MAXIMAL SUBGROUPS OF THE SPORADIC ALMOST-SIMPLE GROUPS

by

SIMON JON NICKERSON

A thesis submitted to
The University of Birmingham
for the degree of
MASTER OF PHILOSOPHY (SC, QUAL)

School of Mathematics and Statistics
The University of Birmingham
DECEMBER 2003

Abstract

Let S be one of the sporadic groups M_{12}, M_{22}, HS, McL, J_2, J_3, Suz, He, HN, O'N, Fi_{22}, Fi_{24}' and let $G = \mathrm{Aut}(S)$. Using a permutation representation of G on a computer, we calculate explicit generators for the maximal subgroups of G. The generators of the maximal subgroups are presented as straight line programs in standard generators of G, making it easy to produce each maximal subgroup for any choice of representation and generators for G. All these straight line programs have been made available on the World Wide Web. In the proof that the generators we produce are correct, we tabulate the orbit shapes of each maximal subgroup in the smallest faithful permutation representation of G, which are useful data when attempting to identify subgroups of G.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Professor R. A. Wilson for his invaluable advice, for introducing me to many interesting groups for the first time, and for allowing me to use his computer cluster for the computational part of this thesis.

I also wish to thank Dr J. N. Bray for sharing some of his knowledge of group computations and sporadic groups with me.

My deepest gratitude goes to my family and friends for all their patience and support.

Finally, I would like to thank EPSRC for financial assistance.

Contents

List of Tables

Notation and Conventions

We follow the Atlas [5] conventions for naming groups, conjugacy classes and characters. We note in particular the following:

D_{2n}	a dihedral group containing $2n$ elements
Fi_{24}	the automorphism group of the simple group Fi'_{24}
$A.B$	an extension (split or non-split) of A by B; that is, a group C with normal subgroup A such that $C/A \simeq B$
$A{:}B$	a split extension of A by B
$A \cdot B$	a non-split extension of A by B

Other notation is given below.

G	sporadic almost simple group
S	the simple group of index 2 in G
Ω	set on which G acts as a permutation group
c, d	standard generators for G
$\mathcal{G}$	isomorphism class of a group
$\mathcal{P}$	pair pre-characterisation or characterisation (see Definition 1.8 on page 11)

$G \dashrightarrow H$	short line program producing generators of H from standard generators of G (see 1.4)
$G \overset{\text{std}}{\dashrightarrow} H$	short line program producing standard generators of H from standard generators of G (see 1.4)
$G \overset{\text{std}}{\dashrightarrow}_Q H$	short line program producing generators of H (which map onto standard generators of Q) from standard generators of G (see 1.4)
g^h	$h^{-1}gh$; g conjugated by h
$\mathrm{Act}(\omega, g)$	ω^g ; the action of g on the point ω
W	the set of all prewords
$\eta(w)$	the value of a preword $w \in W$; see equation 2.2.1 on page 19
$\sigma(w)$	the successor of $w \in W$; see equation 2.2.2 on page 20
$\ell(w)$	the length of a preword $w \in W$
$\mathrm{ccl}(G)$	set of conjugacy classes of G
$\mathcal{C}$	a conjugacy class
$n\mathcal{C}$	the conjugacy class containing nth powers of elements from the conjugacy class $\mathcal{C}$; not to be confused with ...
$\mathcal{C}^n$	an elementary abelian group of order p^n in which every non-identity element is in the conjugacy class $\mathcal{C}$ (and every element of $\mathcal{C}$ has order p)
$H \lesssim K$	H is contained in a G-conjugate of K

Computer output and source code is always `monospaced`.

Chapter 1

Preliminaries

1.1 The sporadic almost-simple groups

Of the 26 sporadic simple groups, 12 of them have non-trivial outer automorphism groups. The full automorphism groups of these simple groups are:

$\mathrm{M}_{12}.2$	$\mathrm{M}_{22}.2$	HS.2	Suz.2
McL.2	$\mathrm{J}_2.2$	He.2	$\mathrm{Fi}_{22}.2$
O'N.2	HN.2	Fi_{24}	$\mathrm{J}_3.2$

We will refer to these 12 groups as *sporadic almost-simple groups*. Here we will introduce these groups (or their simple derived subgroups), giving sketches of constructions where practical.

1.1.1 Mathieu groups M_{12}, M_{22}

Definition 1.1 *A* Steiner system $S(t,k,v)$ *is a finite set* Ω *of* v *points together with a set* $\mathcal{B}$ *of subsets of* Ω *(called* blocks*) such that every block contains* k *points of* Ω*, and any set of* t *points of* Ω *is contained in a unique block. An* automorphism *of the Steiner system is a permutation of* Ω *which preserves the set* $\mathcal{B}$ *of blocks.*

The classical finite projective space of dimension d over the field $\mathbb{F}_q$ is a Steiner system $S(2, q+1, (q^{d+1}-1)/(q-1))$. The case $d = 2$, $q = 4$ (the projective plane over $\mathbb{F}_4$) gives a Steiner system $S(2,5,21)$, and in fact, it is the unique such Steiner system. It can be extended:

Theorem 1.2 (Witt) *There exist unique Steiner systems $S(3,6,22)$, $S(4,7,23)$ and $S(5,8,24)$.*

Proof. There are several ways to prove this. See, for example, chapter 6 of [6]. ■

The group M_{24} is the group of automorphisms of the Steiner system $S(5,8,24)$. The stabilizer of a 2-subset of Ω is the group $\mathrm{M}_{22}{:}2$; it is maximal in M_{24}. This gives rise to a permutation representation of $\mathrm{M}_{22}{:}2$ on 22 points (the embedding of $\mathrm{M}_{22}{:}2$ in M_{24} has orbit shape $2+22$).

The parameters (t,k,v) of a Steiner system determine the number of blocks which intersect a given block in a given subset of that block. For a Steiner system $S(5,8,24)$, there are 77 blocks which intersect a given block in a given 2-set. Accordingly, let B_1 be a block, and let B_2 be any block whose intersection with B_1 has size 2. Then the symmetric difference Σ of B_1 and B_2 is a set of size 12, called a *duodecad*, and $\{\Sigma, \Omega - \Sigma\}$ forms a $12+12$ partition of Ω called a *duum*. The stabilizer of a duodecad in M_{24} is isomorphic to M_{12}, and both of its permutation actions on 12 points are visible inside M_{24}. The stabilizer of a duum in M_{24} is the group $\mathrm{M}_{12}{:}2$, which is maximal in M_{24}. This subgroup is transitive but imprimitive on Ω, and in fact $\mathrm{M}_{12}{:}2$ does not act faithfully on fewer than 24 points.

1.1.2 Rank 3 permutation groups J_2, HS, McL and Suz

Definition 1.3 *The* rank r *of a group G acting transitively on a set Ω is the number of orbits of G in its action on $\Omega \times \Omega$ (or equivalently, the number of orbits of the point stabilizer on Ω).*

We have $r \geq 2$ because the set

$$\{(\omega, \omega) : \omega \in \Omega\} \tag{1.1.1}$$

is always an orbit, known as the *diagonal orbit.* If $r = 2$, the group is 2-transitive. The next case, $r = 3$, gives rise to a number of sporadic simple groups.

Let G be a rank 3 permutation group. We can choose one of the non-diagonal orbits O on $\Omega \times \Omega$ and use it to construct a graph on Ω. Conversely, G acts on this graph: given the graph, we can construct G and $\mathrm{Aut}(G)$. The groups HS.2, McL.2, J_2.2 and Suz.2 have rank 3 permutation actions, and their graphs are summarised in Table 1.1. The sporadic groups Fi_{22} and Fi_{24}' also have rank 3 permutation representations, but they will be discussed in section 1.1.3. The constructions for the graphs of HS and McL are not too complicated, and are given below.

The group HS.2 is a rank 3 permutation group on 100 points discovered by Higman and Sims [10]. Take a Steiner system $S(3, 6, 22)$ with point set Ω and block set $\mathcal{B}$. Label the vertices of a graph HS by Ω, $\mathcal{B}$ and an additional point $\star$ (giving $22 + 77 + 1 = 100$ vertices in total). In this graph, the vertex $\star$ is joined to every point in Ω, and each block $B \in \mathcal{B}$ is joined to each of the points it contains, and also to all the other $B' \in \mathcal{B}$ which are disjoint to B. The group of automorphisms of this graph is HS.2, and the stabilizer of the point $\star$ is $\mathrm{M}_{22}{:}2$. (The group is transitive on this graph, as for each $\alpha \in \Omega$, we can show that the sets:

$$S(\alpha) = \{\text{vertices of } \mathsf{HS} \text{ distance 1 from } \alpha\} \tag{1.1.2}$$

$$= \{\star\} \cup \{B \in \mathcal{B} : \alpha \in B\} \tag{1.1.3}$$

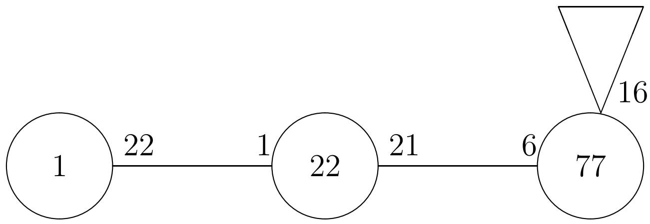
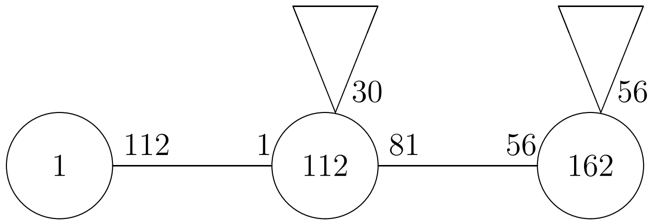
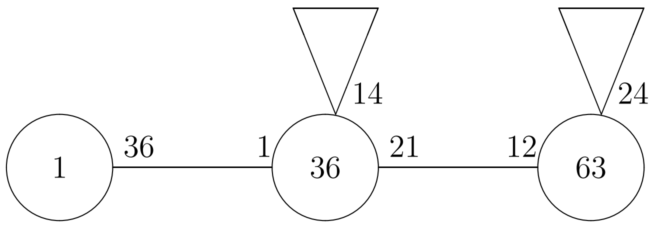
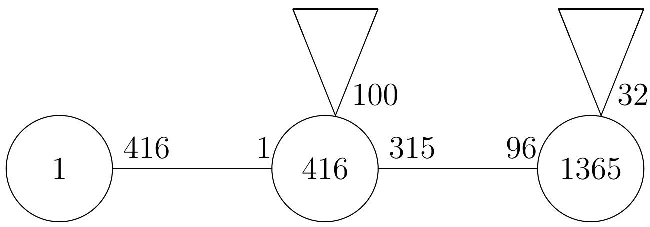
G	G_ω	$\lvert G : G_\omega \rvert$	Graph
HS.2	$M_{22}:2$	100	
McL.2	$U_4(3):2$	275	
J_2.2	$U_3(3):2$	100	
Suz.2	$G_2(4):2$	1782	

Table 1.1: Graphs for rank 3 permutation groups

and

$$B(\alpha) = \{\text{vertices of } \mathsf{HS} \text{ distance 2 from } \alpha\} \tag{1.1.4}$$

$$= (S - \{\alpha\}) \cup (\mathcal{B} - \{B \in \mathcal{B} : \alpha \in B\}) \tag{1.1.5}$$

define another $S(3,6,22)$ Steiner system, and that the graph from the point of view of α is the same as that from $\star$.)

The group McL.2 is a rank 3 permutation group on 275 points [13]. It can be constructed from the group $U_4(3)$, which has a subgroup $3^4.L_2(9)$ of index 112 and a subgroup $L_3(4)$ of index 162. The corresponding permutation representations make $U_4(3)$ a rank 3 permutation group in two different ways. The subdegrees are $1, 30, 81$ and $1, 56, 105$ respectively. Let $\mathcal{S}$ be the conjugacy class of $3^4.L_2(9)$ and $\mathcal{L}$ the conjugacy class of $L_3(4)$. The group $U_4(3)$ acts on $\mathcal{S} \times \mathcal{L}$, and there are two orbits. We select one of these orbits arbitrarily. We define a graph McL on the set $\{\star\} \cup \mathcal{S} \cup \mathcal{L}$ as follows:

- $\star$ is joined to every $S \in \mathcal{S}$;
- $S \in \mathcal{S}$ is joined to each element in the S-orbit of size 30 on $\mathcal{S}$;
- $L \in \mathcal{L}$ is joined to each element in the L-orbit of size 56 on $\mathcal{L}$;
- $S \in \mathcal{S}$ and $L \in \mathcal{L}$ are joined if (S, L) is in the chosen orbit of $\mathcal{S} \times \mathcal{L}$.

The automorphism group of the graph McL is McL:2.

1.1.3 Fischer groups Fi_{22}, Fi_{24}'

The Fischer groups arise as exceptional cases in a theorem of Fischer [7] classifying *3-transposition groups*, *i.e.* groups generated by a conjugacy class of involutions (called the *Fischer class*) such that the product of two non-commuting involutions has order 3.

Theorem 1.4 (Fischer) *Let G be a finite 3-transposition group whose derived subgroup is simple and whose centre is trivial. Then one of the following holds:*

- *G is a symmetric group*
- *G is a symplectic, orthogonal or unitary group over $\mathbb{F}_2$*
- *G is the subgroup of a (projective) orthogonal group over $\mathbb{F}_3$ generated by D, where D is a conjugacy class of reflections (in some representation of G).*
- *$G \simeq \mathrm{Fi}_{22}$, Fi_{23} or Fi_{24}.*

Proof. See chapter 5 of [1]. ■

The Fischer classes in Fi_{22} and Fi_{24} have sizes 3510 and 306936 respectively. The conjugation action of the groups on these classes give rise to the smallest faithful permutation representations. The centralizer of the product of two commuting elements of the Fischer class of Fi_{24} has shape $(2 \times 2 \cdot \mathrm{Fi}_{22}){:}2$, which allows us to find $\mathrm{Fi}_{22}.2$ from Fi_{24}.

There is a nice construction of Fi_{22} (which can easily be extended to a construction of $\mathrm{Fi}_{22}.2$) due to Conway [4] based on the fact that Fi_{22} has a subgroup $2^{10}{:}\mathrm{M}_{22}$. Names are given to the 3510 Fischer transpositions based on the orbit of M_{22} to which they belong, and the conjugation action of the group on this conjugacy class can be explicitly worked out, which gives rise to the permutation representation of the group Fi_{22} (or $\mathrm{Fi}_{22}.2$) on 3510 points.

1.1.4 Monster subgroups He and HN

The group He was discovered by Held [9] while attempting to characterise M_{24} as a simple group with involution centralizer $2_+^{1+6}.L_3(2)$. The only such groups are M_{24}, $L_5(2)$ and He.

The centralizer of an element t in the $7A$ conjugacy class of the Monster group is $\mathrm{He} \times \langle t \rangle$. In fact, He.2 can be embedded in Fi_{24}' (see section 1.1.3), because the normalizer in the Monster of $\langle t \rangle$ has shape $(7:3 \times \mathrm{He}).2$, and the centralizer in the Monster of the element of order 3 (in $7:3$) is $3.\mathrm{Fi}_{24}'$.

The group HN is involved in the Monster group as the centralizer of a $5A$-element modulo the cyclic group generated by that element (although it was constructed before the Monster was shown to exist). The group acts on a graph of valence 462 with 1140000 vertices, each labelled with a subgroup $A_{12} < \mathrm{HN}$. Two vertices are joined if and only if the intersection of the two subgroups is $A_6 \times A_6$. The automorphism group of this graph gives HN in its permutation representation on 1140000 points [15].

1.1.5 Pariahs J_3 and O'N

All but six of the sporadic simple groups are involved in the Monster group. These remaining six groups are called the *pariahs*, and two of them have outer automorphisms.

The group J_3 was discovered by Janko by classifying simple groups having involution centralizer $2_-^{1+4}:A_5$ (the only other such simple group is J_2). The group was then constructed by G. Higman and J. MacKay [11] by a coset enumeration on a subgroup $L_2(16):2$.

The group O'N was discovered by O'Nan by considering simple groups containing:

- a subgroup 2^3 with normalizer $4^3 \cdot L_3(2)$
- an involution with centralizer $4.L_3(4).2$

1.2 Maximal subgroups of almost-simple groups

We will use the following terminology from [5]:

Definition 1.5 *Let G be an almost-simple group with*

$$S \leq G \leq \mathrm{Aut}(S) = S.\mathrm{Out}(S) \tag{1.2.1}$$

Let H be a maximal subgroup of G. We say H is a triviality *if it is of the form $S.B$ where B is a subgroup of* $\mathrm{Out}(S)$*. We say H is* ordinary *if its intersection with S is maximal in S. Otherwise, H is a* novelty*.*

For the 12 sporadic almost-simple groups G, the only trivial maximal subgroup is the simple subgroup G', and generators for G' in terms of generators of G are already known. Thus in this thesis we will only be concerned with the ordinary and novel maximal subgroups.

The main result used in classifying maximal subgroups of simple and almost-simple groups is the following [19]:

Theorem 1.6 *Let G be a finite simple or almost-simple group with a maximal subgroup H which is not a triviality. Then H is the normalizer in G of a direct product of isomorphic simple groups.*

We will need a lemma:

Lemma 1.7 *Let K be a finite group with no characteristic subgroups (we say K is* characteristically simple*). Then K is a direct product of isomorphic simple groups.*

Proof. Let L be a minimal normal subgroup of K. We will show that there exists a sequence of automorphisms $\theta_1 = \mathrm{id}, \theta_2, \dots \theta_n$ of K such that K can be written as an internal direct product:

$$K = L \times \theta_2(L) \times \cdots \times \theta_n(L) \tag{1.2.2}$$

Under this assumption, we see that if $1 \neq N \trianglelefteq L$ then $N \trianglelefteq K$, but L is minimal normal in K, so $N = L$. Thus L is a simple group, and so to prove the theorem it suffices to show that such automorphisms of K exist.

We set $\theta_1 = \mathrm{id}$ and proceed inductively. Suppose $\theta_1, \dots \theta_m$ have already been constructed, and that $\theta_1(L) \times \cdots \times \theta_m(L)$ is a normal subgroup of K. If it is equal to K, then we are done. Otherwise, it is a proper normal subgroup of K, and so there exists an automorphism θ_{m+1} of K such that

$$\theta_{m+1}(L) \not\leq \theta_1(L) \times \cdots \times \theta_m(L) \tag{1.2.3}$$

for if not, $\theta_1(L) \times \cdots \times \theta_m(L)$ would be a proper characteristic subgroup of the characteristically simple group K. Now $\theta_{m+1}(L) \trianglelefteq K$, so $\theta_{m+1}(L) \cap (\theta_1(L) \times \cdots \times \theta_m(L))$ is normal in K (because it is the intersection of two normal subgroups of K) and contained in $\theta_{m+1}(L)$ (which is minimal normal in K). Thus

$$\theta_{m+1}(L) \cap (\theta_1(L) \times \cdots \times \theta_m(L)) = 1 \tag{1.2.4}$$

and:

$$(\theta_1(L) \times \cdots \times \theta_m(L))\theta_{m+1}(L) = \theta_1(L) \times \cdots \times \theta_m(L) \times \theta_{m+1}(L) \trianglelefteq K \tag{1.2.5}$$

Since K is only a finite group, this process must eventually terminate, giving the decomposition in equation (1.2.2). ■

Proof (Theorem 1.6). Let K be a minimal normal subgroup of H. Because H is not a triviality, H (and therefore K) does not contain S, and since any non-identity normal subgroup of G must contain S, K cannot be normal in G. Thus $H \leq N_G(K) < G$, and since H is a maximal subgroup of G, we deduce that $H = N_G(K)$.

