ) = 7;
  gap> repeat
  >       repeat 2B:= Random( G ); until Order( 2B ) mod 2 = 0;
  >       2B:= 2B^( Order( 2B ) / 2 );
  >       c:= Centralizer( G, 2B );
  >    until Size( c ) = 12096;
  gap> RatioOfNongenerationTransPermGroup( G, 2B, s );
  13/27
  gap> repeat
  >       repeat 2C:= Random( G ); until Order( 2C ) mod 2 = 0;
  >       2C:= 2C^( Order( 2C ) / 2 );
  >       c:= Centralizer( G, 2C );
  >    until Size( c ) = 1440;
  gap> RatioOfNongenerationTransPermGroup( G, 2C, s );
  0
```

11.5.25 $U_6(3)$

We show that $S = U_6(3) = PSU(6,3)$ satisfies the following.

(a)   For $s \in S$ of the type $1 \perp 5$ (i. e., the preimage of s in $2.S = SU(6,3)$ decomposes the natural 6-dimensional module for $2.S$ into an orthogonal sum of two irreducible modules of the dimensions 1 and 5, respectively) and of order $(3^5 + 1)/2 = 122$, $\mathbb{M}(S,s)$ consists of one group of the

type $2 \times U_5(3)$, which lifts to a subgroup of the type $4 \times U_5(3) = \mathrm{GU}(5,3)$ in $2.S$. (The preimage of s in $2.S$ has order $3^5 + 1 = 244$.)

(b) $\sigma(S,s) = 353/3\,159$.

By [MSW94], the only maximal subgroup of S that contains s is the stabilizer $H \cong 2 \times U_5(3)$ of the orthogonal decomposition. This proves statement (a).

The character table of S is currently not available in the GAP Character Table Library. We consider the permutation action of S on the orbit of the stabilized 1-space. So M can be taken as a point stabilizer in this action.

```
──────── Example ────────
gap> CharacterTable( "U6(3)" );
fail
gap> g:= SU(6,3);;
gap> orbs:= OrbitsDomain( g, NormedRowVectors( GF(9)^6 ), OnLines );;
gap> List( orbs, Length );
[ 22204, 44226 ]
gap> repeat x:= PseudoRandom( g ); until Order( x ) = 244;
gap> List( orbs, o -> Number( o, v -> OnLines( v, x ) = v ) );
[ 0, 1 ]
gap> g:= Action( g, orbs[2], OnLines );;
gap> M:= Stabilizer( g, 1 );;
```

Then we compute a list of elements in M that covers the conjugacy classes of prime element order, from which the numbers of fixed points and thus $\max\{\mu(S/M,g); g \in M^\times\} = \sigma(S,s)$ can be derived. This way we avoid completely to check the S-conjugacy of elements (class representatives of Sylow subgroups in M).

```
──────── Example ────────
gap> elms:= [];;
gap> for p in PrimeDivisors( Size( M ) ) do
>       syl:= SylowSubgroup( M, p );
>       Append( elms, Filtered( PcConjugacyClassReps( syl ),
>                               r -> Order( r ) = p ) );
>    od;
gap> 1 - Minimum( List( elms, NrMovedPoints ) ) / Length( orbs[2] );
353/3159
```

11.5.26 $U_8(2)$

We show that $S = U_8(2) = \mathrm{SU}(8,2)$ satisfies the following.

(a) For $s \in S$ of the type $1 \perp 7$ (i. e., s decomposes the natural 8-dimensional module for S into an orthogonal sum of two irreducible modules of the dimensions 1 and 7, respectively) and of order $2^7 + 1 = 129$, $\mathbb{M}(S,s)$ consists of one group of the type $3 \times U_7(2) = \mathrm{GU}(7,2)$.

(b) $\sigma(S,s) = 2\,753/10\,880$.

By [MSW94], the only maximal subgroup of S that contains s is the stabilizer $M \cong \mathrm{GU}(7,2)$ of the orthogonal decomposition. This proves statement (a).

The character table of S is currently not available in the GAP Character Table Library. We proceed exactly as in Section 11.5.25 in order to prove statement (b).

--- Example ---

```
gap> CharacterTable( "U8(2)" );
fail
gap> g:= SU(8,2);;
gap> orbs:= OrbitsDomain( g, NormedRowVectors( GF(4)^8 ), OnLines );;
gap> List( orbs, Length );
[ 10965, 10880 ]
gap> repeat x:= PseudoRandom( g ); until Order( x ) = 129;
gap> List( orbs, o -> Number( o, v -> OnLines( v, x ) = v ) );
[ 0, 1 ]
gap> g:= Action( g, orbs[2], OnLines );;
gap> M:= Stabilizer( g, 1 );;
gap> elms:= [];;
gap> for p in PrimeDivisors( Size( M ) ) do
>      syl:= SylowSubgroup( M, p );
>      Append( elms, Filtered( PcConjugacyClassReps( syl ),
>                          r -> Order( r ) = p ) );
>    od;
gap> Length( elms );
611
gap> 1 - Minimum( List( elms, NrMovedPoints ) ) / Length( orbs[2] );
2753/10880
```

References

[Ber00]   Á. Bereczky. Maximal overgroups of Singer elements in classical groups. *J. Algebra*, 234(1):187–206, 2000. 408, 409, 412, 425, 473, 475, 479, 485

[BG]   T. Breuer and R. M. Guralnick. Finite groups can be generated by a pi-subgroup and a pi'-subgroup. arXiv:2103.17216. 293, 294, 296, 300, 301

[BGK08]   T. Breuer, R. M. Guralnick, and W. M. Kantor. Probabilistic generation of finite simple groups, II. *J. Algebra*, 320(2):443–494, 2008. 174, 366, 369, 371, 390, 395, 398, 400, 402, 408, 410, 415, 425, 427, 428, 429, 433, 453, 470, 471, 473, 475, 477

[BGL⁺10]   T. Breuer, R. M. Guralnick, A. Lucchini, A. Maróti, and G. P. Nagy. Hamiltonian cycles in the generating graphs of finite groups. *Bull. London Math. Soc.*, 42(4):621–633, 2010. 172, 173, 174, 184

[BGS11]   T. C. Burness, R. M. Guralnick, and J. Saxl. On base sizes for symmetric groups. *Bull. Lond. Math. Soc.*, 43(2):386–391, 2011. 197, 198

[BMO17]   T. Breuer, G. Malle, and E. A. O'Brien. Reliability and reproducibility of Atlas information. In M. Bhargava, R. Guralnick, G. Hiss, K. Lux, and P. H. Tiep, editors, *Finite simple groups: thirty years of the atlas and beyond*, volume 694 of *Contemp. Math.*, page 21–31. Amer. Math. Soc., Providence, RI, 2017. 14

[BN95]   T. Breuer and S. P. Norton. *Improvements to the Atlas*, page 297–327. Volume 11 of *London Mathematical Society Monographs. New Series* [JLPW95], 1995. Appendix 2 by T. Breuer and S. Norton, Oxford Science Publications. 79, 216, 226, 238, 330

[BP98]   T. Breuer and G. Pfeiffer. Finding possible permutation characters. *J. Symbolic Comput.*, 26(3):343–354, 1998. 243, 246, 247, 264, 265, 409, 413

[Brea]   T. Breuer. Ambiguous class fusions in the GAP character table library. https://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib/doc2/manual.pdf. 484

[Breb]   T. Breuer. GAP computations concerning probabilistic generation of finite simple groups. arXiv:0710.3267. 174, 176, 178

[Brec]   T. Breuer. Using table automorphisms for constructing character tables in GAP. https://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib/doc2/manual.pdf. 406

[Bre91]   T. Breuer. Potenzabbildungen, Untergruppenfusionen, Tafel-Automorphismen. Diplo-marbeit, Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, 1991. 15

[Bre11]   T. Breuer. Computing character tables of groups of type *M.G.A*. *LMS J. Comput. Math.*, 14:173–178, 2011. 57

[Bre25]   T. Breuer. The GAP Character Table Library, Version 1.3.11. `https://www.math.rwth-aachen.de/~Thomas.Breuer/ctbllib`, May 2025. GAP package. 51, 162, 173, 178, 209, 213, 235, 239, 243, 303, 326, 352, 367

[BW75]   J. L. Brenner and J. Wiegold. Two-generator groups. I. *Michigan Math. J.*, 22:53–64, 1975. 400, 417, 418, 420, 436

[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. 45, 53, 55, 77, 81, 86, 87, 88, 91, 92, 93, 94, 97, 100, 107, 125, 126, 130, 131, 132, 142, 152, 159, 170, 197, 198, 209, 210, 212, 213, 216, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 229, 231, 233, 235, 236, 237, 238, 243, 248, 249, 251, 252, 257, 260, 261, 262, 264, 265, 266, 268, 269, 270, 273, 277, 278, 282, 304, 308, 311, 319, 320, 321, 322, 323, 343, 369, 393, 394, 395, 397, 400, 401, 402, 403, 404, 405, 406, 414, 415, 417, 419, 422, 427, 437, 440, 442, 445, 448, 449, 450, 453, 455, 462, 463, 465, 467, 481, 484, 486, 489, 490