Minimal normal subgroups are characteristically simple, so by Lemma 1.7, K is the direct product of isomorphic simple groups. Hence H is the normalizer of a direct product of isomorphic simple groups. ■

The classification of (conjugacy classes of) maximal subgroups of G can therefore be achieved by the following procedure:

1. Find all conjugacy classes of direct products of simple groups inside G (we might have to rely on the Classification of Finite Simple Groups here).

2. Calculate their normalizers in G.

3. Determine whether there are any inclusions (up to conjugacy) among the groups found.

Subgroups of the form $N_G(p^n)$ (for p prime) are called *p-local subgroups*. The analysis of the p-local subgroups is generally easier than that of the non-local subgroups. The procedure has been completed for all the sporadic groups and their automorphism groups, with the exception of the Monster group, for which the non-local analysis is not yet complete. However, there are complete lists of the maximal subgroups for the 12 sporadic almost-simple groups. We will assume throughout that the lists of maximal subgroups given in [5] (with corrections in an appendix to [12] and in [20]) are correct and complete where claimed.

If G is a finite group, we let $K_{G;i}$ denote an arbitrary representative of the ith conjugacy class of maximal subgroups of G, labelled in descending order of size. Where there is no confusion, we will write K_i for $K_{G;i}$. We are concerned with the problem of finding maximal subgroups of G explicitly: given generators of G, find generators of H_i (a group G-conjugate to K_i). Such generators have been found for a number of other groups, including many of the sporadic simple groups [20] and the almost-simple group $J_3.2$ [17]. In this thesis, we will find generators for all 12 of the sporadic almost-simple groups (including $J_3.2$ for completeness).

In section 1.3 we will describe the particular generators of G which we will use. In section 1.4 we will describe the way that generators of H_i can be specified in terms of generators of G, and in section 1.5 we make some non-rigorous remarks on which generators of H_i we prefer. Finally, in section 1.6 we describe the computer system that we used for finding generators and testing them.

1.3 Characterised and standard generators

In this section, we will define certain special generators of G (called *standard generators*) that we will always use as our starting point for finding generators of the maximal subgroups of H. Standard generators were defined by Robert Wilson in [21].

The following three terms are not standard, but will be useful in the sequel.

Definition 1.8 *Let $\mathcal{G}$ be an isomorphism class of groups. A* pair pre-characterisation $\mathcal{P}$ *for $\mathcal{G}$ is a set of predicates $\{P_1(x, y), \ldots P_n(x, y)\}$ (in variables x and y) in the theory of $\mathcal{G}$. If $G \in \mathcal{G}$ and c and d are elements of G satisfying the predicates $P_1, \ldots P_n$, then (c, d, G) is said to be a $\mathcal{P}$-*triple*, or alternatively, c and d are $\mathcal{P}$-elements of G.*

Definition 1.9 *Let $\mathcal{P}$ be a pair pre-characterisation. We say $\mathcal{P}$ is a* pair characterisation *if whenever (c, d, G) and $(\hat{c}, \hat{d}, \hat{G})$ are $\mathcal{P}$-triples there is an isomorphism $\theta : \langle c, d\rangle \to \langle \hat{c}, \hat{d}\rangle$ induced by $c \mapsto \hat{c}$, $d \mapsto \hat{d}$.*

Definition 1.10 *A pair characterisation $\mathcal{P}$ is a* generating pair characterisation *if there exists a $\mathcal{P}$-triple (c, d, G) such that G is generated by c and d. In this case, c and d are called $\mathcal{P}$-generators of G.*

Note that if $\mathcal{P}$ is a pair characterisation and G is generated by c and d for one $\mathcal{P}$-triple (c, d, G), the same is true for all $\mathcal{P}$-triples.

Example 1.11 *Let $\mathcal{G}$ be the isomorphism class of the group A_5. We define some predi-*

cates in the theory of $\mathcal{G}$:

$$P_1(x,y) := \text{“}x \text{ has order } 2\text{”} \tag{1.3.1}$$

$$P_2(x,y) := \text{“}y \text{ has order } 2\text{”} \tag{1.3.2}$$

$$P_3(x,y) := \text{“}y \text{ has order } 3\text{”} \tag{1.3.3}$$

$$P_4(x,y) := \text{“}xy \text{ has order } 5\text{”} \tag{1.3.4}$$

Then:

- $\mathcal{P}_1 := \{P_1, P_3\}$ *is a pair pre-characterisation but not a pair characterisation (xy could have order* 3, 4 *or* 5*, so we cannot induce an isomorphism* θ*);*
- $\mathcal{P}_2 := \{P_1, P_2, P_4\}$ *is a pair characterisation, as the* $(2,2,5)$ *class multiplication coefficient in* $\mathrm{Aut}(A_5) = S_5$ *is* 1 *and the centralizer of a* $(2,2,5)$*-subgroup of* S_5 *has order* 1*. However, all* $(2,2,5)$ *subgroups are isomorphic to* D_{10}*, so* $\mathcal{P}_2$ *is not a generating pair characterisation.*
- $\mathcal{P}_3 := \{P_1, P_3, P_4\}$ *is another pair characterisation, because the* $(2,3,5)$ *class multiplication coefficient in* $\mathrm{Aut}(A_5) = S_5$ *is* 1 *and the centralizer of a* $(2,3,5)$*-subgroup of* S_5 *has order* 1*. Moreover,* $\mathcal{P}_3$ *is a generating pair characterisation, because we can* $(2,3,5)$*-generate* A_5 *with* $(12)(45)$ *and* (135)*.*

In general, there may be many choices for generating pair characterisations $\mathcal{P}$ for an isomorphism class of groups $\mathcal{G}$ (or indeed there may be none if the group cannot be 2-generated). The concept is most useful when it is easy to find $\mathcal{P}$-generators for any group $G \in \mathcal{G}$. Good choices of $\mathcal{P}$ for many groups — including all the sporadic simple groups and their automorphism groups — are given in [20]. These choices for $\mathcal{P}$ are the *standard generating pair characterisations*. If $\mathcal{P}$ is the standard generating pair characterisation for $\mathcal{G}$, and c and d are $\mathcal{P}$-generators of G, we say that c and d are *standard generators of G*.

Standard generators are useful because the induced isomorphism θ is explicit. If we describe a subgroup of G in terms of words in standard generators of G, then the subgroup is defined up to automorphisms of G. In particular, if we are interested in finding a certain subgroup $\hat{H}$ of $\hat{G}$ in a large representation, we can use the following procedure:

1. Find the equivalent subgroup H in a smaller representation G, expressing its generators in standard generators c, d of G.

2. Find standard generators $\hat{c}$, $\hat{d}$ of G.

3. Copy across the words found in step 1, substituting $\hat{c}$ for c and $\hat{d}$ for d. These generate the subgroup $\hat{H}$.

Let G be any sporadic almost-simple group, and let H be a maximal subgroup of G. We will find generators for a G-conjugate of H in terms of standard generators of G (step 1 of the above procedure). In this way, we will have found the maximal subgroup H 'once and for all', as it is then easy to find it in any other representation of G.

1.4 Format of generators

For consistency with [20], we express our generators of H in terms of *straight line programs*. These are lists of instructions which produce group elements from other group elements, and they reflect how the generators would be calculated in a black box group. Calculated group elements are labelled by natural numbers, which can be thought of as labelled sections of computer memory. Initially, the standard generators of G are elements 1 and 2. There are three types of instruction:

- `mu` a b c

  Multiply group elements a and b and store the answer in c.

- `iv` a b

  Invert group element a and store the answer in b.

- `pwr` n a b

  Raise group element a to the power n and store in b.

Numbers can be re-used in the process of a calculation, providing that if element a is used as input for a given operation, the output cannot be put in a. At the end of the straight line program, there may be a meta-instruction of the form "`oup` n $a_1 \ldots a_n$", which indicates that the group elements $a_1, \ldots a_n$ are the required generators. This line may be omitted if the required generators are elements 1 and 2.

Note that in theory it is possible to write down the generators of H as explicit words in the standard generators of G from a straight line program. However, such words may be extremely long (the length of words required can potentially increase exponentially with the length of the short line program).

Example 1.12 *Table 1.2 gives a straight line program which produces generators for the group* $\langle c, [(cdcd^2)^7]^{cd} \rangle$*. Observe that element 1 is* c *at the end of the program because no instruction changes it.*

Program instruction	Meaning	Element calculated
`mu 1 2 3`	$t_3 := t_1 t_2$	cd
`mu 3 2 4`	$t_4 := t_3 t_2$	cd^2
`mu 3 4 5`	$t_5 := t_3 t_4$	$cdcd^2$
`pwr 7 5 4`	$t_4 := t_5^7$	$(cdcd^2)^7$
`iv 3 2`	$t_2 := t_3^{-1}$	$(cd)^{-1}$
`mu 2 4 5`	$t_5 := t_2 t_4$	$(cd)^{-1}(cdcd^2)^7$
`mu 5 3 2`	$t_2 := t_5 t_3$	$(cd)^{-1}(cdcd^2)^7(cd)$

Table 1.2: A straight line program with its meaning

We will denote a straight line program which produces generators for H from standard generators of G by $G \dashrightarrow H$. If the generators of H are standard too, we may denote the program by $G \overset{\text{std}}{\dashrightarrow} H$. If N is a normal subgroup of H and the images of the generators of H under the natural map $x \mapsto xN$ are standard generators of $H/N \simeq Q$, we denote the program by $G \overset{\text{std}}{\dashrightarrow}_Q H$.

The straight line programs themselves can be found online at [14], with filenames following the conventions in [20]. Chapter 3 gives GAP [8] code which produces the same generators as the straight line programs.

1.5 Preferred generators

Let G be a sporadic almost-simple group, H a maximal subgroup of G. In finding generators for H we make the following preferences:

- We prefer to generate H by as few elements as possible. In most cases, it is possible to generate H with two elements.
- Short programs are preferred to long ones, but we will not attempt to find minimal length programs.
- If feasible, and if it does not conflict too much with the preference for short programs, we would prefer the generators of H to be standard generators (if they are defined). If H has no standard generators defined, but some homomorphic image $\bar{H}$ of H does, then we would like the generators of H to map onto standard generators of $\bar{H}$. If a shorter program is possible without this extra constraint, then we would like both programs.

We would like the generators of H to be standard because it aids the 'inductive' finding of subgroups. For instance, we can find a subgroup A_7 in HS.2 by using the chain HS.2 $>$ M_{22}.2 $>$ A_7. If we have programs $p_1 = \text{HS.2} \overset{\text{std}}{\dashrightarrow} M_{22}.2$ and $p_2 = M_{22}.2 \dashrightarrow A_7$ then we can produce a program HS.2 $\dashrightarrow A_7$ by concatenating p_1 and p_2 (possibly with some renumbering of group elements).

1.6 Computer environment

The majority of the computational work was done on a single node of the Linux Beowulf cluster funded by EPSRC grant GR/R95265/01. Each node had two AMD Athlon MP 1900+ CPUs with 2GB memory. The discrete algebra system GAP [8] was used for group theoretic calculations.

Chapter 2

Finding generators

Let G be a sporadic almost-simple group with standard generators c and d. In this chapter, we describe the methods used to find straight line programs for generators of the maximal subgroups of a sporadic almost-simple G starting with the elements c and d. The straight line programs that we found are given in [14] and equivalent GAP code is given in chapter 3. The programs will be shown to be correct in chapter 4.

2.1 Representations of G

The existence of standard generators for G means that we are free to choose whichever representation of G is the most convenient. If we treat G purely as a black box group, then it makes sense to choose a representation on the basis of how fast we can multiply, invert and compare elements of G, and perhaps how much space is needed to store an element of G. This would suggest that we look at faithful permutation representations on small numbers of points and faithful (sparse!) matrix representations over small finite fields, and then choose whichever representation allows the fastest multiplications, inversions and comparisons. However, matrix and permutation groups can provide more information than just black box computations. In matrix groups, we can look at the action of a matrix on a vector to investigate the order of an element, or find the trace of the matrix to investigate conjugacy. In permutation groups, the action of a permutation on a point and the cycle structure (respectively) provide the same sort of information. However, the most efficient and time-honoured algorithms for computing with groups are designed for permutation groups (see [16] for an exposition of such algorithms, or [8] for implementations of some of them), so we would prefer to work with permutation representations if they are available and are of moderate size. For the groups we are investigating, permutation representations on a reasonable number of points have been calculated (see Table 2.1 for the sizes of the smallest set Ω on which G can act), so we will use these representations most of the time.

G	H	$\lvert G:H\rvert = \lvert\Omega\rvert$
$\mathrm{M}_{22}.2$	$L_3(4){:}2$	22
$\mathrm{M}_{12}.2$	M_{11}	24
HS.2	$\mathrm{M}_{22}{:}2$	100
$\mathrm{J}_2.2$	$U_3(3){:}2$	100
McL.2	$U_4(3){:}2$	275
Suz.2	$G_2(4){:}2$	1782
He.2	$S_4(4){:}4$	2058
$\mathrm{Fi}_{22}.2$	$2 \cdot U_6(2).2$	3510
$\mathrm{J}_3.2$	$L_2(16){:}4$	6156
O'N.2	$L_3(7){:}2$	245520
Fi_{24}	$\mathrm{Fi}_{23} \times 2$	306936
HN.2	S_{12}	1140000

Table 2.1: Smallest faithful permutation representations

Frequently when working within a subgroup of $H < G$, we may focus on the action of H on one of its orbits, giving rise to a smaller and faster permutation representation of H. This is particularly useful when Ω is a large set.

There are two circumstances where we may need to use a different representation entirely: to distinguish conjugacy classes in cases of ambiguity (see section 2.2.2), and to prove that one subgroup of G is not contained in another one (see for example section 4.2.7). All the representations used can be found in [20].

2.2 A computational toolkit

In this section, we will discuss some computational tools and techniques that will be needed when searching for and constructing maximal subgroups of G. Some of the material in this section relies upon treating G as a group of permutations of Ω. Implementations in GAP are given in the appendix.

2.2.1 Listing elements of G

Recall that c and d are standard generators of G.

Definition 2.1 *Let $n \geq 1$ be an arbitrary integer. A* preword of length n *is an n-tuple of integers $(r_1, \ldots r_n)$ where $1 \leq r_i < o(d)$ for $1 \leq i \leq n$. The length of a preword w is denoted $\ell(w)$. We will denote the set of all prewords by W. To each preword, we associate an element of G via the* evaluation map*:*

$$\eta : (r_1, \ldots r_n) \mapsto cd^{r_1} \cdots cd^{r_n} \tag{2.2.1}$$

If a and b are prewords, we say a and b are equivalent *if $\eta(a) = \eta(b)$.*

When computing $\eta(w)$ for many prewords w, it saves time to have the 'atoms' cd^i precomputed.

Program 2.2 *The function* `TKevalpreword` *in section A.2.1 calculates $\eta(w)$ for a preword w.*

For all the groups G we are considering, the generator c is an involution, so when one writes down a word in c and d and makes any obvious cancellations, the word looks like $\eta(w)$ for some preword w, except that there may be no c at the beginning, and there may be a c at the end. The next lemma shows that these provisoes are harmless.

Lemma 2.3 *Any element $g \in G$ can be written as $\eta(w)$ for some preword w.*

Proof. It suffices to show that c and d can be written as images of prewords. Clearly $d = (cd)^{-1}cd^2$, which is just $(cd)^{o(cd)-1}cd^2$, so d can be written as the image of a preword. Moreover, $c = (cd)d^{-1} = (cd)d^{o(d)-1}$, so c can also be written as the image of a preword. ■

Definition 2.4 *Let $w = (r_1, \ldots r_n)$ be a preword. If $r_1 = \cdots = r_n = o(d)-1$, then define:*

$$\sigma(w) = (\underbrace{1, \ldots 1}_{(n+1)}) \tag{2.2.2}$$

Otherwise, let i be minimal such that $r_i < o(d) - 1$. Then define:

$$\sigma(w) = (\underbrace{1, \ldots 1}_{(i-1)}, r_i + 1, r_{i+1}, \ldots r_n) \tag{2.2.3}$$

Then σ defines a function $W \to W$ known as the successor function.

The function σ lists the prewords in order of length, reverse-lexicographically on prewords of the same length. Thus we can list the elements of G by using σ to list prewords and by evaluating them with η. Since this procedure involves listing all prewords and there are only finitely many elements in G, elements will be listed more than once.

Program 2.5 *The* GAP *function* `TKlistprewords` *defined in section A.2.2 can be used to list all prewords up to a certain length. Used with with* `TKevalpreword`*, it can list elements of G.*

In a more sophisticated element-listing procedure, it may be worth taking into account certain equivalences of prewords (especially if cd^i has a fairly small order), but the listing procedure in its present form was found to be effective enough. In practice, we specify an upper limit to the number of prewords to evaluate, so that the procedure terminates.

2.2.2 Identifying conjugacy classes

It is frequently important to be able to identify (or at least, narrow down the range of possibilities for) the conjugacy class of an element $x \in G$. Conjugate elements of G have the same cycle shape (because they must be conjugate in the corresponding symmetric group), and in many cases, looking at the cycle shape of x is enough to identify its conjugacy class.

Definition 2.6 *If $\mathcal{C}$ is a conjugacy class of G, let $n\mathcal{C}$ denote the conjugacy class characterised by the property*

$$x \in \mathcal{C} \implies x^n \in n\mathcal{C} \tag{2.2.4}$$

The map

$$\mathrm{ccl}(G) \to \mathrm{ccl}(G); \qquad \mathcal{C} \mapsto n\mathcal{C} \tag{2.2.5}$$

is called the nth power map *of G.*

Lemma 2.7 *Let χ be the permutation character of G, $\mathcal{C}$ a conjugacy class of G, and z_d the number of d-cycles in the disjoint cycle representation of an element of $\mathcal{C}$. Then:*

$$\sum_{d|n} dz_d = \chi(n\mathcal{C}) \tag{2.2.6}$$

Proof. Let x be an arbitrary member of $\mathcal{C}$. The term dz_d is the number of points contained in d-cycles in x. When x is raised to the power n, all these d-cycles disappear, so the left hand side is the total number of points which are fixed when x is raised to the nth power. This is just the the permutation character applied to x^n, *i.e.* $\chi(n\mathcal{C})$. ∎

If the permutation representation of G is that of cosets of a subgroup H, then χ is just the trivial character of H induced to G. The irreducible constituents of χ are given in [5] for all the groups we are investigating, with the exception of HN.2. The permutation character of HN on 1140000 points is:

$$\chi = 1a + 133\text{ab} + 760\text{a} + 3344\text{a} + 8910\text{a} + 16929\text{a} + \\ 35112\text{ab} + 267520\text{a} + 365750\text{a} + 406296\text{a} \tag{2.2.7}$$

and it extends to HN.2 in the obvious manner. The power maps of G are all known and given in [5]. Thus the numbers $\chi(n\mathcal{C})$ can all be calculated, and so by Lemma 2.7, we can recursively calculate the numbers z_d and hence the cycle shape of elements of $\mathcal{C}$. Hence we can find the cycle shape associated with each ATLAS class. In many cases, this allows us to identify the ATLAS class of elements of G.

Program 2.8 *The* GAP *function* `TKccl` *defined in section A.3.2 attempts to find the* ATLAS *class of an element by this method.*

Occasionally, looking at cycle shapes is not enough. If necessary, computations can be transferred (via the procedure mentioned in section 1.3) either to another permutation representation, or to a matrix representation (where the trace provides a good invariant for conjugacy classes).

2.2.3 Finding conjugacy class representatives

Closely related to the task of identifying conjugacy classes of elements is that of finding a representative of a particular conjugacy class. If there is a fast way of identifying the conjugacy class of an element of G, then finding conjugacy class representatives can be done by searching through elements of G using the procedure in section 2.2.1 until we find one in the correct class.

We can often improve significantly on this naive procedure by *powering up*. If $\mathcal{C}$ contains elements of low order, it is probably a very small conjugacy class, and we might have to search through a large number of elements of G before finding an element of $\mathcal{C}$.