[CP96]   J. J. Cannon and C. Playoust. An introduction to algebraic programming in Magma. `http://www.math.usyd.edu.au:8000/u/magma`, 1996. 13, 14

[Dad66]   E. C. Dade. Blocks with cyclic defect groups. *Ann. of Math. (2)*, 84:20–48, 1966. 43

[DLP25]   H. Dietrich, M. Lee, and T. Popiel. The maximal subgroups of the monster. Advances in Mathematics 469 (2025), article 110214, 2025. 172, 184, 185, 230, 300

[DLPP24]   H. Dietrich, M. Lee, A. Pisani, and T. Popiel. Explicit construction of the maximal subgroups of the monster. arXiv:2411.12230, 2024. 348, 390

[DNT13]   S. Dolfi, G. Navarro, and P. H. Tiep. Finite groups whose same degree characters are Galois conjugate. *Israel J. Math.*, 198(1):283–331, 2013. 8, 352

[Fei82]   W. Feit. *The representation theory of finite groups*, volume 25 of *North-Holland Mathematical Library*. North-Holland Publishing Co., 1982. xiv+502 pp., ISBN 0-444-86155-6. 57

[Gag86]   S. M. Gagola, Jr. Formal character tables. *Michigan Math. J.*, 33(1):3–10, 1986. 10, 105

[GAP24]   GAP – Groups, Algorithms, and Programming, Version 4.13.1. `https://www.gap-system.org`, Mar 2024. 51, 162, 172, 209, 213, 235, 243, 303, 352, 366

[GJMS89]   R. L. Griess Jr., U. Meierfrankenfeld, and Y. Segev. A uniqueness proof for the Monster. *Ann. of Math. (2)*, 130(3):567–602, 1989. 290

[GK00]   R. M. Guralnick and W. M. Kantor. Probabilistic generation of finite simple groups. *J. Algebra*, 234(2):743–792, 2000. Special issue in honor of Helmut Wielandt. 371, 477

[GM01]   S. Ganief and J. Moori. On the spread of the sporadic simple groups. *Comm. Algebra*, 29(8):3239–3255, 2001. 390

[GPPS99] R. Guralnick, T. Penttila, C. E. Praeger, and J. Saxl. Linear groups with orders having certain large prime divisors. *Proc. London Math. Soc.*, 78(1):167–214, 1999. 406, 407, 411

[HL89]   G. Hiss and K. Lux. *Brauer trees of sporadic groups*. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1989. 40, 43

[HL94]   G. Hiss and K. Lux. The 5-modular characters of the sporadic simple Fischer groups Fi_{22} and Fi_{23}. *Comm. Algebra*, 22(9):3563–3590, 1994. With an appendix by Thomas Breuer. 59

[Hol08]   D. Holt. cohomolo, computing cohomology groups and Schur multipliers, Version 1.6. http://www.maths.warwick.ac.uk/~dfh/cohomolo, 2008. GAP package. 352

[HP89]   D. F. Holt and W. Plesken. *Perfect groups*. Oxford Mathematical Monographs. The Clarendon Press Oxford University Press, New York, 1989. With an appendix by W. Hanrath, Oxford Science Publications. 10, 13, 16, 17, 18, 19, 21, 22, 23, 25, 26, 27, 28

[Hul05]   A. Hulpke. Constructing transitive permutation groups. *J. Symbolic Comput.*, 39(1):1–30, 2005. 415

[Hup67]   B. Huppert. *Endliche Gruppen. I*. Die Grundlehren der Mathematischen Wissenschaften, Band 134. Springer-Verlag, Berlin, 1967. 194, 264

[HW04]   P. E. Holmes and R. A. Wilson. $PSL_2(59)$ is a subgroup of the Monster. *J. London Math. Soc.*, 69(1):141–152, 2004. 348, 349, 390

[HW08]   P. E. Holmes and R. A. Wilson. On subgroups of the Monster containing A_5's. *J. Algebra*, 319(7):2653–2667, 2008. 45, 349

[Isa76]   I. M. Isaacs. *Character theory of finite groups*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1976. Pure and Applied Mathematics, No. 69. 57, 61, 262

[JLPW95] C. Jansen, K. Lux, R. Parker, and R. Wilson. *An atlas of Brauer characters*, volume 11 of *London Mathematical Society Monographs. New Series*. The Clarendon Press Oxford University Press, New York, 1995. Appendix 2 by T. Breuer and S. Norton, Oxford Science Publications. 29, 34, 38, 39, 40, 97, 352, 494

[KL90]   P. Kleidman and M. Liebeck. *The subgroup structure of the finite classical groups*, volume 129 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1990. 406, 407, 408, 409, 411, 412, 413, 473, 475, 477, 479, 485

[Kle87]   P. B. Kleidman. The maximal subgroups of the finite 8-dimensional orthogonal groups $P\Omega_8^+(q)$ and of their automorphism groups. *J. Algebra*, 110(1):173–242, 1987. 197, 201, 468, 470

[LP10]   K. Lux and H. Pahlings. *Representations of groups*, volume 124 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010. A computational approach. 57

[LW91]   S. A. Linton and R. A. Wilson. The maximal subgroups of the Fischer groups Fi_{24} and Fi'_{24}. *Proc. London Math. Soc. (3)*, 63(1):113–164, 1991. 332

[MNP19]   T. Merkwitz, L. Naughton, and G. Pfeiffer. TomLib, the gap library of tables of marks, Version 1.2.9. `https://gap-packages.github.io/tomlib`, Oct 2019. GAP package. 209, 213, 235, 240, 352

[MSW94]   G. Malle, J. Saxl, and T. Weigel. Generation of classical groups. *Geom. Dedicata*, 49(1):85–116, 1994. 413, 471, 492

[Nav98]   G. Navarro. *Characters and blocks of finite groups*, volume 250 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1998. 57

[Nor]   S. P. Norton. Improvements to the ATLAS–II. `http://brauer.maths.qmul.ac.uk/ Atlas/info/fullatlasmods.html`. 319

[NPP84]   J. Neubüser, H. Pahlings, and W. Plesken. CAS; design and use of a system for the handling of characters of finite groups. In M. D. Atkinson, editor, *Computational group theory (Durham, 1982)*, page 195–247, London, 1984. Academic Press. 15

[NR14]   G. Navarro and N. Rizo. Nilpotent and perfect groups with the same set of character degrees. *J. Algebra Appl.*, 13(8):1450061, 3, 2014. 29

[NW02]   S. P. Norton and R. A. Wilson. Anatomy of the Monster. II. *Proc. London Math. Soc. (3)*, 84(3):581–598, 2002. 350

[NW13]   S. P. Norton and R. A. Wilson. A correction to the 41-structure of the Monster, a construction of a new maximal subgroup $\mathrm{L}_2(41)$ and a new Moonshine phenomenon. *J. Lond. Math. Soc. (2)*, 87(3):943–962, 2013. 172, 185, 209, 230, 301, 350, 351

[Ost86]   T. Ostermann. Charaktertafeln von Sylownormalisatoren sporadischer einfacher Gruppen. Technical report, Universität Essen, Essen, 1986. 15

[Vdo00]   E. P. Vdovin. Large nilpotent subgroups of finite simple groups. *Algebra Log.*, 39(5):526–546, 630, 2000. 235

[Wil]   R. A. Wilson. ATLAS: Monster group M. `http://brauer.maths.qmul.ac.uk/ Atlas/spor/M`. 236, 301

[Wil93a] R. A. Wilson. More on maximal subgroups of the Baby Monster. *Arch. Math. (Basel)*, 61(6):497–507, 1993. 333, 334

[Wil93b] R. A. Wilson. Some new subgroups of the Baby Monster. *Bull. London Math. Soc.*, 25(1):23–28, 1993. 336

[Wil98]   R. A. Wilson. The McKay conjecture is true for the sporadic simple groups. *J. Algebra*, 207(1):294–305, 1998. 293

[Wil99]   R. A. Wilson. The maximal subgroups of the Baby Monster. I. *J. Algebra*, 211(1):1–14, 1999. 225, 333, 334, 335, 336, 390

[Wil10]   R. A. Wilson. New computations in the Monster. In J. Lepowsky, J. McKay, and M. P. Tuite, editors, *Moonshine: the first quarter century and beyond*, volume 372 of *London Math. Soc. Lecture Note Ser.*, page 393–403. Cambridge Univ. Press, Cambridge, 2010. 172, 209

[WPN⁺22] R. A. Wilson, R. A. Parker, S. Nickerson, J. N. Bray, and T. Breuer. AtlasRep, a GAP Interface to the Atlas of Group Representations, Version 2.1.6. `https://www.math.rwth-aachen.de/~Thomas.Breuer/atlasrep`, Oct 2022. GAP package. 229, 233, 313, 328, 329, 340, 352, 369

[WWT⁺]   R. A. Wilson, P. Walsh, J. Tripp, I. Suleiman, R. A. Parker, S. P. Norton, S. Nickerson, S. Linton, J. Bray, and R. Abbott. ATLAS of Finite Group Representations. `http://brauer.maths.qmul.ac.uk/Atlas/v3`. 38, 297, 313, 328, 329, 340, 346, 352, 369, 440

Index

CTblLibXpls, 1