However, it is fairly likely that there exists a larger conjugacy class $\mathcal{C}'$ which powers up to $\mathcal{C}$; that is; there exists an integer n such that $n\mathcal{C}' = \mathcal{C}$.

This can also help to find elements in classes where the cycle shape does not give enough information to identify the class. For example, with HN.2 in its permutation representation on 1140000 points, the classes $5A$ and $5D$ both have cycle shape

$$5A, 5D : \qquad 1^{50}\, 5^{227990} \tag{2.2.8}$$

and so it is not possible to tell them apart by cycle shape alone. However, the classes $20F$ and $20G$ power up to $5A$ and $5D$ respectively, and they have different cycle shapes:

$$20F : \qquad 1^{10}\, 4^{10}\, 5^{218}\, 10^{770}\, 20^{56558} \tag{2.2.9}$$

$$20G : \qquad 2^{5}\, 4^{10}\, 5^{220}\, 10^{769}\, 20^{56558} \tag{2.2.10}$$

Program 2.9 *The* GAP *function* `TKfindcclrep` *defined in section A.3.3 can be used to find representatives of a given* ATLAS *class in G.*

2.2.4 Tools for large degree permutation groups

During the search for (or construction of) a subgroup $H < G$, we often want to find a preword w satisfying a certain property. Such properties might include:

(P1) a particular word in g and $h^{\eta(w)}$ (say $f := gh^{\eta(w)}$) has order n;

(P2) $\eta(w)$ commutes with g;

(P3) $h^{\eta(w)}$ commutes with g;

(P4) $\eta(w)$ normalizes the subgroup K;

(P5) $h^{\eta(w)}$ normalizes the subgroup K.

Most prewords w will not have the required property, but if Ω is very large, then checking the property naively for each w is likely to be expensive. Checking property (P1) for a preword w in the obvious way involves:

- calculating $u = \eta(w)$ ($\ell(w) - 1$ multiplications)
- calculating $f = gh^u$ (3 multiplications and 1 inversion)
- finding the order of f

For $G =$ HN.2 as a permutation group on 1140000 points, multiplying two elements takes about 170ms, so checking (P1) for even a single preword could take a couple of seconds. Since we want to be able to test a very large number of group elements (sometimes millions), this is unacceptably slow.

Fortunately, it is frequently not necessary to perform so many multiplications. For each of the properties (P1)-(P5), we will describe a test which quickly eliminates most prewords w which fail to have the property. If a preword w passes the test, this does not necessarily show that w has the property, so we may still have to check the property explicitly for some prewords, but there will be fewer prewords to check than before.

Before testing any prewords, we require a set $\Omega_0 \subset \Omega$ of points to be chosen. The choice of points may be random, but if a short base for G on Ω is known, then it would make a good choice. If we have a subgroup K that we wish to normalize, then the test is more effective if the K-orbits of the points of Ω_0 are small. Generally, increasing $|\Omega_0|$ will make the test more reliable, although for the sake of speed, Ω_0 should be significantly smaller than Ω. We will usually take $|\Omega_0| \leq 20$.

So that the test is quick, we precalculate cd^i and $(cd^i)^{-1}$ for $1 \leq i < o(d)$. For properties (P4) and (P5), we also calculate the K-orbit of each ω. For property (P5) we will additionally need to precompute h^{-1}.

The details of each test for a preword $w = (r_1, \ldots, r_n)$ are given below. In order to make the notation cleaner, we write $\mathrm{Act}(\omega, g)$ in place of ω^g.

(P1) Testing that a particular word in g and $h^{\eta(w)}$ (say $f := gh^{\eta(w)}$) has order n.

For each $\omega \in \Omega_0$, we set $\omega_0 = \omega$ and calculate recursively:

$$\begin{aligned} \omega_{i+1} &= \mathrm{Act}(\omega_i, gh^{\eta(w)}) \\ &\equiv \mathrm{Act}(\omega_i, g\eta(w)^{-1}h\eta(w)) \end{aligned} \tag{2.2.11}$$

until $\omega_r = \omega_0$ for $r = r(\omega) > 0$. We then calculate the least common multiple k of all the $r(\omega)$. We know that the order of f is a multiple of k.

- If k does not divide n, then w fails.
- If $k = n$, then w passes.
- If k is a proper factor of n, then w fails, although it is possible that f has the correct order. This false negative should not occur very often if the points of Ω_0 were chosen well. Accepting w as a candidate would mean that too many false prewords passed the test, making it less useful.

Program 2.10 *The* GAP *function* `TKorderword` *in section A.4.2 is an implementation of this test for arbitrary words in g and $h^{\eta(w)}$.*

(P2) Testing that $\eta(w)$ commutes with g.

We check that

$$\mathrm{Act}(\omega, \eta(w)g) = \mathrm{Act}(\omega, g\eta(w)) \tag{2.2.12}$$

for all $\omega \in \Omega_0$. We usually want to ignore the case $\eta(w) = 1$, so we insist that $\mathrm{Act}(\omega, \eta(w)) \neq \omega$ for some ω.

Program 2.11 *The* GAP *function* `TKcentralizertest` *in section A.4.3 is an implementation of this test.*

(P3) Testing that $h^{\eta(w)}$ commutes with g.

We check that

$$\mathrm{Act}(\omega, \eta(w)^{-1}h\eta(w)g) = \mathrm{Act}(\omega, g\eta(w)^{-1}h\eta(w)) \tag{2.2.13}$$

for all $\omega \in \Omega_0$.

Program 2.12 *The* GAP *function* `TKcentralizerccltest` *defined in section A.4.4 is an implementation of this test.*

(P4) Testing that $\eta(w)$ normalizes a subgroup K.

Let $\{k_1, \dots k_s\}$ be a set of generators for K. It is worth spending some time to make this set as small as possible. We check for each $1 \leq i \leq s$ that

$$\mathrm{Act}(\omega, k_i^{\eta(w)}) \equiv \mathrm{Act}(\omega, \eta(w)^{-1}k_i\eta(w)) \in \omega^K \tag{2.2.14}$$

for each $\omega \in \Omega_0$ with K-orbit ω^K.

Program 2.13 *The* GAP *function* `TKnormalizertest` *in section A.4.5 is an implementation of this test.*

(P5) Testing that $h^{\eta(w)}$ normalizes a subgroup K.

Let $k_1, \dots k_s$ be a small set of generators for K. We check for each $1 \leq i \leq s$ that

$$\mathrm{Act}(\omega, k_i^{(h^{\eta(w)})}) \equiv \mathrm{Act}(\omega, \eta(w)^{-1}h^{-1}\eta(w)k_i\eta(w)^{-1}h\eta(w)) \in \omega^K \tag{2.2.15}$$

for each $\omega \in \Omega_0$ with K-orbit ω^K.

Program 2.14 *The* GAP *function* `TKnormalizerccltest` *in section A.4.6 is an implementation of this test.*

Note that none of the tests requires any multiplication or inversion. In particular, we never need to explicitly calculate $\eta(w)$. An expression of the form $\mathrm{Act}(\omega, x_1 x_2 \dots x_t)$ can be calculated by successive point actions as $\mathrm{Act}(\cdots \mathrm{Act}(\mathrm{Act}(\omega, x_1), x_2), \dots x_n)$. Any time $\eta(w)$ appears in a formula, it should be replaced by its expansion $(cd^{r_1}). \cdots (cd^{r_n})$ of precalculated group entries; similarly, $\eta(w)^{-1}$ should be replaced by $(cd^{r_n})^{-1} \cdots (cd^{r_1})^{-1}$.

2.2.5 Involution centralizers

The following lemmas provide elements which commute with involutions; accordingly, they can be used to find generators of involution centralizers (often maximal subgroups). The first is due to John Bray, and the second is attributed to Richard Parker; both are described in [2].

Lemma 2.15 *Let t, g be elements of G such that t has order 2. Let n be the order of the element $t.t^g$. Define:*

$$z = \begin{cases} (t.t^g)^{n/2} & \text{if } n \text{ is even} \\ g(t.t^g)^{(n-1)/2} & \text{if } n \text{ is odd} \end{cases} \tag{2.2.16}$$

Then z commutes with t.

Proof. If n is even, write $n = 2m$. Then $\langle t, t^g \rangle$ is isomorphic to D_{4m} so it has a central element of order 2 (corresponding to a rotation by π of a regular $2m$-gon) given by $z = (t.t^g)^m$.

If n is odd, write $n = 2m + 1$. Then:

$$\begin{aligned}
[t, z] &= t(t^g.t)^m g^{-1} t g (t.t^g)^m \\
&= t(t^g t)^m t (t.t^g)^{m+1} \\
&= (t.t^g)^m t^2 (t.t^g)^{m+1} \\
&= (t.t^g)^n \\
&= 1
\end{aligned}$$

■

Lemma 2.16 *If t, u are non-conjugate involutions of G and n is the order of tu, then n is even and $(tu)^{n/2}$ commutes with t and u.*

Proof. The group $\langle t, u \rangle$ is isomorphic to D_{2n}, but the groups D_{2m+2} have a unique conjugacy class of involutions. Thus n is even, and $(tu)^{n/2}$ is the central element of $\langle t, u \rangle$. ■

In practice, these lemmas are very effective, and it usually only takes a few iterations before the whole centralizer has been generated.

These lemmas are also useful in constructing a maximal subgroup stepwise (see section 2.5).

2.2.6 Recognising subgroups

Some of the techniques mentioned in this chapter (for example trawling; section 2.4.1) involve producing a large number of sets T of elements of G, and checking whether $\langle T \rangle$ is an interesting group (perhaps a maximal subgroup of G). We will call these *candidate sets*. We need a way of recognising when $H = \langle T \rangle$ is the group we are looking for. Our main tools for doing this are the orbit shape and group order.

The orbit shape is useful because it is relatively quick to calculate, and it is invariant under conjugation. Moreover, we have the following fact:

Theorem 2.17 *Let G be one of the 12 sporadic almost simple groups in its smallest faithful permutation representation, and let H be a transitive subgroup of G not entirely contained in the simple group G'. Then either $H = G$ or H is contained in a maximal subgroup K for one of the pairs (G, K) listed in Table 2.2.*

G	K				
$M_{12}.2$	$L_2(11){:}2$	$(2^2 \times A_5){:}2$	$4^2{:}D_{12}.2$	$S_4 \times S_3$	S_5
$M_{22}.2$	$L_2(11){:}2$				
HS.2	$M_{22}{:}2$	$5^{1+2}.[2^5]$	$5{:}4 \times S_5$		
$J_2.2$	$(A_5 \times D_{10}).2$	$5^2{:}(4 \times S_3)$			
Suz.2	$U_5(2){:}2$	$3^5.(M_{11} \times 2)$			
He.2	$7^2{:}2L_2(7).2$	$7_+^{1+2}{:}(S_3 \times 6)$			
$Fi_{22}.2$	${}^2F_4(2)$				

Table 2.2: Transitive maximal subgroups $K < G$, $K \nleq G'$

Proof. This will follow from the tables given in chapter 3. ■

In particular, we have:

Corollary 2.18 *Let $G \in \{\mathrm{McL}.2, \mathrm{J}_3.2, \mathrm{O'N}.2, \mathrm{HN}.2, \mathrm{Fi}_{24}\}$ in its smallest faithful permutation representation. If H is a transitive subgroup of G not entirely contained in G', then $H = G$.*

Thus if the candidate set T contains any outer elements of G, then testing whether $\langle T \rangle$ is transitive can be a fairly effective test of whether it is a proper subgroup of G. Much of the time, this is the most important case to rule out. Note that the three groups with the largest permutation representations (and which are therefore the most difficult groups in which to perform calculations) are covered by Corollary 2.18.

Unfortunately, the orbit shape does not necessarily characterise the subgroup, and more checks may need to be carried out to ensure that the group is the one sought. Of these tests, checking the group order is the most simple. In the smaller groups especially, it can be very effective, and can help in the cases where the orbit shape fails to distinguish non-conjugate groups. It is important to note that we check the orbit shape first, as calculating the orbits of a subgroup is substantially quicker than calculating its order.

2.3 Finding generators by sifting — a rejected approach

There is a standard algorithm (called *sifting*) for testing whether a permutation π is contained in a permutation group G ([16] section 4.1). A slight modification to this algorithm gives a procedure to express an element of G as a word in the generators of G. Thus we can find words for generators of a (maximal) subgroup H of G by finding H inside G and sifting the generators of H. In GAP this can be done by constructing a homomorphism from the free group $F_2 = \langle x, y \rangle$ into G (mapping x and y to the standard generators of G) and then choosing a preimage of each generator of H.

Unfortunately, this method gives extremely long words. For instance, the words for standard generators of a maximal subgroup M_{11} of M_{12} in terms of its standard generators were of length 442 and 605 (the exact numbers vary depending on the choices of preimages in F_2). With the maximal subgroup $U_4(3)$ of McL (a permutation group on 275 points), words were around 10^4 in length. These give rise to extremely long straight line programs compared to those given in [20], and the problem gets worse as the degree of G and the index of H in G increase. For this reason, the method was rejected as unworkable.

2.4 Maximal subgroups by random searching

The methods in this section are not new, and have been used by John Bray and others [3] to find generators for the maximal subgroups for the sporadic simple groups of order less than 10^{14}.

2.4.1 Trawling

The following method, which we call 'trawling', attempts to find some maximal subgroups of G with as little effort as possible. When it works, the generators it finds are short words in c and d, giving rise to fairly short straight line programs. It is not aimed at finding specific subgroups $H < G$; it is just a first attempt effort at eliminating some easy cases.

- Fix $g \in G$ (often $g = c$).
- For each short preword w, look at the group $\langle g, \eta(w) \rangle$. If it is a proper subgroup of G, add to the list of candidate groups.

The meaning of 'short' is somewhat context-dependent, but it is usually not worth evaluating more than about 5000 prewords. The generators found are usually not standard generators of H.

Maximal subgroups that were found by trawling are listed in Table 2.3. The corresponding words for the generators are given in Chapter 3.

2.4.2 Searching by conjugacy classes

This method is often effective when searching for a group H by standard generators. Let $\mathcal{P}$ be the standard generating pair characterisation for H, if such exists. When the group H does not have standard generators defined, a preliminary step is finding a good way of generating H, and defining an appropriate pair pre-characterisation $\mathcal{P}$. It is not necessary to ensure that $\mathcal{P}$ is a pair characterisation.

- Find out which H-conjugacy classes the generators of H need to be in. Usually this information is part of $\mathcal{P}$.
- Find out the corresponding G-conjugacy classes (by using the fusion maps from H to G, by looking for likely class multiplication coefficients, or by constructing H by

G	g	Maximal subgroups found		
$M_{12}.2$	c	$L_2(11):2$	$L_2(11):2$	$(2^2 \times A_5):2$
		$2_+^{1+4}.S_3.2$	$3_+^{1+2}:D_8$	$S_4 \times S_3$
		S_5		
$M_{22}.2$	c	$L_3(4):2$	$2^3:L_3(2) \times 2$	$A_6:2^2$
		$L_2(11):2$		
HS.2	c	$M_{22}.2$	$L_3(4).2.2$	$S_8 \times 2$
		$4^3.(2 \times L_3(2))$		
McL.2	c	$U_4(3):2$	$U_3(5):2$	$3^{1+4}:4.S_5$
		$3^4:(M_{10} \times 2)$	$L_3(4):2:2$	$2 \cdot S_8$
		$M_{11} \times 2$		
$J_2.2$	c	$U_3(3):2$	$3.A_6.2.2$	$2^{1+4}.A_5.2$
		$(A_4 \times A_5):2$	$L_3(2):2 \times 2$ S_5	
	d	$(A_5 \times D_{10}).2$		
Suz.2	c	$G_2(4):2$	$3.U_4(3).2.2$	$J_2:2 \times 2$
		$2^{4+6}:3.S_6$	$(A_4 \times L_3(4):2):2$	$2^{2+8}:(S_5 \times S_3)$
		$M_{12}:2 \times 2$	$PGL_2(9) \times A_5).2$	$L_2(25):2$
		S_7		
He.2	c	$S_4(4):4$	$2^2 \cdot L_3(4).D_{12}$	$2_+^{1+6}.L_3(2).2$
		$S_4 \times L_3(2):2$		
	d	$3 \cdot S_7 \times 2$	$(S_5 \times S_5):2$	
$Fi_{22}.2$	c	$2 \cdot U_6(2).2$	$O_8^+(2):S_3 \times 2$	$2^7:S_6(2)$
		$3^5:(2 \times U_4(2):2)$		
$J_3.2$	c	$2^4:(3 \times A_5).2$	$L_2(17) \times 2$	$(3 \times M_{10}):2$
		$3^2.(3 \times 3^2):8.2$	$2_-^{1+4}.S_5$	$2^{2+4}:(S_3 \times S_3)$
O'N.2				
HN.2				
Fi_{24}	c	$Fi_{23} \times 2$	$(2 \times 2 \cdot Fi_{22}):2$	$3^7 \cdot O_7(3):2$
		$3_+^{1+10}:(2 \times U_5(2):2)$		

Table 2.3: Maximal subgroups found by trawling

hand and checking what the classes are). If there is more than one possibility, try them all.

- Find words in the standard generators of G for representatives of the relevant G-conjugacy classes (see section 2.2.3). Let the corresponding elements be called g and h.

- For each preword w, check whether g and $h^{\eta(w)}$ satisfy the predicates in $\mathcal{P}$. If so, add $\langle g, h^{\eta(w)} \rangle$ to the list of candidate groups.

If a large number of candidate groups produced are not of the right isomorphism type, then we can add extra predicates (satisfied by H but not by the incorrect candidate groups) to $\mathcal{P}$.

The likelihood of success of this method can be estimated by considering the sizes of the conjugacy classes involved. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be conjugacy classes of G, with $x_1, x_1' \in \mathcal{C}_1$, $x_2, x_2' \in \mathcal{C}_2$. The probability that (x_1, x_2, x_1x_2) is conjugate to $(x_1', x_2', x_1'x_2')$ is:

$$p = \frac{|C_G(x_1)|.|C_G(x_2)|}{|G|}. \tag{2.4.1}$$

Thus this approach has more chance of working with small groups or when the subgroup H can be generated by elements in small conjugacy classes.

2.5 Maximal subgroups by construction

Sometimes a subgroup cannot easily be generated by random searching, and must be constructed in stages. This is often the case for the p-local subgroups of some of the larger almost simple groups. This method is used for groups which cannot be generated by 2 elements.

The usual plan is to construct as much of a particular subgroup as possible inside a known subgroup, and then 'add on' the missing parts. We produce a number of elements

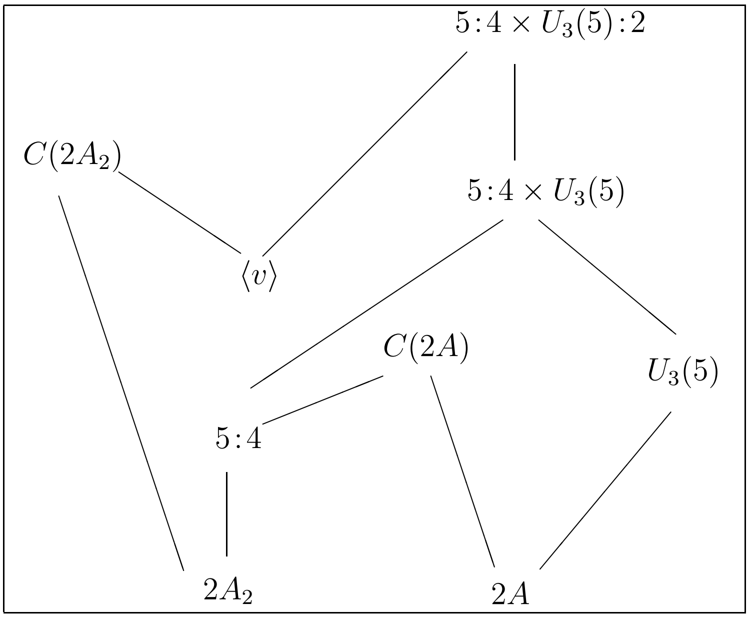


Figure 2.1: Construction of $5:4 \times U_3(5):2 < \mathrm{HN}.2$

inside the maximal subgroup that we are seeking, and stop when we have enough to generate the whole subgroup. After this, we look for a way of 2-generating (or at worst, 3-generating) the subgroup by taking words in the elements that we found.

2.5.1 HN.2

- $K_6 \simeq 5:4 \times U_3(5):2$

  The scheme is shown in Figure 2.1. We found standard generators a, b of a subgroup $U_3(5)$. The element $(ab^2)^4$ is in G-class $2A$, and the subgroup $5:4$ must centralize this element. We constructed its centralizer $C(2A)$ and then looked for elements in $C(2A)$ which centralize both generators of $U_3(5)$. This gave us a subgroup $5:4$ which contains other elements in class $2A$: we chose one of these and denoted it $2A_2$. We constructed the centralizer $C(2A_2)$ of this involution and looked for elements therein which normalize $U_3(5)$ without being inside $U_3(5)$. We found such an element v. Then v, together with $5:4$ and $U_3(5)$, generate the whole of $5:4 \times U_3(5):2$.

- $K_9 \simeq (S_6 \times S_6){:}2^2$

  We take the maximal subgroup S_{12} which we found by looking for its standard generators, and find the subgroup $(S_6 \times S_6) : 2 = S_6 \,\mathrm{wr}\, S_2$: by converting to the permutation on representation on 12 points it is easy to find generators for this subgroup. We then search for an involution which centralizes an involution complementing $S_6 \times S_6$ and which normalizes the whole of $(S_6 \times S_6){:}2$, giving the required maximal subgroup $(S_6 \times S_6){:}2^2$.

  Note that this subgroup cannot be 2-generated.

- $K_7 \simeq 5^{1+4}_{+}{:}2^{1+4}_{-}.5.4.2$

  This subgroup is the normalizer of a $5B$ element. We started by trying to construct as much of this normalizer as possible inside $C(2B)$. We then took $C(2B)$ and searched for a $5B$ element therein. We then looked for elements in $C(2B)$ which normalized this element, giving a subgroup K of $C_{N(5B)}(2B)$ of order 3200 (in fact $C_{N(5B)}(2B)$ itself has order 6400). We then found a different $2B$ element in K (say $2B_2$) and looked for elements in $C(2B_2)$ which normalize the $5B$ element. This was enough to generate the whole of the maximal subgroup.

- $K_{10} \simeq 2^3.2^2.2^6.(3 \times L_3(2)).2$

  This subgroup is the normalizer of an elementary abelian group of order 8 containing 7 $2B$ elements. We find this subgroup $2B^3$ by looking inside the previously constructed $N(2B)$ (which contains a Sylow 2-subgroup of HN.2). We then take two elements $2B$, $2B_2$ inside this 2^3, and construct their centralizers $C(2B)$, $C(2B_2)$. These centralizers are much smaller than HN.2, and it is fairly straightforward to find the normalizers of 2^3 inside these groups: $N_{C(2B)}(2^3)$ and $N_{C(2B_2)}(2^3)$. But these groups together generate the whole of $N(2B^3)$, which is the subgroup we want.

- $K_{11} \simeq 5^2.5.5^2.4A_5.2$

  We used the previously constructed subgroup $N(5B)$ to produce a Sylow 5-subgroup of HN.2, and we found a subgroup $5^2.5.5^2.4.5.2$ which normalizes it (we do not take the full normalizer of the Sylow 5-subgroup inside $N(5B)$ because this normalizer is not a subgroup of $N(5B^2)$). This allows us to find an involution which commutes with the A_5 that we need to find. We search in this involution centralizer for elements which normalize the subgroup $[5^5]$. This gives us enough elements to generate $5^2.5.5^2.4A_5$, which is contained in HN.

  The class fusions for $5^2.5.5^2.4A_5$ and the 2-power map suggest that we try affixing a $2C$ element if the extension is split, or a $4F$ element otherwise. We found a suitable outer $4F$ element after abandoning the search for a $2C$ element. This gave us enough elements to generate the subgroup.

- $K_{13} \simeq 3_+^{1+4}{:}4S_5$

  We considered the subgroup $3_+^{1+4}{:}4A_5$ for which generators were already available in [20]. Taking the 10th power of an element of order 20 gives a $2B$ element with centralizer $3 : 4A_5$ of order 720. We characterised this subgroup and found generators for a conjugate subgroup inside our previously constucted $C(2B)$. The $3B$ in $3 : 4A_5$ can be found easily by powering up. We then found another $2B$ element (called $2B_2$) inside this $3 : 4A_5$, found generators for $C(2B_2)$ and found an element in $C(2B_2)$ normalizing $3B$. This gave enough elements to generate the whole subgroup.

- $K_{12} \simeq 3^4{:}2(S_4 \times S_4).2$

  The scheme is illustrated in Figure 2.2. We used the subgroup $3_+^{1+4}{:}4S_5$ previously constructed to find a subgroup $3^4{:}2(S_3 \times S_3).2$, the normalizer of a Sylow 3-subgroup 3^{4+2} therein. We then took the subgroup $2(S_3 \times S_3).2$ and used the involution

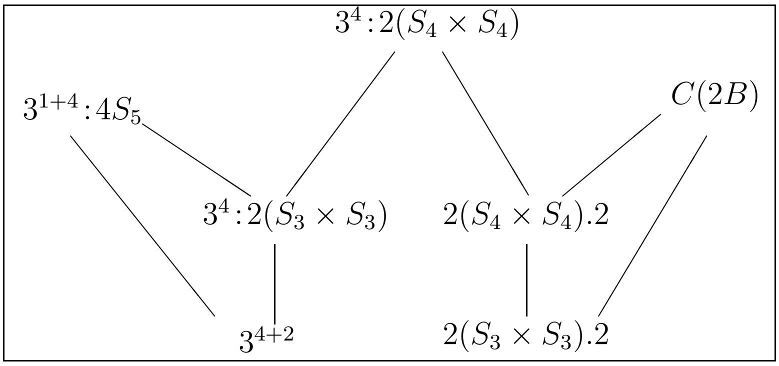


Figure 2.2: Construction of $3^4 : 2(S_4 \times S_4).2 < \mathrm{HN}.2$

centralizer $C(2B)$ to extend it to $2(S_4 \times S_4).2$. The subgroups $3^4:2(S_3 \times S_3).2$ and $2(S_4 \times S_4).2$ together generate $3^4:2(S_4 \times S_4).2$.

2.5.2 Fi_{24}

Fi_{24} is a very large group, and most of its conjugacy classes are very large too. One notable exception is the $2C$ conjugacy class of 306936 Fischer transpositions. Whenever possible, we will try to take advantage of this small conjugacy class.

- $S_3 \times O_8^+(3):S_3$

  We found S_3 by a generating triple $(2C, 2C, 3A)$, and then found $O_8^+(3) : S_3$ by standard generators in the centralizer of one of the $2C$ elements.

- $2^{12} \cdot \mathrm{M}_{24}$

  The scheme is illustrated in Figure 2.3. We took the centralizer of a $2C$ element $\mathrm{Fi}_{23} \times 2$ and found its subgroup Fi_{23}. We then applied the program $\mathrm{Fi}_{23} \dashrightarrow 2^{11} \cdot \mathrm{M}_{23}$ from [20] to get a subgroup $2^{11} \cdot \mathrm{M}_{23}$. A $2C$ element normalizes the subgroup 2^{11} to give $2^{12} \cdot \mathrm{M}_{23}$. We then found a $2A$ element normalizing 2^{12}, giving the full $2^{12} \cdot \mathrm{M}_{24}$.

- $3^3.[3^{10}].(L_3(3) \times 2^2)$

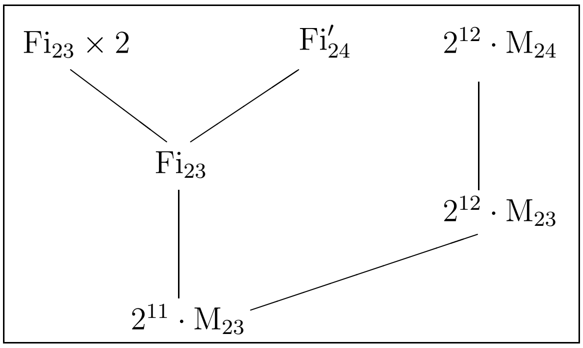


Figure 2.3: Construction of $2^{12} \cdot \mathrm{M}_{24} < \mathrm{Fi}_{24}$

We extended the program $\mathrm{Fi}_{24} \dashrightarrow 3^7 \cdot O_7(3):2$ to a program $\mathrm{Fi}_{24} \overset{\mathrm{std}}{\dashrightarrow}_{O_7(3):2} 3^7 \cdot O_7(3):2$. We then found a program $O_7(3):2 \dashrightarrow 3^{3+3}.(L_3(3) \times 2)$. By concatenating these two programs, we obtained a subgroup $3^7.3^{3+3}.(L_3(3) \times 2)$. By adjoining a suitable $2C$ element, we get the maximal subgroup required.

- $3^2.3^4.3^8.(S_5 \times 2S_4)$

  We produced a program $O_7(3):2 \dashrightarrow 3_+^{1+6}:(S_4 \times 2S_4)$ and adjoined it to the program $\mathrm{Fi}_{24} \overset{\mathrm{std}}{\dashrightarrow}_{O_7(3):2} 3^7 \cdot O_7(3):2$ produced earlier. By adjoining some extra generators of 3^7, we got a subgroup $3^7.3^{1+6}:(S_4 \times 2S_4)$. We then found the elementary abelian normal subgroup $3B^2$ and looked for $2C$ elements normalizing it. This was enough to give the whole maximal subgroup.

- $S_4 \times O_8^+(2):S_3$

  We found S_4 by seeking a triple $(2C, 3A, 4D)$. The $O_8^+(2):S_3$ is generated by $2C$ elements, so we looked for $2C$ elements commuting with the S_4.

- $2^{3+12}.(L_3(2) \times S_6)$

  We took the previously constructed $\mathrm{Fi}_{24} \dashrightarrow 2^{12} \cdot \mathrm{M}_{24}$ and used it to produce $\mathrm{Fi}_{24} \overset{\mathrm{std}}{\dashrightarrow}_{\mathrm{M}_{24}} 2^{12} \cdot \mathrm{M}_{24}$. We took the program $\mathrm{M}_{24} \dashrightarrow 2^6.(L_3(2) \times S_3)$ from [20] and joined the programs together, giving a group $2^{12}.2^6.(L_3(2) \times S_3)$. We extended this to the group we are seeking by taking the normal subgroup $2B^3$ and finding a $2C$ element which normalises it.

- $2^{7+8}.(S_3 \times A_8)$

  We took $\mathrm{Fi}_{24} \overset{\mathrm{std}}{\dashrightarrow}_{\mathrm{M}_{24}} 2^{12} \cdot \mathrm{M}_{24}$ and concatenated $\mathrm{M}_{24} \dashrightarrow 2^4.A_8$ from [20] to give a subgroup $2^{7+8}.(2 \times A_8)$. We found a normal subgroup 2^8 containing 28 $2A$ elements and 35 $2B$ elements, and found a $2C$ element normalizing it. This element together with the subgroup generated before was the required maximal subgroup.

- $(S_3 \times S_3 \times G_2(3)){:}2$

  We found $G_2(3){:}2$ by standard generators (examining the class fusion showed that we needed to look for triples in Fi_{24}-classes $(2D, 4G, 13A)$). We then took the derived subgroup $G_2(3)$ and looked in the centralizer of one of the $2B$ elements therein for some $2A$ elements centralizing the $G_2(3)$. This gives a group $3^2{:}2 \times G_2(3)$. We then found a $2C$ element centralizing $G_2(3)$, giving $S_3 \times S_3 \times G_2(3)$. This together with $G_2(3){:}2$ generates $(S_3 \times S_3 \times G_2(3)){:}2$.

- $S_5 \times S_9$

  We found S_5 by standard generators. The S_9 is generated by transpositions in the symmetric group sense, which turn out to be transpositions in the Fischer sense. Thus S_9 is generated by $2C$ elements commuting with S_5.

- $S_6 \times L_2(8){:}3$

  We found $L_2(8){:}3$ by standard generators in classes $(2B, 3C, 9B)$, and then found S_6 by looking for $2C$ elements commuting with $L_2(8){:}3$.

- $7{:}6 \times S_7$

  The transpositions of the S_7 are $2C$ elements, so the subgroup $7{:}6$ is contained in the centralizer of a $2C$ element (the group $\mathrm{Fi}_{23} \times 2$). We first find the subgroup D_{14}

by looking for triples $(2B, 2B, 7A)$. We then extend this to $7{:}6$ by looking for an element of order 3 in Fi_{23} which commutes with one of the $2B$ elements. Since S_7 is generated by transpositions, we can find the S_7 by searching for $2C$ elements which commute with the generators of $7{:}6$.

- $7_+^{1+2}{:}(6 \times S_3).2$

This subgroup arises from the fusion of two $\mathrm{He}{:}2$ subgroups in Fi_{24}'. It is easy to find one of these $\mathrm{He}{:}2$ subgroups by standard generators, and to find the subgroup $7_+^{1+2}{:}(6 \times S_3)$ which is maximal therein.

The subgroup $(6 \times S_3).2$ acts on the vector space $\mathbb{F}_7^2$:

$$(6 \times S_3).2 \simeq \left\langle \begin{bmatrix} x & 0 \\ 0 & y \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \middle| x, y \in \mathbb{F}_7 \right\rangle \leq GL_2(7) \tag{2.5.1}$$

The involution $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ complements the normal subgroup $6 \times S_3$, so the extension splits. No $2C$ elements were found which normalize the subgroup $7_+^{1+2} : (6 \times S_3)$, so we tried looking for a $2D$ element instead. Such an element must commute with the cube of the 6-element; (a $2B$ involution). A search among the conjugacy classes of $C(2B)$ containing $2D$ elements gave an element normalizing the $7_+^{1+2} : (6 \times S_3)$ we had found earlier.

- $29{:}28$

We found $29{:}2 = D_{58}$ by looking for elements $t, u \in 2B$ whose product has order 29. We then looked for elements of 7 or 14 in $C_G(t)$ which normalize the element of order 29.

Chapter 3

The generators

The straight line programs we produced can be found at [14]. For reasons of practicality, the short line programs themselves are not reproduced here. Instead, we give the GAP code that was used to produce the straight line programs.

In all the GAP code below, the standard generators of the almost simple group are called C and D, and the group produced is called H.

3.1 Maximal subgroups of $\mathrm{M}_{12}.2$

- $\mathrm{M}_{12}.2 \overset{\text{std}}{\dashrightarrow} L_2(11){:}2$ (novelty)

```
H := Group(C, D*D*C*D*C*D*C*D*D*C*D);
```

- $\mathrm{M}_{12}.2 \dashrightarrow L_2(11){:}2$ (novelty)

```
H := Group(C, D*D*C*D*C*D);
```

- $\mathrm{M}_{12}.2 \overset{\text{std}}{\dashrightarrow} L_2(11){:}2$ (ordinary)

```
H := Group(C, ((C*D)^4)^(D*C*D*C*D));
```

- $\mathrm{M}_{12}.2 \dashrightarrow L_2(11){:}2$ (ordinary)

```
H := Group(C, (C*D)^4);
```

- $\mathrm{M}_{12}.2 \dashrightarrow (2^2 \times A_5){:}2$

```
H := Group(C, C*D*C*D*D*C*D*D*C*D*D*C*D*C*D*D*C*D*C*D*C*D);
```

- $\mathrm{M}_{12}.2 \dashrightarrow 2^{1+4}.S_3.2$

```
H := Group(C, C*D*C*D*D*C*D*D*C*D*D*C*D*C*D*D*C*D*C*D*C*D);
```

- $\mathrm{M}_{12}.2 \dashrightarrow 4^2 : D_{12}.2$

```
H := Group((C*D)^4*(C*D*D*C*D)^2*D,
           ((C*D*C*D*C*D*C*D*D*C*D*C*D*D*C*D*D)^3)^(D*C*D*C*D*D*C*D));
```

- $\mathrm{M}_{12}.2 \dashrightarrow 3^{1+2} : D_8$

```
H := Group(C, C*D*D*C*D*C*D*D*C*D*D*C*D*C*D*C*D);
```

- $\mathrm{M}_{12}.2 \dashrightarrow S_4 \times S_3$

```
H := Group(C, C*D*C*D*D*(C*D*C*D*C*D*C*D*D)^2*C*D);
```

- $\mathrm{M}_{12}.2 \overset{\text{std}}{\dashrightarrow} S_5$

```
H := Group(C, (C*D*C*D*C*D)^D);
```

3.2 Maximal subgroups of $\mathrm{M}_{22}.2$

- $\mathrm{M}_{22}.2 \overset{\text{std}}{\dashrightarrow} L_3(4){:}2$

```
H := Group(C, ((C*D*D*C*D)^2)^(D*D*C*D*C*D));
```

- $\mathrm{M}_{22}.2 \dashrightarrow L_3(4){:}2$

```
H := Group(C, C*D*C*D*D*D*C*D*D*C*D);
```

- $\mathrm{M}_{22}.2 \overset{\text{std}}{\dashrightarrow}_{S_6} 2^4{:}A_6{:}2$

```
H := Group(C, (C*D*D*C*D*C*D*D*C*D)^(D*D*C*D));
```

- $\mathrm{M}_{22}.2 \overset{\text{std}}{\dashrightarrow}_{S_5} 2^5{:}S_5$

```
H := Group(C, (C*D*D*C*D*D*C*D)^(C*D*C*D));
```

- $\mathrm{M}_{22}.2 \overset{\text{std}}{\dashrightarrow}_{L_3(2)} 2^3{:}L_3(2) \times 2$

```
H := Group(D*D, (C*D*D)^(D*C*D*C*D*D*D*C*D*D*D*
                           C*D*D*C*D*D*D*C*D*C));
```

- $\mathrm{M}_{22}.2 \dashrightarrow 2^3{:}L_3(2) \times 2$

```
H := Group(C, D^(C*D*D*C*D));
```

- $\mathrm{M}_{22}.2 \overset{\text{std}}{\dashrightarrow} A_6{:}2^2$

```
H := Group(C, C*D*D*C*D*D*C*D);
```

- $\mathrm{M}_{22}.2 \overset{\text{std}}{\dashrightarrow} L_2(11){:}2$

```
H := Group((C*D*D*C*D)^5, (C*D*D*C*D*D)^(D*C*D*C*D*D));
```

- $\mathrm{M}_{22}.2 \dashrightarrow L_2(11){:}2$

```
H := Group(C*D*D*C*D, D);
```

3.3 Maximal subgroups of HS.2

- $\mathrm{HS}.2 \overset{\text{std}}{\dashrightarrow} \mathrm{M}_{22}{:}2$

```
H := Group(C, (C*D*D*D*D*C*D*D*C*D*D*C*D*C*D)^D);
```

- $\mathrm{HS}.2 \dashrightarrow \mathrm{M}_{22}{:}2$

```
H := Group(C, D^(C*D));
```

- $\mathrm{HS}.2 \dashrightarrow L_3(4).2.2$

```
H := Group(C, (C*D*D*C*D*C*D*C*D*D*C*D*D*C*D*C*D));
```

- $\mathrm{HS}.2 \overset{\text{std}}{\dashrightarrow}_{S_8} S_8 \times 2$

```
H := Group(C, (C*D*C*D*C*D*D*D)^(C*D*D*C*D*C*D));
```

- $\mathrm{HS}.2 \dashrightarrow S_8 \times 2$

```
H := Group(C, (C*D)^5);
```

- $\mathrm{HS}.2 \overset{\text{std}}{\dashrightarrow}_{S_6} 2^5.S_6$

```
H := Group(C, (C*D*D*C*D*C*D*D*D*D)^(D*C*D*D*C*D*C*D))
```

- HS.2 $\overset{\text{std}}{\dashrightarrow}_{L_3(2)}$ $4^3.(2 \times L_3(2))$

```
H := Group(C^(D*C), C*D*D*C*D*D*C*D*C*D*C*D*D*D);
```

- HS.2 $\dashrightarrow$ $4^3.(2 \times L_3(2))$

```
H := Group(C, C*D*D*D*C*D*D*C*D);
```

- HS.2 $\overset{\text{std}}{\dashrightarrow}_{S_5}$ $2^{1+6}.S_5$

```
H := Group(C, (C*D*D*C*D*C*D*C*D*D*D)^((D*D*C*D*C)^2*D));
```

- HS.2 $\dashrightarrow$ $(2 \times A_6.2.2).2$

```
H := Group((C*D*D)^10,
           ((C*D*D*C*D*C*D*D)^3)^(D*C*D*D*D*C*D*D*D*C*D*C),
           ((C*D*D*C*D*C*D*C*D*D*D)^2)^(D*D*D*C*D*D*C*D*D*C*D*D*C));
```

- HS.2 $\dashrightarrow$ $5^{1+2}.[2^5]$

```
H := Group((C*D*D)^5,
           ((C*D)^5*D)^(D*C*D*D*D*(C*D*D*D*C*D)^2));
```

- HS.2 $\overset{\text{std}}{\dashrightarrow}_{S_5}$ $5{:}4 \times S_5$

```
H := Group((C*D^2*C*D*C*D^4)^5,
           (((C*D)^5*D)^5)^(D*C*D^4*C*D^2*C*D));
```

3.4 Maximal subgroups of McL.2

- McL.2 $\dashrightarrow$ $U_4(3){:}2$

```
H := Group(C, C*D*C*D*D*C*D*C*D*D*C*D*D*C*D*C*D);
```

- McL.2 $\dashrightarrow$ $U_3(5){:}2$

```
H := Group(C, C*D*D*C*D*C*D*D*C*D*C*D*C*D*D*C*D);
```

- McL.2 $\dashrightarrow$ $3^{1+4}{:}4.S_5$

```
H := Group(C, C*D*D*C*D*D*C*D*C*D*C*D*C*D*D*C*D*C*D*D);
```

- McL.2 $\dashrightarrow$ $3^4:(\mathrm{M}_{10} \times 2)$

```
H := Group(C, (C*D*D*C*D)^3*C*D*(C*D*C*D*D)^2*C*D*D);
```

- McL.2 $\dashrightarrow$ $L_3(4):2:2$

```
H := Group(C, C*D*C*D*D*(C*D)^9);
```

- McL.2 $\overset{\text{std}}{\dashrightarrow}$ $2 \cdot S_8$

```
H := Group(((C*D*C*D*C*D*C*D*C*D*D*C*D*C*D*D)^5)^(C*D*D),
           ((C*D)^3*(C*D*D)^3)^((C*D*C*D*D*C*D)^2*
                (C*D)^5*(C*D*D*C*D)^2));
```

- McL.2 $\dashrightarrow$ $2 \cdot S_8$

```
H := Group(C, (C*D*C*D*D*C*D)^2*(C*D)^2);
```

- McL.2 $\overset{\text{std}}{\dashrightarrow}_{\mathrm{M}_{11}}$ $\mathrm{M}_{11} \times 2$

```
H := Group(C, (C*D*C*D*C*D*D*C*D*D*C*D*C*D*D)^
               ((C*D*C*D*D*C*D)^2*(C*D)^4));
```

- McL.2 $\dashrightarrow$ $\mathrm{M}_{11} \times 2$

```
H := Group(C, C*D*(C*D*D)^3*(C*D)^6);
```

- McL.2 $\dashrightarrow$ $5^{1+2}:3:8.2$

```
H := Group((C*D)^6*(C*D*D)^2,
           ((C*D)*(C*D*C*D*C*D*D)^2)^(D*((C*D*D)^3*C*D)^2*
                C*D*D*C*D*C));
```

- McL.2 $\dashrightarrow$ $2^{2+4}:(S_3 \times S_3)$

```
H := Group((C*D*C*D*C*D*D*C*D*D*C*D*C*D*D)^2,
           (C*D*C*D*C*D*D)^(D*D*C*D*C*D*C*D));
```

3.5 Maximal subgroups of $\mathrm{J}_2.2$

- $\mathrm{J}_2.2 \stackrel{\text{std}}{\dashrightarrow} U_3(3){:}2$

```
H := Group(C, (C*D^3*(C*D)^6)^(C*D*C*D*C*D*D));
```

- $\mathrm{J}_2.2 \dashrightarrow U_3(3){:}2$

```
H := Group(C, (C*D*D)^2*(C*D)^3);
```

- $\mathrm{J}_2.2 \stackrel{\text{std}}{\dashrightarrow} 3.A_6.2.2$

```
H := Group(C, (C*D*D*D*(C*D)^6)^(D*D*C*D*C*D));
```

- $\mathrm{J}_2.2 \dashrightarrow 3.A_6.2.2$

```
H := Group(C, C*D*D*(C*D*D*C*D*D*C*D)^2);
```

- $\mathrm{J}_2.2 \dashrightarrow 2^{1+4}.A_5.2$

```
H := Group(C, D*D*C*D*D*C*D*D*C);
```

- $\mathrm{J}_2.2 \dashrightarrow 2^{2+4}{:}(3 \times S_3).2$

```
H := Group(C, (C*D*D*C*D*C*D*D)^2*C*D*C*D*D);
```

- $\mathrm{J}_2.2 \dashrightarrow (A_4 \times A_5){:}2$

```
H := Group(C, D*D*C*D*C*D*C*D);
```

- $\mathrm{J}_2.2 \dashrightarrow (A_5 \times D_{10}).2$

```
H := Group(C*D*(C*D*D)^2*(C*D)^4, D);
```

- $\mathrm{J}_2.2 \stackrel{\text{std}}{\dashrightarrow}_{L_3(2)2} L_3(2){:}2 \times 2$

```
H := Group(C, (C*D*D*(C*D)^4)^(D^4*C*D*C*D^4));
```

- $\mathrm{J}_2.2 \dashrightarrow L_3(2){:}2 \times 2$

```
H := Group(C, C*D*C*D*D*(C*D)^4*C*D*D*C*D);
```

- $\mathrm{J}_2.2 \dashrightarrow 5^2{:}(4 \times S_3)$

```
H := Group((C*D*D)^12, ((C*D*D*C*D*C*D)^3)^(C*D*C*D*D*C*D));
```

- $\mathrm{J}_2.2 \stackrel{\text{std}}{\dashrightarrow} S_5$

```
H := Group(C, ((C*D*D*C*D*C*D)^3)^(D*C*D^-1*C));
```

- $\mathrm{J}_2.2 \dashrightarrow S_5$

```
H := Group(C, C*D*C*D*C*D*D*C*D*C*D);
```

3.6 Maximal subgroups of Suz.2

- $\mathrm{Suz}.2 \overset{\mathrm{std}}{\dashrightarrow} G_2(4){:}2$

```
H := Group(C, ((C*D*D*(C*D*D*C*D*C*D)^2*(C*D)^4)^3)^(D*D*C*D));
```

- $\mathrm{Suz}.2 \dashrightarrow G_2(4){:}2$

```
H := Group(C, C*D*D*C*D*D*C*D*C*D*C*D*C*D*D*C*D);
```

- $\mathrm{Suz}.2 \dashrightarrow 3.U_4(3).2.2$

```
H := Group(C, D*(C*D*D)^4*(C*D)^3);
```

- $\mathrm{Suz}.2 \overset{\mathrm{std}}{\dashrightarrow} U_5(2){:}2$

```
H := Group((C*D*D*C*D*D*C*D*C*D*C*D*C*D*C*D)^5,
           ((C*D*D*(C*D*D*C*D*C*D)^2*(C*D)^4)^3)^
                 ((D*D*C)^2*D*C*D*(C*D*C*D*D)^2*C));
```

- $\mathrm{Suz}.2 \overset{\mathrm{std}}{\dashrightarrow}_{U_4(2).2} 2^{1+6}.U_4(2).2$

```
H := Group(C, (C*D*D*C*D*D*C*D*C*D*C*D*C*D)^
                 (C*D*C*D*C*D*D*(C*D)^6));
```

- $\mathrm{Suz}.2 \dashrightarrow 2^{1+6}.U_4(2).2$

```
H := Group(C, C*D*(C*D*C*D*D)^2*(C*D)^8);
```

- $\mathrm{Suz}.2 \overset{\mathrm{std}}{\dashrightarrow}_{\mathrm{M}_{11}} 3^5.(\mathrm{M}_{11} \times 2)$

```
H := Group((C*D*D*C*D*C*D*C*D*C*D)^4,
           ((C*D*D*(C*D*D*C*D*C*D)^2*(C*D)^4)^3)^
                  (D^2*(C*D*C*D*C*D^2*C*D^2)^2*
                  C*D^2*C*D*C*D^2*C*D^2*C*D));
```

- $\mathrm{Suz}.2 \overset{\mathrm{std}}{\dashrightarrow}_{\mathrm{J}_2 2} \mathrm{J}_2{:}2 \times 2$

```
H := Group(C, (C*D*D*C*D*C*D*D*C*D*C*D)^
                (C*D*(C*D*D)^3*(C*D*D*C*D)^2*
                C*D*C*D*D*C));
```

- Suz.2 $\dashrightarrow$ $J_2{:}2 \times 2$

```
H := Group(C, C*D*C*D*D*(C*D*D*C*D)^3);
```

- Suz.2 $\overset{\text{std}}{\dashrightarrow}_{S_6}$ $2^{4+6}{:}3.S_6$

```
H := Group(C, ((C*D*D*C*D*C*D)^2*C*D*C*D)^
                  (D*C*D*C*D*D*(C*D)^5*(C*D*D)^2*C));
```

- Suz.2 $\dashrightarrow$ $2^{4+6}{:}3.S_6$

```
H := Group(C, C*D*C*D*D*(C*D)^5*C*D*D*C*D);
```

- Suz.2 $\dashrightarrow$ $(A_4 \times L_3(4){:}2){:}2$

```
H := Group(C, C*D*(C*D*D)^3*(C*D)^3*C*D*D*C*D*C*D);
```

- Suz.2 $\dashrightarrow$ $2^{2+8}{:}(S_5 \times S_3)$

```
H := Group(C, C*D*D*(C*D)^3*C*D*D*(C*D*D*C*D)^2);
```

- Suz.2 $\overset{\text{std}}{\dashrightarrow}_{M_{12}2}$ $M_{12}{:}2 \times 2$

```
H := Group((C*D)^14,
            (C*D*D*(C*D*D*C*D*C*D*D*C*D*C*D)^2)^
             (D*D*C*D*C*D*D*(C*D)^4*(C*D*D)^4*C*D*C*D*C));
```

- Suz.2 $\dashrightarrow$ $M_{12}{:}2 \times 2$

```
H := Group(C, (D*C*D*C*D*C*D)^3);
```

- Suz.2 $\dashrightarrow$ $3^{2+4}{:}2.(S_4 \times D_8)$

```
H := Group((C*D*D*C*D*C*D*C*D*C*D)^6,
            ((C*D*D*C*D*C*D*C*D*C*D)^3)^
                 (D*D*C*D*C*D*C*D*C*D*C*D*D*
                  C*D*D*C*D*C*D*D*C*D*C*D*D*
                  C*D*C*D*C*D*C*D),
            (C*D*D*C*D*D*C*D*C*D*C*D*D*C*D*C*
                 D*C*D*C*D*C*D*C*D)^
                 (C*D*D*C*D*C*D*C*D*D*C*D*D*
                  C*D*D*C*D*C*D*C*D*
                  C*D*D*C*D*C*D*D*C*D*D*C*D*C));
```

- Suz.2 $\overset{\text{std}}{\dashrightarrow}_{P\Gamma L_2(9)}$ $(PGL_2(9) \times A_5).2$

```
H := Group(C, ((C*D*D*C*D*C*D*C*D*C*D)^2)^
                 ((C*D)^5*C*D*D*(C*D)^3));
```

- Suz.2 $\dashrightarrow$ $(PGL_2(9) \times A_5).2$

```
H := Group(C, C*D*(C*D*D)^3*(C*D*C*D*D*C*D)^2*
                (C*D*D)^2*C*D*C*D);
```

- Suz.2 $\overset{\text{std}}{\dashrightarrow}_{S_6}$ $(A_6 \times 3^2{:}8).2$

```
H := Group(C, ((C*D*D)^2*(C*D)^7)^(D*D*C*D*(C*D*D)^4*
                  C*D*(C*D*C*D*D)^2*C*D*C*D));
```

- Suz.2 $\dashrightarrow$ $L_2(25){:}2$

```
H := Group(C, (C*D)^5*C*D*D*C*D*C*D*D*(C*D)^5);
```

- Suz.2 $\overset{\text{std}}{\dashrightarrow}$ S_7

```
H := Group((C*D*D*C*D*D*C*D*C*D*C*D*C*D*C*D)^5,
           (C*D*D*C*D*D*C*D*C*D*D*(C*D)^5*
                C*D*D*C*D)^(D*(C*D*D*C*D*C*D*D)^2*
                (C*D)^2*(C*D*D)^4*C));
```

- Suz.2 $\dashrightarrow$ S_7

```
H := Group(C, C*D*D*(C*D*D)^3*(C*D)^4*C*D*D*C*D);
```

3.7 Maximal subgroups of He.2

- He.2 $\overset{\text{std}}{\dashrightarrow}$ $S_4(4){:}4$

```
H := Group((C*D*C*D*D*D)^6,
           (C*D*D*C*D*C*D*C*D^5)^(D*C*D*C*D*D*C*D*D));
```

- He.2 $\dashrightarrow$ $S_4(4){:}4$

```
H := Group(C, (D^2*C*D)^(C*D));
```

- He.2 $\overset{\text{std}}{\dashrightarrow}_{L_3(4).D_{12}} 2^2 \cdot L_3(4).D_{12}$

```
H := Group(D^3, ((C*D^3*C*D)^3)^(C*D*C*D*C*D^3*C*D^3*C));
```

- He.2 $\dashrightarrow 2^2 \cdot L_3(4).D_{12}$

```
M := Group(D^3, D*C*D*C*D^5*C*D^3*C*D*C);
```

- He.2 $\overset{\text{std}}{\dashrightarrow}_{L_3(2).2} 2_+^{1+6}.L_3(2).2$

```
H := Group(C*D^5*C*D*D*C*D*C*D,
           ((C*D*C*D*D*C*D*D)^4)^
               (D^4*C*D*C*D^4*C*D^5*C*D*D));
```

- He.2 $\dashrightarrow 2_+^{1+6}.L_3(2).2$

```
H := Group(C, C*D^3*C*D^3*C*D*C*D*C*D^3);
```

- He.2 $\overset{\text{std}}{\dashrightarrow}_{2.L_2(7).2} 7^2{:}2L_2(7).2$

```
H := Group(C*D*C*D*C*D^3,
           ((C*D*C*D*D*C*D*D)^8)^
               (D*D*C*D^4*C*D*C*D*D*C*D^3));
```

- He.2 $\overset{\text{std}}{\dashrightarrow}_{3 \cdot S_7} 3 \cdot S_7 \times 2$

```
H := Group((C*D*C*D*D*D)^6, D^3*C*D*C*D*D*C*D*C);
```

- He.2 $\dashrightarrow 3 \cdot S_7 \times 2$

```
H := Group(C*D*C*D*D*D*C*D*C*D, D);
```

- He.2 $\dashrightarrow (S_5 \times S_5){:}2$

```
H := Group(D,  (C*D^3)^3*D*(C*D^3*C*D)^2*C);
```

- He.2 $\dashrightarrow 2^{4+4}.(S_3 \times S_3).2$

```
H := Group(C, (D^4*C)^2*D*D*C*D*C*D*C*D*C*D*D*C);
```

- He.2 $\dashrightarrow 7_+^{1+2}{:}(S_3 \times 6)$

```
H := Group(C*D^3*C*D*C*D*D*C*D,
           (C*D^3*C*D*C*D*D*C*D)^
               (C*D*C*D^-1*C*D^-1*C*D*C*D*C*D^-1));
```

- He.2 $\overset{\text{std}}{\dashrightarrow}_{L_3(2)2}$ $S_4 \times L_3(2){:}2$

```
H := Group(D^3, (C*D*C*D^3)^
             (D^2*C*D^4*C*D*C*D^4*C*D*C*D*C*D*C));
```

- He.2 $\dashrightarrow$ $S_4 \times L_3(2){:}2$

```
H := Group(C, C*D^5*C*D*D*(C*D)^4*C*D*D*C*D*D*D*C*D);
```

- He.2 $\overset{\text{std}}{\dashrightarrow}_{L_3(2)}$ $7{:}6 \times L_3(2)$

```
H := Group(D^3, ((C*D*D*C*D*D*C*D)^8)^(D*C*D^4*C*D^3*C));
```

- He.2 $\dashrightarrow$ $5^2{:}4S_4$

```
H := Group(C*D*D*C*D*D*C*D*C*D*C*D,
           (((C*D*D)^2*C*D)^8)^(D*D*C*D^3*(C*D*D)^3*C));
```

3.8 Maximal subgroups of $\mathrm{Fi}_{22}.2$

- $\mathrm{Fi}_{22}.2$ $\overset{\text{std}}{\dashrightarrow}$ $2 \cdot U_6(2).2$

```
H := Group(D^9, ((C*D^11*C*D^5*C*D)^3)^(D^5*C*D^4*C));
```

- $\mathrm{Fi}_{22}.2$ $\dashrightarrow$ $2 \cdot U_6(2).2$

```
H := Group(C, (C*D)^3);
```

- $\mathrm{Fi}_{22}.2$ $\dashrightarrow$ $O_8^+(2){:}S_3 \times 2$

```
H := Group(C, (D*D*C*D*C)^2*D);
```

- $\mathrm{Fi}_{22}.2$ $\overset{\text{std}}{\dashrightarrow}_{\mathrm{M}_{22}2}$ $2^{10}{:}\mathrm{M}_{22}{:}2$

```
H := Group((C*D^15*C*D*D)^3,
           ((C*D^7)^3)^(D*C*D^14*C*D^17*C));
```

- $\mathrm{Fi}_{22}.2$ $\dashrightarrow$ $2^7{:}S_6(2)$

```
H := Group(C, D^4*C*D^15*C*D^4);
```

- $\mathrm{Fi}_{22}.2 \dashrightarrow (2 \times 2_+^{1+8}{:}U_4(2){:}2){:}2$

```
H := Group(C*D^14*C*D^5*C*D,
           (C*D^4*C*D^3)^(D*D*(C*D*C*D*D*C*D)^2*
                D*C*D*D*(C*D)^4*D*(C*D)^3*D));
```

- $\mathrm{Fi}_{22}.2 \dashrightarrow S_3 \times U_4(3).(2^2)_{122}$

```
H := Group(C*D^9,
           (C*D^-1*C*D*D*C*D)^(D^6*C*D^14*C*D^10*C*D^2*C*D),
           ((C*D^6)^3)^(D^3*C*D^4*C*D^-3*C*D^5));
```

- $\mathrm{Fi}_{22}.2 \overset{\text{std}}{\dashrightarrow} {}^2F_4(2)$

```
H := Group((C*D^2*C*D*C*D^2*C*D*C*D*C*D)^12,
           ((C*D^2*C*D^2*C*D)^3)^(D*C*D^4*D^4*D^4*C*D));
```

- $\mathrm{Fi}_{22}.2 \dashrightarrow 2^{5+8}{:}(S_3 \times S_6)$

```
H := Group(C*D^4*C*D^2*C*D,
           (C*D^-1*C*D^6*C*D)^(D*D*C*D^6*C*D^2*C*D^13*C*D^2));
```

- $\mathrm{Fi}_{22}.2 \dashrightarrow 3^5{:}(2 \times U_4(2){:}2)$

```
H := Group(C, (C*D*D*C*D)^2*(C*D)^3);
```

- $\mathrm{Fi}_{22}.2 \dashrightarrow 3_+^{1+6}{:}2^{3+4}{:}3^2{:}2.2$

```
H := Group(C*D^7*C*D*C*D,
           (C*D^-2*C*D^-2*C*D)^(D^4*C*D^-2*C*D^3*C*D));
```

- $\mathrm{Fi}_{22}.2 \overset{\text{std}}{\dashrightarrow} G_2(3){:}2$

```
H := Group((C*D*D*C*D*D*C*D*D*C*D)^15,
           ((C*D*D*C*D*D*C*D)^3)^(C*D^6*C*D^2*C*D^4));
```

- $\mathrm{Fi}_{22}.2 \overset{\text{std}}{\dashrightarrow} \mathrm{M}_{12}{:}2$

```
H := Group((C*D^3)^15, ((C*D^7)^4)^(D^16*C*D^8*C*D^5*C*D));
```

3.9 Maximal subgroups of $\mathrm{J}_3.2$

- $\mathrm{J}_3.2 \overset{\text{std}}{\dashrightarrow} L_2(16){:}4$

```
H := Group((C*D)^12, ((C*D^2*C*D)^2*C*D)^
                              ((C*D^2)^3*C*D*(C*D*C*D^2)^2));
```

- $\mathrm{J}_3.2 \dashrightarrow 2^4{:}(3 \times A_5).2$

```
H := Group(C, C*D*(C*D^2)^3*C*D*C*D^2*(C*D)^6);
```

- $\mathrm{J}_3.2 \overset{\text{std}}{\dashrightarrow}_{L_2(17)} L_2(17) \times 2$

```
H := Group(C, ((C*D^2*C*D)^3)^(C*D*C*D*C*D^2*C*D^2*C*D*C*D));
```

- $\mathrm{J}_3.2 \dashrightarrow L_2(17) \times 2$

```
H := Group(C, C*D*(C*D^2)^5*(C*D)^4);
```

- $\mathrm{J}_3.2 \dashrightarrow (3 \times \mathrm{M}_{10}){:}2$

```
H := Group(C, C*D*(C*D^2)^4*C*D*(C*D*C*D^2)^2*C*D*C*D);
```

- $\mathrm{J}_3.2 \dashrightarrow 3^2.(3 \times 3^2){:}8.2$

```
H := Group(C, C*D*(C*D^2)^4*C*D*C*D*C*D^2*C*D);
```

- $\mathrm{J}_3.2 \dashrightarrow 2_-^{1+4}.S_5$

```
H := Group(C, (C*D)^4*(C*D^2*C*D)^2*(C*D)^4);
```

- $\mathrm{J}_3.2 \dashrightarrow 2^{2+4}{:}(S_3 \times S_3)$

```
H := Group(C, C*D*C*D^2*C*D^2*(C*D)^6*C*D^2*C*D);
```

- $\mathrm{J}_3.2 \overset{\text{std}}{\dashrightarrow} 19{:}18$

```
H := Group(C, (C*D^2*(C*D)^4)^(C*D*C*D^2*(C*D)^2*(C*D^2)^2*(C*D)^2));
```

3.10 Maximal subgroups of O'N.2

- O'N.2 $\dashrightarrow$ $J_1 \times 2$

```
H := Group(D*D*C*D*(C*(C)^(D*D*C*D))^5,
           D*D*C*D*C*D*C*D*(C*(C)^(D*D*C*D*C*D*C*D))^9);
```

- O'N.2 $\dashrightarrow$ $4_2 \cdot L_3(4) \cdot 2^2$

```
H := Group((C*D*D)^18*C,
           C*D*C*D*D*C*D*D*C*D*(D*D*(D*D)^(C*D*C*D*D*C*D*D*C*D))^9);
```

- O'N.2 $\dashrightarrow$ $(3^2{:}4 \times A_6) \cdot 2^2$

```
H := Group(C, ((C*D*D*C*D*D*C*D*C*D)^4)^(C*D*(C*D*D)^5));
```

- O'N.2 $\dashrightarrow$ $3^4{:}2^{1+4}D_{10}.2$

```
H := Group(D, ((C*D^3*C*D*C*D)^7)^(C*D*C*D^3*C*D^2*
                 (C*D^3)^3*C*D*C*D*C));
```

- O'N.2 $\dashrightarrow$ $4^3 \cdot L_3(2) \times 2$

```
H := Group(D*D, (((C*D^3)^2*(C*D)^3)^5)^(C*D*
                 (C*D^3*C*D^2)^2*C*D*C));
```

- O'N.2 $\dashrightarrow$ $7_+^{1+2}{:}(3 \times D_{16})$

```
H := Group((C*D^3*C*D^3*(C*D)^3)^5,
           (D*D)^((C*D^3)^2*(C*D)^3*(C*D^3)^3));
```

- O'N.2 $\dashrightarrow$ $31{:}30$

```
H := Group(C, ((C*D^3*C*D^3*C*D*C*D*C*D)^4)^
                (C*D*D*C*D*C*D*D*D*C*D*C*D));
```

- O'N.2 $\overset{\text{std}}{\dashrightarrow}$ $A_6{:}2_2$

```
H := Group(C, ((C*D^3*C*D^3*(C*D)^3)^10)^((C*D*D*C*D)^2*D));
```

- O'N.2 $\overset{\text{std}}{\dashrightarrow}$ $L_2(7){:}2$

```
H := Group(C, ((C*D^3*C*D^3*(C*D)^3)^10)^
                (D^2*C*D^3*C*D*C*D^2*C*D^3*C*D));
```

3.11 Maximal subgroups of HN.2

In this section and the next, the generators are calculated stepwise (words for the generators are only implicit). This is for reasons of efficiency both in terms of paper and computing time — computing the generators of H from c and d in one go would take a prohibitively long time.

- HN.2 $\overset{\text{std}}{\dashrightarrow}$ S_{12}

```
H := Group(C, ((C*D^3*(C*D*C*D^2)^2)^4)^
          (C*D*C*D^2*C*D^2*C*D^2*C*D*C*D*C*
                  D^3*C*D*C*D^3*C*D*D));
```

- HN.2 $\dashrightarrow$ $4 \cdot \mathrm{HS}.2$

```
2a := (C*D*D*C*D*D*C*D)^4;;
i1 := D*(2a*2a^D)^2;;
i2 := D*C*(2a*2a^(D*C))^2;;
i7 := C*D*D*D*C*D*C*D*(2a*2a^(C*D*D*D*C*D*C*D))^2;;
H := Group(i1*i2, i2*i7);
```

- HN.2 $\overset{\text{std}}{\dashrightarrow}$ $U_3(8){:}6$

```
H := Group(C, ((C*D)^14)^(C*D^2*C*D^4*C*D*C*D));
```

- HN.2 $\dashrightarrow$ $2^{1+8}_{+}.(A_5 \times A_5).2^2$

```
2b := (C*D^3*C*D)^5;;
H := Group(D*D*C*D*D*(2b*2b^(D*D*C*D*D))^5,
                C*D^3*C*D*(2b*2b^(C*D*D*C*D*C*D*D))^15);
```

- HN.2 $\dashrightarrow$ $5{:}4 \times U_3(5){:}2$

```
4b := (C*D*D*C*D*D*C*D)^2;;
u := 4b^(C*D^3*C*D^4*C*D^2*C*D^4*C*D^3*C*D^4*C*D*C*D);;
a := (C*u*C*u^2*C*u)^4;;
b := (C*u*(C*u*C*u^2)^2)^4;;
2a := (a*b^2)^4;;
t1 := C*D^3*(2a*2a^(C*D^3))^2;;
```

```
t3 := C*D*C*D*C*D^3*(2a*2a^(C*D*C*D*C*D^3))^2;;
e1 := t1*t3^5*t1^3*t3^12*t1*t3^2*t1;;
e2 := t1^2*t3^10*t1*t3^15*t1^5*t3^5*t1^4;;
2aa := e1^2;;
u1 := C*D*(2aa*2aa^(C*D))^2;;
u2 := C*D^3*(2aa*2aa^(C*D^3))^2;;
v := u1^7*u2*u1^2*u2^5;;
H := Group(a*e1, b*e2*v);
```

- HN.2 $\dashrightarrow 5_+^{1+4}{:}2_-^{1+4}.5.4.2$

```
2b := (C*D^3*C*D)^5;;
a := D*D*C*D*D*(2b*2b^(D*D*C*D*D))^5;;
b := C*D^3*C*D*(2b*2b^(C*D*D*C*D*C*D*D))^15;;
p := a^7*b^6*a*b^2;;
q := (a^2*b^7)^(a^2*b^6*a*b^7);;
z := (p*q)^2;;
r := C*D*(z*z^(C*D))^5;;
s := C*D^3*(z*z^(C*D^3))^10*C*(z*z^C)^10;;
t := r^6*s^3*r^6*s^9*r^3;;
H := Group(p*q, q*t);
```

- HN.2 $\overset{\text{std}}{\dashrightarrow}_{U_4(2).2}\ 2^6 \cdot U_4(2).2$

```
H := Group(C, ((C*D^2)^2)^((C*D^3)^2*C*D*C*D^2*
                    C*D*(C*D^3)^2*C*D));
```

- HN.2 $\dashrightarrow (S_6 \times S_6){:}2^2$

```
g1 := C;;
conj := C*D*(C*D*D)^3*C*D*(C*D*C*D^3)^2*C*D*D;;
g2 := ((C*D^3*(C*D*C*D^2)^2)^4)^conj;;
e := (g1*g2^2*g1*g2*g1*g2)^3;;
f := (g2^4*g1*g2*g1*g2*g1);;
g := C^(C*D^4*C*D*C*D^3*C*D^4*C*D^2*C*D^3*C*D^2*C*D*C*D);;
H := Group(e, f, g);
```

- HN.2 $\dashrightarrow 2^3.2^2.2^6.(3 \times L_3(2)).2$

```
2b := (C*D^3*C*D)^5;;
x := D*D*C*D*D*(2b*2b^(D*D*C*D*D))^5;;
y := C*D^3*C*D*(2b*2b^(C*D*D*C*D*C*D*D))^15;;
```

```
r := x^5*y^4;;
2b2 := (x^11*y^3)^2;;
xp := C*D*D*C*(2b2*2b2^(C*D*D*C))^12*C*D*(2b2*2b2^(C*D))^7;;
yp := (2b2*2b2^(D*D*C*D*D))^15;;
t := yp*xp^3*yp*xp^4;;
H := Group(r,t);
```

- HN.2 --→ $5^2.5.5^2.4A_5.2$

```
2b := (C*D^3*C*D)^5;;
a := D*D*C*D*D*(2b*2b^(D*D*C*D*D))^5;;
b := C*D^3*C*D*(2b*2b^(C*D*D*C*D*C*D*D))^15;;
p := a^7*b^6*a*b^2;;
q := (a^2*b^7)^(a^2*b^6*a*b^7);;
z := (p*q)^2;;
r := C*D*(z*z^(C*D))^5;;
s := C*D^3*(z*z^(C*D^3))^10*C*(z*z^C)^10;;
t := r^6*s^3*r^6*s^9*r^3;;
c := p*q;;
d := q*t;;
f := (c*d*d)^(c^2*d^3);;
inv := f^10;;
u := (C*D^4*(inv*inv^(C*D^4))^7)^5;;
v := (C*D*(inv*inv^(C*D))^12)^3;;
4c := u^2*v^3*u*v^3*u*v^2*(u*v)^2;;
conj := (C*D^3)^3*D*C*D*C*D^4*C*D^3*C*D*C*D^2*C*D;;
outer := ((C*D^2*C*D*C*D)^5)^conj;;
H := Group(4c*f, outer);
```

- HN.2 --→ $3^4{:}2(S_4 \times S_4).2$

```
2b := (C*D^3*C*D)^5;;
a := D*D*C*D*D*(2b*2b^(D*D*C*D*D))^5;;
b := C*D^3*C*D*(2b*2b^(C*D*D*C*D*C*D*D))^15;;
c := b^4*a^2;;
d := (b^8*a^2)^(b^5*a^14*b^5*a^19);;
f := c*d*d;;
m1 := C*D*D*(f*f^(C*D*D))^7;;
m2 := C*D*C*D*D*(f*f^(C*D*C*D*D))^12;;
e := m1^4*m2^7*m1^5*m2*m1^2*m2^6;;
y := e^2*c^3*e*c^-3;;
inv := (e^3*c^3*e*c^4)^6;;
l2 := C*D*(inv*inv^(C*D))^7;;
```

```
l1 := C*D*D*(inv*inv^(C*D*D))^5;;
z := l1*l2^3*l1^9*l2^3*l1*l2;;
H := Group(y, z);
```

- HN.2 --→ $3_+^{1+4}{:}4S_5$

```
2b := (C*D^3*C*D)^5;;
a := D*D*C*D*D*(2b*2b^(D*D*C*D*D))^5;;
b := C*D^3*C*D*(2b*2b^(C*D*D*C*D*C*D*D))^15;;
c := b^4*a^2;;
d := (b^8*a^2)^(b^5*a^14*b^5*a^19);;
z := c*d*d;;
l1 := C*D*D*(z*z^(C*D*D))^7;;
l2 := C*D*C*D*D*(z*z^(C*D*C*D*D))^12;;
e := l1^4*l2^7*l1^5*l2*l1^2*l2^6;;
H := Group(c,e);
```

3.12 Maximal subgroups of Fi_{24}

- Fi_{24} --→ $\mathrm{Fi}_{23} \times 2$

```
H := Group(C, D^(C*D));
```

- Fi_{24} --→ $(2 \times 2 \cdot \mathrm{Fi}_{22}){:}2$

```
H := Group(C, (C*D*D*C*D)^(C*D^5*C*D^5*C*D^4));
```

- Fi_{24} --→ $S_3 \times O_8^+(3){:}S_3$

```
t := D*(C*C^D);;
u := D^2*(C*C^(D^2));;
H := Group(C*t^28*u^4*t^40*u^5, C^D*t^31*u^5*t^3*u^5*t^5);
```

- Fi_{24} --→ $O_{10}^-(2){:}2$

```
H := Group(C, ((C*D^4*C*D)^6)^(C*D^3*C*D^3*C*D^2*C*D*C*D^4));
```

- Fi_{24} --→ $3^7 \cdot O_7(3){:}2$

```
H := Group(C, ((C*D^6)^2*C*D)^(C*D^5*C*D^2*C*D^3));
```

- Fi_{24} --→ $3_+^{1+10}{:}(2 \times U_5(2){:}2)$

```
H := Group(C, (C*D^4*C*D^2*C*D*C*D^2*C*D)^(C*D*C*D^2*C*D^6));
```

- Fi_{24} --→ $2^{12} \cdot \mathrm{M}_{24}$

```
2a := (C*D^2)^10;;
u := D^(C*D);;
t1 := C*u;;
t2 := u^2;;
a := (t1*t2)^14;;
conj := (t1^7*t2)^4;;
b := conj^(t2^2*t1^4);;
g1 := a;;
g2 := (a*b)^7*(b*a)^2;;
g3 := 2a^(C*D*(C*D^3)^2*C*D^7*C*D^4*C*D^5*C*D^3);;
g4 := C^(C*D^5*(C*D^2)^3*C*D);;
H := Group(g2, g3*g4);
```

- Fi_{24} --→ $(2 \times 2^2 \cdot U_6(2)){:}S_3$

```
t := (C*D^4)^3;;
u := (C*D^2)^3;;
v := (C*D^3*C*D^2)^2;;
H := Group(u*(t*t^u), v*(t*t^v)^17);
```

- Fi_{24} --→ $2_+^{1+12} \cdot 3U_4(3).(2^2)_{122}$

```
H := Group(C*D*D*C*D*C*D*(D^4*(D^4)^
                (C*D*D*C*D*C*D))^19,
           C*D^3*(D^4*(D^4)^(C*D^3))*
                ((D^4)*(D^4)^(C*D^3*C*D*D))^6);
```

- Fi_{24} --→ $3^3.[3^{10}].(L_3(2) \times 2^2)$

```
a := C;;
b := ((C*D^6)^2*C*D)^(C*D^5*C*D^2*C*D^3);;
c := (b^5*C*b^2*C*b*C)^13;;
d := ((C*b)^4)^(b^4);;
g1 := (d*c)^13;;
g2 := (d^3*c)^12;;
g3 := (d^4*c)^4;;
g4 := (d^2*c*d)^12;;
g5 := C^(C*D^2*C*D^5*C*D^6*C*D*C*D^5*C*D^6);;
H := Group(g1*g2*g3, g4*g5);
```

- Fi_{24} --→ $3^2.3^4.3^8.(S_5 \times 2S_4)$

```
b := ((C*D^6)^2*C*D)^(C*D^5*C*D^2*C*D^3);;
c := (b^5*C*b^2*C*b*C)^13;;
d := ((C*b)^4)^(b^4);;
g1 := d^4*c;;
g2 := (d^2*c*d^4*c*d^3*c*d*c)^4;;
g3 := C^(C*D*C*D^3*(C*D^7)^2*C*D^4*C*D);;
H := Group(g1, g2*g3);
```

- $\mathrm{Fi}_{24} \dashrightarrow S_4 \times O_8^+(2){:}S_3$

```
CD := C*D;;
CD2 := CD * D;;
CD3 := CD2 * D;;
CD4 := CD3 * D;;
CD5 := CD4 * D;;
CD6 := CD5 * D;;
CD7 := CD6 * D;;
t1 := C^(CD2*CD3*CD*CD6*CD*CD);;
t2 := C^(CD2*CD5*CD7*CD6*CD2*CD);;
t3 := C^(CD*CD3*CD7*CD3*CD4*CD);;
t4 := C^(CD4*CD3*CD7*CD4*CD5*CD);;
t5 := C^(CD2*CD*CD3*CD6*CD7*CD);;
t6 := C;;
t7 := (CD4^CD2)^2;;
g1 := t1*t2*t3*t6;;
g2 := t4*t5*t3*t7;;
H := Group(g1, g2);
```

- $\mathrm{Fi}_{24} \dashrightarrow 2^{3+12}.(L_3(2) \times S_6)$

```
2a := (C*D^2)^10;;
u := D^(C*D);;
t1 := C*u;;
t2 := u^2;;
a := (t1*t2)^14;;
conj := (t1^7*t2)^4;;
b := conj^(t2^2*t1^4);;
aa := (a*b)^7*(b*a)^2;;
g3 := 2a^(C*D*(C*D^3)^2*C*D^7*C*D^4*C*D^5*C*D^3);;
g4 := C^(C*D^5*(C*D^2)^3*C*D);;
bb := g3*g4;;
u1 := (bb*aa*bb*aa*bb*aa^2)^5;;
u2 := (bb*aa^3*bb*aa*bb*aa^3)^5;;
```

```
u3 := u2^(bb*aa*bb);;
v1 := u1;;
v2 := u3;;
v3 := v1*v2;;
v4 := v3*v2;;
v5 := v3*v4;;
v6 := v3*v5;;
v9 := v3^7;;
v10 := v5^11;;
v8 := v9*v10;;
v5 := v6^3;;
v9 := v8^-1;;
v7 := v9*v5;;
v2 := v7*v8;;
h3 := C^(C*D^5*C*D*C*D^4*C*D^3*C*D^2);;
h4 := v1*u3^3*h3;;
H := Group(h4, v2);
```

- $\mathrm{Fi}_{24} \dashrightarrow 2^{7+8}.(S_3 \times A_8)$

```
2a := (C*D^2)^10;;
u := D^(C*D);;
t1 := C*u;;
t2 := u^2;;
a := (t1*t2)^14;;
conj := (t1^7*t2)^4;;
b := conj^(t2^2*t1^4);;
g1 := a;;
g2 := (a*b)^7*(b*a)^2;;
g3 := 2a^(C*D*(C*D^3)^2*C*D^7*C*D^4*C*D^5*C*D^3);;
g4 := C^(C*D^5*(C*D^2)^3*C*D);;
aa := g2;;
bb := g3*g4;;
v1 := (bb*aa*bb*aa*bb*aa^2)^5;;
u2 := (bb*aa^3*bb*aa*bb*aa^3)^5;;
v2 := u2^(bb*aa*bb);;
v3 := v1*v2;;
v4 := v3*v2;;
v5 := v3*v4;;
v6 := v5^9;;
v7 := v3^11;;
v8 := v7*v6;;
v6 := v5*v5;;
```

```
v7 := v3*v6;;
v6 := v7*v7;;
v7 := v8^-1;;
v5 := v7*v6;;
v2 := v5*v8;;
gg := C^(C*D^3*C*D*C*D^5*C*D*C*D^4*C*D^3*C*D^6);;
H := Group(v1, v2 * bb^2 * gg);
```

- $\mathrm{Fi}_{24} \dashrightarrow (S_3 \times S_3 \times G_2(3)){:}2$

```
2d := (C*D^4)^3;;
4g := (C*D^2*C*D*C*D*C*D)^7;;
k1 := 2d;;
k2 := 4g^(C*D^4*C*D^2*C*D^7*C*D^5*C*D*C*D^3*C*D^3*C*D^2);;
t := k1 * k2;;
u := k2^2;;
g1 := D*(u*u^D)^19;;
g2 := C*D^3*(u*u^(C*D^3))^14;;
ccl := ((g1*g2)^2*(g1*g2^2)^14)^3;;
t1 := ccl^(g1^8*g2^5*g1^3);;
t2 := ccl^(g1^7*g2^3*g1^6*g2);;
g5 := C^(C*D^6*C*D*C*D^5*C*D^5*C*D^7*C*D^2*C*D^3*C*D^4);;
H := Group(t1*k1, t1*t2*k2*g5);
```

- $\mathrm{Fi}_{24} \dashrightarrow S_5 \times S_9$

```
z1 := C;;
z2 := ((C*D*D)^5)^(C*D);;
inv := z1*z1^(z1^z2*z2);;
a := D*inv*inv^D;;
b := D^2*(inv*inv^(D^2))^2;;
2c := (a*b^8)^9;;
z3 := 2c^(a^3*b^3);;
z4 := z3^(b^6);;
z5 := z4^(b^6);;
z6 := z5^(b^6);;
z7 := 2c^(a^8*b^2*a);;
z8 := z6^(b*a);;
z9 := 2c^(a^2*b^14*a^4);;
z10 := z5^(b^5*a^4);;
H := Group(z1*z10,z2*z3*z4*z5*z6*z7*z8*z9*z10);
```

- $\mathrm{Fi}_{24} \dashrightarrow S_6 \times L_2(8){:}3$

```
CD := C * D;;     CD2 := CD * D;;  CD3 := CD2 * D;;
CD4 := CD3 * D;; CD5 := CD4 * D;;
CD6 := CD5 * D;; CD7 := CD6 * D;;
l1 := D^4;;
3c := (CD7*CD3*(CD)^2)^4;;
l2 := 3c^(CD4*CD5*CD4*CD6*CD5*CD7*CD5*(CD)^2);;
z1 := C^(CD3*CD2*CD4*CD3*CD3*CD5*C);;
z2 := C^(CD*(CD6)^3*CD4*(CD)^2);;
z3 := C^(CD*CD3*(CD)^2*(CD2)^2*CD);;
z6 := C^(CD6*(CD*CD4)^2*CD2*(CD)^2);;
z8 := C^(CD*(CD6)^2*(CD6*CD)*CD3*(CD6*CD));;
S6 := Group(z1,z2,z3,z6,z8);
H := Group(z2*z6*z8*z3*l1, z1*l2);
```

- $\mathrm{Fi}_{24} \dashrightarrow 7{:}6 \times S_7$

```
2b := D^4;;
D14 := Group(2b, 2b^(C*D^7*C*D^3*(C*D)^2));
IC2 := Group(t,u);
q := 2b;;
r := 2b^(C*D^7*C*D^3*(C*D)^2);;
t := C*D*D*C*D*C*D*(D^4*(D^4)^(C*D*D*C*D*C*D))^19;;
u := C*D^3*(D^4*(D^4)^(C*D^3))*((D^4)*(D^4)^(C*D^3*C*D*D))^6;;
v := ((u*t^2)^6)^(t^5*u^3*t^5*u^2*t*u^3*t^2*u^2*t^4*u^2);;
z1 := C^(C*D*C*D^5*C*D*C*D^5*C);;
z2 := z1^(D^4);;
z3 := C^(C*D*C*D*C*D^4*C*D^7*C*D^3*C*D^2);;
z4 := C^(C*D^5*C*D^2*C*D^4*C*D^6*C*D^3*C*D^5*C);;
z5 := C^(C*D^3*C*D^4*C*D^2*C*D^4*C*D^5*C*D^5*C);;
z6 := C^(C*D*C*D^2*C*D^4*C*D^6*C*D^7*C*D*C*D^2);;
H := Group(q*z1*z3*z6 ,r*v*z2*z3*z4*z5);
```

- $\mathrm{Fi}_{24} \dashrightarrow 7_+^{1+2}{:}(6 \times S_3).2$

```
2b := D^4;;
6b := (C*D^4*C*D*C*D^2*C*D*C*D)^7;;
t := 2b;;
u := 6b^(C*D*C*D^5*C*D*C*D^2*C*D^5*C*D^2*C*D);;
p := t*u^3*t*u*t*u*u*t*u;;
q := p^(t*u*t*u^-1*t*u^-1*t*u*t*u*t*u^-1);;
t := p^2*q^3;;
z := t^3;;
i1 := D*(z*z^D)^19;;
```

```
i2 := D*C*(z*z^(D*C))^13;;
i3 := (z*z^(C*D^3))^6;;
J := i2*i1;;
K := i3;;
H := Group(p, q * ((J^7*K)^3)^(J^3*K*J^8*K*J^6*K*J^2*K));
```

- $\mathrm{Fi}_{24} \dashrightarrow 29{:}28$

```
t := C*D*D*C*D*C*D*(D^4*(D^4)^(C*D*D*C*D*C*D))^19;;
u := C*D^3*(D^4*(D^4)^(C*D^3))*((D^4)*(D^4)^(C*D^3*C*D*D))^6;;
H := Group(D^4*(D^4)^(C*D^3*(C*D)^4),
   (t*u^3*t*u*t*u)^(t^3*u^2*t*u*t^4*u^3*t^2*u*t^5*u*
                    t*u*t*u^2*t^4*u*t*u^3));
```

Chapter 4

Proof of correctness

Let $G = S.2$ be a sporadic almost simple group with $n = n_G$ conjugacy classes of maximal subgroups. Recall that $K_i = K_{G;i}$ is (a representative of) the ith conjugacy class of maximal subgroups of G. For each integer $1 \leq i \leq n$, we define $H_i = H_{G;i}$ to be the subgroup of G produced by the first straight line program labelled $G \dashrightarrow K_i$ in [14].

This chapter will be devoted to a proof of the following theorem:

Theorem 4.1 *Let* $S \in \{\mathrm{M}_{12}, \mathrm{M}_{22}, \mathrm{HS}, \mathrm{McL}, \mathrm{J}_2, \mathrm{Suz}, \mathrm{He}, \mathrm{Fi}_{22}, \mathrm{J}_3, \mathrm{O'N}, \mathrm{HN}, \mathrm{Fi}_{24}'\}$ *and let* $G = \mathrm{Aut}(S) = S.2$. *Then for* $1 \leq i \leq n_G$ *we have that* $H_{G;i}$ *is conjugate to* $K_{G;i}$ *in* G.

The theorem says essentially that the straight line programs are labelled correctly, and together produce all the maximal subgroups of G.

A large section of the proof is computational. Computing each set of generators from a straight line program and finding the orbit shapes and group orders required for the proof took approximately 9 minutes on a single CPU (see section 1.6 for a description of the computer hardware used).

4.1 General arguments

Let G be a group. For convenience, we define a *k-portfolio* for G to be a set of k non-conjugate maximal subgroups of G. A *full portfolio* for G is an n_G-portfolio (i.e. a set containing one subgroup from each conjugacy class of maximal subgroups of G).

Theorem 4.2 *Let $\{K_i : 1 \leq i \leq n\}$ be a full portfolio for G, and let $\{H_i : 1 \leq i \leq n\}$ be a set of subgroups of G. Suppose that:*

(A1) *For each i, H_i is a proper subgroup of G.*

(A2) *For each i, $|K_i|$ divides $|H_i|$.*

(A3) *For each i, j (distinct), H_i is not contained in any conjugate of H_j.*

Then $\{H_i : 1 \leq i \leq n\}$ is a full portfolio for G and $|H_i| = |K_i|$ for each i.

Proof. By renumbering the K_i and H_i if necessary, we may assume that:

$$|K_1| \geq |K_2| \geq \cdots \geq |K_n| \tag{4.1.1}$$

We will prove by induction on k that $\{H_i : 1 \leq i \leq k\}$ is a k-portfolio for G satisfying $|H_i| = |K_i|$ (for $1 \leq i \leq k$). Setting $k = n$ gives the theorem.

H_1 is a proper subgroup of G by (A1), so it must be contained in a conjugate of some K_j. But $|K_j| \leq |H_1|$ for all j by equation 4.1.1 and (A2). So H_1 must be a maximal subgroup itself, and $|H_1| = |K_1|$.

Now suppose that $1 \leq k < n$ and that $\{H_i : 1 \leq i \leq k\}$ is a k-portfolio of G with $|H_i| = |K_i|$ for each i, $1 \leq i \leq k$. The subgroup H_{k+1} is a proper subgroup of G by (A1), so it must be contained in a conjugate of some K_j.

Suppose $|H_{k+1}| < |K_j|$. By (A2), $|K_{k+1}| \leq |H_{k+1}|$, and so $|K_{k+1}| < |K_j|$. But by the induction hypothesis and equation 4.1.1, the set $\{H_i : 1 \leq i \leq k\}$ contains representatives for all conjugacy classes of maximal subgroups of G with order strictly greater than $|K_{k+1}|$, so in particular, K_j is conjugate to H_i for some $1 \leq i \leq k$. But this means that H_{k+1} is contained in a conjugate of H_i, contradicting (A3).

So $|H_{k+1}| = |K_j|$, and H_{k+1} is conjugate to K_j (and so is a maximal subgroup). Moreover, $|H_{k+1}| \geq |K_{k+1}|$ by (A2), and we must have equality because the maximal subgroups with orders strictly greater than $|K_{k+1}|$ are already represented among the H_i with $1 \leq i \leq k$, and by (A3), H_{k+1} is not conjugate to any of them. So $\{H_i : 1 \leq i \leq k+1\}$ is a $(k+1)$-portfolio for G. ■

To apply Theorem 4.2, we have to show that the three hypotheses (A1), (A2) and (A3) hold.

(A1) In all cases, one of the following is easy to show:

- In some permutation representation, G is transitive and H_i is not.
- The order of H_i is less than the order of G.
- The G-conjugacy classes of the generators of H_i are all contained in the simple group G'.

Nothing more will be said about this hypothesis in the sequel.

(A2) Showing that $|K_i|$ divides $|H_i|$ is easy for the small degree groups: in these cases, it is easy enough to show equality directly. For larger degree groups such as $G = \mathrm{Fi}_{24}$, we usually find the order of some quotient of H_i (by considering its action on one of its orbits).

(A3) In a large number of cases, we can show that there are no inclusions up to conjugacy among the H_i by looking at the orders or orbit shapes for each group. Observe that if H_i is contained in a conjugate of H_j, then the order of H_i must divide the order of H_j, and it must be possible to produce the orbit shape for H_j by gluing together orbits of H_i.

The case $j = 1$ is treated differently. To show that H_i is not contained in any conjugate of H_1, we use the *outer class argument.* We find the conjugacy classes of the generators of H_j and H_1. If all the generators for H_1 lie in inner classes, and at least one generator for H_i lies in an outer class, then clearly H_i cannot be contained in any conjugate of H_1.

This does not always eliminate every case, and sometimes more specific arguments are needed.

Note that to prove Theorem 4.1 for G it is not enough to show that $\{H_i : 1 \leq i \leq n\}$ is a full portfolio. Once we have applied Theorem 4.2 to a group G, we may have some extra work to show that H_i is conjugate to K_i rather than to some other K_j with the same order as K_i.

4.2 Groups of small degree

In this section we will deal with groups which have a fairly small permutation representation. For these groups, many computations are fast and have modest storage requirements. In particular, calculating the order of a subgroup is feasible. The groups concerned are M_{12}.2, M_{22}.2, HS.2, McL.2, J_2.2, Suz.2, He.2, Fi_{22}.2 and J_3.2.

For each group we give a table of orbit shapes and orders for the subgroups H_i. For reference purposes and to clarify the numbering of the K_i, putative structures are given as well, but these structures are neither proved nor used here. In all cases, the group orders in the table prove statement (A2).

To prove (A3), note that the outer class argument holds in each case, and for many $i, j > 1$ distinct, the group orders and orbit shapes in the table show that H_i is contained

Group	Order	Orbit shape	Putative structure
H_1	95040	12^2	M_{12}
H_2	1320	$2^1\ 22^1$	$L_2(11){:}2$
H_3	1320	24^1	$L_2(11){:}2$
H_4	480	24^1	$(2^2 \times A_5){:}2$
H_5	384	$8^1\ 16^1$	$2_+^{1+4}.S_3.2$
H_6	384	24^1	$4^2{:}D_{12}.2$
H_7	216	$6^1\ 18^1$	$3_+^{1+2}{:}D_8$
H_8	144	24^1	$S_4 \times S_3$
H_9	120	24^1	S_5

Table 4.1: Orders and orbit shapes of H_i for $M_{12}.2$

in no conjugate of H_j. If there are cases left over, they will be dealt with in the appropriate subsection below.

This will allow us to apply Theorem 4.2. To show that H_i is conjugate to K_i, the group orders in the table are often enough (i.e. H_i is known to be a maximal subgroup, and the only maximal subgroup with the right order is K_i). Ambiguous cases are also dealt with separately in the appropriate subsection.

When referring to a specific subgroup $H_{G;i}$, its generators will be referred to as a and b.

4.2.1 $M_{12}.2$

$M_{12}.2$ has a permutation representation on 24 points. The sizes of the H_i and their orbits are given in Table 4.1.

We need to show that H_9 is not contained in any conjugate of H_3 or H_4. We will instead prove directly the stronger statement that H_9 is conjugate to K_9, the novel S_5 maximal subgroup.

The subgroup H_9 has order 120 and its generators satisfy the relations in this presentation:

$$\langle a, b : a^2 = b^4 = (ab)^5 = [a, b]^3 = 1\rangle \simeq S_5 \tag{4.2.1}$$

and so H_9 is isomorphic to S_5. (This presentation, like all the presentations referred to in this chapter, can be found in [20]). The first generator is a $2C$ element, and the sum of all the $(2C, 4, 5)$ class coefficients for $\mathrm{M}_{12}.2$ is 5. There is a contribution of 2+2 from the two conjugacy classes of $L_2(11){:}2 < \mathrm{M}_{12}{:}2$, but in fact there is no subgroup S_5 inside $L_2(11) : 2$ (the $(2, 4, 5)$ elements of $L_2(11) : 2$ actually generate the whole group). Thus there is a unique conjugacy class of subgroups S_5 in $\mathrm{M}_{12} : 2$ which contains some outer elements. But the maximal subgroup K_9 of $\mathrm{M}_{12}.2$ is such a subgroup S_5, so H_9 must be conjugate to K_9.

There are two cases where the maximal subgroup cannot be distinguished by its order alone.

- H_2 and H_3.

  From the orbit shapes, it is clear that H_2 is a novelty arising from the fusion of two point stabilizing M_{11} subgroups, whereas H_3 is the ordinary $L_2(11) : 2$ which normalizes the maximal $L_2(11)$ in M_{12}.

- H_5 and H_6.

  Both these subgroups have order 384, but H_5 has a normal subgroup of order 2 (generated by b^6) and H_6 has no such normal subgroup. So we have:

$$H_5 \simeq K_5 = 2^{1+4}{:}S_3.2 \tag{4.2.2}$$

  and

$$H_6 \simeq K_6 = 4^2{:}D_{12}{:}2 \tag{4.2.3}$$

4.2.2 $\mathrm{M}_{22}.2$

$\mathrm{M}_{22}.2$ has a permutation representation on 22 points. Table 4.2 is enough to prove (A3), and the subgroups H_i can be distinguished by their orders alone.

Group	Order	Orbit shape	Putative structure
H_1	443520	22^1	M_{22}
H_2	40320	$1^1\ 21^1$	$L_3(4):2$
H_3	11520	$6^1\ 16^1$	$2^4:A_6:2$
H_4	3840	$2^1\ 20^1$	$2^5:S_5$
H_5	2688	$8^1\ 14^1$	$2^3:L_3(2) \times 2$
H_6	1440	$10^1\ 12^1$	$A_6 \cdot 2^2$
H_7	1320	22^1	$L_2(11):2$

Table 4.2: Orders and orbit shapes of H_i for $M_{22}.2$

Group	Order	Orbit shape	Putative structure
H_1	44352000	$1^1\ 22^1\ 77^1$	HS
H_2	887040	100^1	$M_{22}:2$
H_3	80640	$2^1\ 42^1\ 56^1$	$L_3(4):2^2$
H_4	80640	$30^1\ 70^1$	$S_8 \times 2$
H_5	23040	$2^1\ 6^1\ 32^1\ 60^1$	$2^5.S_6$
H_6	21504	$8^1\ 28^1\ 64^1$	$4^3.(2 \times L_3(2))$
H_7	15360	$20^1\ 80^1$	$2^{1+6}.S_5$
H_8	5760	$40^1\ 60^1$	$(2 \times A_6.2.2).2$
H_9	4000	100^1	$5^{1+2}.[2^5]$
H_{10}	2400	100^1	$5:4 \times S_5$

Table 4.3: Orders and orbit shapes of H_i for HS.2

4.2.3 HS.2

HS.2 has a permutation representation on 100 points. Table 4.3 gives the orbit shapes and orders for the H_i.

We need to eliminate the possibility that H_2 might contain conjugates of H_3, H_4 or H_8. By considering the permutation representation of HS.2 on 1100 points (corresponding

Group	Order	Orbit shape	Putative structure
H_1	898128000	275^1	McL
H_2	6531840	$1^1\ 112^1\ 162^1$	$U_4(3){:}2$
H_3	252000	$100^1\ 175^1$	$U_3(5){:}2$
H_4	116640	$5^1\ 270^1$	$3^{1+4}{:}4.S_5$
H_5	116640	$2^1\ 30^1\ 81^1\ 162^1$	$3^4{:}(\mathrm{M}_{10}\times 2)$
H_6	80640	$2^1\ 56^1\ 105^1\ 112^1$	$L_3(4){:}2{:}2$
H_7	80640	$35^1\ 240^1$	$2\cdot S_8$
H_8	15840	$11^1\ 22^1\ 110^1\ 132^1$	$\mathrm{M}_{11}\times 2$
H_9	6000	$125^1\ 150^1$	$5^{1+2}{:}3{:}8.2$
H_{10}	2304	$3^1\ 8^1\ 32^1\ 64^1\ 72^1\ 96^1$	$2^{2+4}{:}(S_3\times S_3)$

Table 4.4: Orders and orbit shapes of H_i for McL.2

to cosets of $S_8 \times 2$), we get the following orbit shapes:

$$H_2:\quad 330^1\ 770^1$$
$$H_3:\quad 120^1\ 420^1\ 560^1$$
$$H_4:\quad 1^1\ 28^1\ 105^1\ 336^1\ 630^1$$
$$H_8:\quad 2^1\ 24^1\ 30^1\ 144^1\ 180^1\ 360^2$$

which are incompatible.

To distinguish H_3 from H_4, note that b^7 is the central element of order 2 in H_4, whereas H_3 has trivial centre.

4.2.4 McL.2

McL.2 has a permutation representation on 275 points. Table 4.4 gives the orbit shapes and orders for the H_i.

The only possible inclusion left to eliminate is $H_{10} \lesssim H_7$. Suppose this were the case. Then by comparing the orders of the groups, we see that H_{10} must contain a conjugate of a Sylow 2-subgroup of H_7. Now H_7 contains a central element of order 2 (namely a^2), and this element must be in every Sylow 2-subgroup. Thus H_{10} contains this element,

Group	Order	Orbit shape	Putative structure
H_1	604800	100^1	J_2
H_2	12096	$1^1\ 36^1\ 63^1$	$U_3(3):2$
H_3	4320	$10^1\ 90^1$	$3.A_6.2.2$
H_4	3840	$20^1\ 80^1$	$2^{1+4}.A_5.2$
H_5	2304	$12^1\ 24^1\ 64^1$	$2^{2+4}:(3 \times S_3).2$
H_6	1440	$40^1\ 60^1$	$(A_4 \times A_5):2$
H_7	1200	100^1	$(A_5 \times D_{10}).2$
H_8	672	$2^1\ 14^1\ 42^2$	$L_3(2):2 \times 2$
H_9	600	100^1	$5^2:(4 \times S_3)$
H_{10}	120	$20^2\ 60^1$	S_5

Table 4.5: Orders and orbit shapes of H_i for $J_2.2$

and because $H_{10} \lesssim H_7$, this element must be central in H_{10}. But computationally we find that H_{10} has trivial centre, so our original assumption must be false. Hence there are no inclusions among the H_i.

To distinguish H_6 and H_7, observe that H_6 has trivial centre, whereas H_7 (as we have already observed) contains a central involution. To distinguish H_4 and H_5, observe that H_4' has a centre of order 3 (generated by $[a, b^2]^4$), whereas H_5' has a trivial centre.

4.2.5 $J_2.2$

$J_2.2$ has a permutation representation on 100 points. Table 4.5 gives the orbit shapes and orders for the H_i.

We must eliminate the following possible inclusions:

- $H_9 \lesssim H_7$.

  Suppose this were the case; say $H_9 < H_7^g$. Then H_9 is index 2 in H_7^g and hence a normal subgroup. Now $(H_9)^{(3)} = 1$, so H_9 is a soluble group. Thus H_7^g is a cyclic extension of a soluble group, and hence H_7 is soluble. However, $(H_7)^{(2)} = (H_7)^{(3)} \neq 1$; a contradiction.

- $H_{10} \lesssim H_4$, H_6, H_7 or H_9.

We will show directly that H_{10} is the maximal S_5 of J$_2$.2. There are (at least) 3 conjugacy classes of subgroups A_5 in J$_2$.2 (namely, inside $K_6 \simeq (A_4 \times A_5) : 2$, $K_7 \simeq (A_5 \times D_{10}).2$ and $K_{10} \simeq S_5$). The centralizers of these A_5 subgroups have orders respectively 12, 10 and 1. But there are only three non-zero $(2,3,5)$ class multiplication coefficients for J$_2$.2, and they are given by:

$$\xi(2A, 3B, 5A) = \frac{1}{12} \tag{4.2.4}$$
$$\xi(2B, 3A, 5B) = \frac{1}{10} \tag{4.2.5}$$
$$\xi(2B, 3B, 5B) = 1 \tag{4.2.6}$$

Thus it suffices to show that H_{10} contains a $(2B, 3B, 5B)$ A_5 subgroup, because:

- K_{10} is the normalizer of such a subgroup;
- such an A_5 would have index 2 (and would therefore be normal) in H_{10}; and
- H_{10} has the same size as K_{10}.

We take b^2 as the $2B$ element and b^3aba as the $3B$ element. The product of these is a $5B$ element. Thus H_{10} is the maximal S_5 in J$_2$.2.

The H_i can be distinguished from each other by their orders alone.

4.2.6 Suz.2

Suz.2 has a permutation representation on 1782 points. Table 4.6 gives the orders and orbit shapes for the H_i.

The inclusions not eliminated by Table 4.6 are:

- $H_{11} \lesssim H_4$.

  Note that $abab^2$ in H_{11} is an element of order 22. By considering the order of H_4

Group	Order	Orbit shape	Putative structure
H_1	448345497600	1782^1	Suz
H_2	503193600	$1^1\ 416^1\ 1365^1$	$G_2(4){:}2$
H_3	39191040	$162^1\ 1620^1$	$3.U_4(3).2.2$
H_4	27371520	1782^1	$U_5(2){:}2$
H_5	6635520	$54^1\ 1728^1$	$2^{1+6}.U_4(2).2$
H_6	3849120	1782^1	$3^5.(\mathrm{M}_{11} \times 2)$
H_7	2419200	$2^1\ 100^1\ 630^1\ 1050^1$	$\mathrm{J}_2{:}2 \times 2$
H_8	2211840	$6^1\ 240^1\ 1536^1$	$2^{4+6}{:}3.S_6$
H_9	967680	$42^1\ 480^1\ 1260^1$	$(A_4 \times L_3(4){:}2){:}2$
H_{10}	737280	$2^1\ 20^1\ 96^1\ 640^1\ 1024^1$	$2^{2+8}{:}(S_5 \times S_3)$
H_{11}	380160	$792^1\ 990^1$	$\mathrm{M}_{12}{:}2 \times 2$
H_{12}	279936	$324^1\ 1458^1$	$3^{2+4}{:}2.(S_4 \times D_8)$
H_{13}	86400	$12^1\ 150^1\ 720^1\ 900^1$	$(PGL_2(9) \times A_5).2$
H_{14}	51840	$72^1\ 270^1\ 1440^1$	$(A_6 \times 3^2{:}8).2$
H_{15}	15600	$52^1\ 130^1\ 300^1\ 650^2$	$L_2(25){:}2$
H_{16}	5040	$42^1\ 120^1\ 210^2\ 360^1\ 420^2$	S_7

Table 4.6: Orders and orbit shapes of H_i for Suz.2

and the fact that one of its generators is not contained in Suz, we see that it must be the maximal $U_5(2){:}2$, and this group does not contain elements of order 22.

- $H_{14} \lesssim H_4$.

  The second generator b of H_{14} has order 40, and $U_5(2) : 2$ does not contain any elements of order 40.

- $H_{16} \lesssim H_3$.

  Computationally, we find that the elements of order 4 in H_3 are in classes $4A$, $4C$ and $4E$, whereas the elements of order 4 in H_{16} are in classes $4D$ and $4F$.

- $H_{16} \lesssim H_9$.

  By considering the order of H_9 and the conjugacy classes of its generators, we see that it must be the maximal $(A_4 \times L_3(4){:}2){:}2$. H_{16} has order 7! and its generators

Group	Order	Orbit shape	Putative structure
H_1	4030387200	2058^1	He
H_2	3916800	$1^1\ 272^1\ 425^1\ 1360^1$	$S_4(4){:}4$
H_3	967680	$42^1\ 336^1\ 1680^1$	$2^2 \cdot L_3(4).D_{12}$
H_4	43008	$42^1\ 672^1\ 1344^1$	$2_+^{1+6}.L_3(2).2$
H_5	32928	2058^1	$7^2{:}2L_2(7).2$
H_6	30240	$42^1\ 630^2\ 756^1$	$3 \cdot S_7 \times 2$
H_7	28800	$2^1\ 36^1\ 50^1\ 120^1\ 150^1\ 200^1$ $600^1\ 900^1$	$(S_5 \times S_5){:}2$
H_8	18432	$2^1\ 8^1\ 48^1\ 64^1\ 128^1\ 144^1$ $384^1\ 512^1\ 768^1$	$2^{4+4}.(S_3 \times S_3).2$
H_9	12348	2058^1	$7_+^{1+2}{:}(S_3 \times 6)$
H_{10}	8064	$42^1\ 84^1\ 252^1\ 672^1\ 1008^1$	$S_4 \times L_3(2){:}2$
H_{11}	7056	$294^1\ 1764^1$	$7{:}6 \times L_3(2)$
H_{12}	2400	$3^1\ 30^1\ 75^1\ 150^1\ 300^2\ 600^2$	$5^2{:}4S_4$

Table 4.7: Orders and orbit shapes of H_i for He.2

satisfy the presentation:

$$\langle a, b : a^2 = b^6 = (ab)^7 = [a,b]^3 = [a,bab]^2 = 1\rangle \simeq S_7 \qquad (4.2.7)$$

Thus $H_{16} \simeq S_7$. Suppose $H_{16} \lesssim H_9$. Then by repeatedly taking derived subgroups, our assumption tells us that A_7 can be embedded in $L_3(4)$. But the largest maximal subgroup of $L_3(4)$ has order $960 < \frac{1}{2}7!$, giving a contradiction.

4.2.7 He.2

He.2 has a permutation representation on 2058 points. Table 4.7 gives the orders and orbit shapes of the H_i.

The only potential inclusion not ruled out by Table 4.7 is H_{10} in H_3. In the permutation representation of He.2 on 8330 points (corresponding to cosets of $K_3 \simeq 2^2 \cdot L_3(4).D_{12}$),

Group	Order	Orbit shape	Putative structure
H_1	64561751654400	3510^1	Fi_{22}
H_2	36787322880	$1^1\ 693^1\ 2816^1$	$2 \cdot U_6(2).2$
H_3	2090188800	$360^1\ 3150^1$	$O_8^+(2){:}S_3 \times 2$
H_4	908328960	$22^1\ 1024^1\ 2464^1$	$2^{10}{:}\mathrm{M}_{22}{:}2$
H_5	185794560	$126^1\ 1080^1\ 2304^1$	$2^7{:}S_6(2)$
H_6	106168320	$2^1\ 180^1\ 1024^1\ 2304^1$	$(2 \times 2_+^{1+8}{:}U_4(2){:}2){:}2$
H_7	78382080	$3^1\ 126^1\ 1680^1\ 1701^1$	$S_3 \times U_4(3).(2^2)_{122}$
H_8	35942400	3510^1	${}^2F_4(2)$
H_9	35389440	$6^1\ 48^1\ 1536^1\ 1920^1$	$2^{5+8}{:}(S_3 \times S_6)$
H_{10}	25194240	$108^1\ 486^1\ 2916^1$	$3^5{:}(2 \times U_4(2){:}2)$
H_{11}	10077696	$27^1\ 1296^1\ 2187^1$	$3_+^{1+6}{:}2^{3+4}{:}3^2{:}2.2$
H_{12}	8491392	$702^1\ 2808^1$	$G_2(3){:}2$
H_{13}	190080	$144^1\ 792^1\ 990^1\ 1584^1$	$\mathrm{M}_{12}{:}2$

Table 4.8: Orders and orbit shapes of H_i for $\mathrm{Fi}_{22}.2$

the orbit shapes for H_3 and H_{10} are:

$$H_3: \quad 1^1\ 105^1\ 720^1\ 1344^1\ 1680^1\ 4480^1$$

$$H_{10}: \quad 3^1\ 32^1\ 63^1\ 84^1\ 168^1\ 252^1\ 336^2\ 672^1\ 1008^3\ 1344^1\ 2016^1$$

These orbits are incompatible.

The H_i can be distinguished by their orders alone.

4.2.8 $\mathrm{Fi}_{22}.2$

$\mathrm{Fi}_{22}.2$ has a permutation representation on 3510 points. Table 4.8 gives the orders and orbit shapes of the H_i.

The table rules out all potential inclusions, and the H_i can be distinguished by their orders alone.

4.2.9 $\mathrm{J}_3.2$

$\mathrm{J}_3.2$ has a permutation representation on 6156 points. Table 4.9 gives the orders and orbit shapes of the H_i.

Group	Order	Orbit shape	Putative structure
H_1	50232960	6156^1	J_3
H_2	16320	$1^1\ 85^1\ 120^1\ 510^1\ 680^1\ 2040^1\ 2720^1$	$L_2(16){:}4$
H_3	5760	$6^1\ 30^1\ 120^1\ 480^1\ 720^1\ 1920^1\ 2880^1$	$2^4{:}(3 \times A_5).2$
H_4	4896	$36^1\ 612^2\ 1224^2\ 2448^1$	$L_2(17) \times 2$
H_5	4320	$36^1\ 360^1\ 540^2\ 1080^1\ 1440^1\ 2160^1$	$(3 \times M_{10}){:}2$
H_6	3888	$324^1\ 972^2\ 1944^2$	$3^2.(3 \times 3^2){:}8.2$
H_7	3840	$16^1\ 60^1\ 160^1\ 480^1\ 640^1\ 960^1\ 1920^2$	$2_-^{1+4}.S_5$
H_8	2304	$12^1\ 48^1\ 144^1\ 192^1\ 384^1\ 768^1\ 1152^2\ 2304^1$	$2^{2+4}{:}(S_3 \times S_3)$
H_9	342	342^{18}	$19{:}18$

Table 4.9: Orders and orbit shapes of H_i for $J_3.2$

The subgroup orders in the table rule out any possible inclusions among the H_i. Each of the H_i may be identified by its order alone.

4.3 Groups of large degree

For the groups in this section (O'N.2, HN.2, Fi_{24}), it is sometimes expensive in terms of computer memory and time to compute directly the order of a subgroup. For each group G, we will tabulate the orbit shapes for the H_i in the smallest permutation representation, but will not give the group orders.

To show (A2), we calculate the order of the image of H_i on one of its smaller orbits. (The orbit concerned is underlined in the corresponding table.) If this order is $|K_i|/t$ with $t > 1$, we will give generators for a subgroup of order t contained in the kernel of the action homomorphism. For H_1, this is too computationally expensive: however, we can see that H_1 has the correct order directly because its orbit shape is incompatible with that of any other maximal subgroup.

To show (A3), in most cases the orbit shapes can be seen to be incompatible. The remaining cases are dealt with separately.

Group	Orbit shape	Putative structure
H_1	122760^2	O'N
H_2	$\underline{3080^1}\ 8360^1\ 29260^3\ 58520^1\ 87780^1$	$J_1 \times 2$
H_3	$240^1\ 480^1\ \underline{13440^1}\ 20160^1\ 23040^1\ 26880^1$ $40320^2\ 80640^1$	$4_2 \cdot L_3(4) \cdot 2^2$
H_4	$\underline{360^1}\ 720^1\ 1440^1\ 2160^1\ 3240^1\ 6480^2\ 8640^2$ $12960^2\ 25920^5\ 51840^1$	$(3^2{:}4 \times A_6) \cdot 2^2$
H_5	$\underline{360^1}\ 2160^1\ 3240^1\ 6480^1\ 12960^2$ $25920^2\ 51840^3$	$3^4{:}2^{1+4}D_{10}.2$
H_6	$\underline{336^1}\ 448^1\ 896^2\ 1024^1\ 1792^2\ 3584^1$ $5376^3\ 7168^2\ 10752^{13}\ 21504^3$	$4^3 \cdot L_3(2) \times 2$
H_7	$2^1\ 28^1\ \underline{196^1}\ 392^1\ 686^2\ 1372^1\ 2058^3$ $2744^2\ 4116^4\ 5488^3\ 8232^{10}\ 16464^7$	$7_+^{1+2}{:}(3 \times D_{16})$
H_8	$\underline{310}^9\ 930^{261}$	$31{:}30$
H_9	$\underline{30^1}\ 60^1\ 90^3\ 120^5\ 180^6\ 240^{11}$ $360^{73}\ 720^{298}$	$A_6{:}2_2$
H_{10}	$\underline{14^1}\ 16^1\ 28^1\ 42^3\ 56^2\ 84^6\ 112^{19}$ $168^{76}\ 336^{684}$	$L_2(7){:}2$

Table 4.10: Orbit shapes of H_i for O'N.2

4.3.1 O'N.2

O'N.2 has a permutation representation on 245520 points. Table 4.10 gives the orbit shapes for the H_i.

For several i, we can verify (A2) by observing that H_i has an orbit of size $|K_i|$.

The only putative inclusions we need to eliminate for (A3) are:

- $H_4 \lesssim H_5$.

  Each of these groups has order at least 51840, and neither is contained in O'N. But the only other K_i with orders which are multiples of 51840 are K_4 and K_5, which both have exactly this order. Thus both H_4 and H_5 have order 51840, but their orbit shapes show that they are not conjugate.

- $H_{10} \lesssim H_2$ or H_6.

We will show directly that H_{10} is conjugate to K_{10}, the maximal $L_2(7):2$. The generators of H_{10} satisfy the presentation:

$$\langle a, b : a^2 = b^3 = (ab)^8 = [a,b]^4 = 1\rangle \simeq L_2(7){:}2 \qquad (4.3.1)$$

and H_{10} has orbits of size 336. Thus H'_{10} is an $L_2(7)$ subgroup of O'N.

The subgroups of O'N isomorphic to $L_2(7)$ are classified in [18] and [22]. There are four conjugacy classes of such:

- one class containing $7A$ elements;
- two classes (conjugate in O'N.2) containing $7B$ elements, having normalizer $L_2(7){:}2$ in O'N, and contained in an $L_3(7)$ of O'N; and
- one class containing $7B$ elements, self-normalizing in O'N but having normalizer in O'N.2 which is maximal therein.

We construct an $L_2(7)$ in O'N by taking the chain:

$$\text{O'N} > L_3(7) > L_2(7){:}2 > L_2(7) \qquad (4.3.2)$$

and find that it has orbit shape:

$$1^1\ 8^1\ 14^2\ 21^7\ 28^5\ 42^{12}\ 56^{39}\ 84^{151}\ 168^{1368} \qquad (4.3.3)$$

But H'_{10} has orbit shape:

$$7^2\ 8^2\ 14^2\ 21^6\ 28^4\ 42^{12}\ 56^{38}\ 84^{152}\ 168^{1368} \qquad (4.3.4)$$

Since these are not the same, H'_{10} is not of the second type. Moreover, $[a, bab]$ is a $7B$ element in H'_{10}, so it is not of the first type either. Thus H_{10} must be the normalizer of the self-normalizing $L_2(7)$; i.e. H_{10} is conjugate to K_{10} in O'N.2.

- $H_9 \lesssim H_3$, H_4 or H_5.

  The subgroup H_9 satisfies the presentation:

  $$\langle a, b : a^2 = b^3 = (ab)^8 = [a, b]^5 = [a, bababab^{-1}]^2 = 1\rangle \simeq PGL_2(9) \tag{4.3.5}$$

  and has an orbit of size 720, so it must be isomorphic to $PGL_2(9)$. Thus H'_9 is an A_6 subgroup of O'N. Such subgroups are classified in [18] and [22]:

  - one class whose normalizer is $(3^2{:}4 \times A_6) \cdot 2$;
  - two classes (conjugate in O'N.2) with normalizer M_{10} contained in M_{11}; and
  - one class which is self-normalizing in O'N and is contained in A_7 but which has normalizer $PGL_2(9)$ in O'N.2.

  We can construct an A_6 of the first type by finding the third commutator subgroup of $H_4 \simeq (3^2{:}4 \times A_6) \cdot 2^2$. Its orbit shape is:

  $$15^{24}\ 60^{36}\ 90^{108}\ 180^{72}\ 360^{612} \tag{4.3.6}$$

  We can construct an A_6 of the second type using the chain

  $$\mathrm{O'N} > \mathrm{M}_{11} > \mathrm{M}_{10} > A_6 \tag{4.3.7}$$

  It has orbit shape:

  $$10^1\ 40^2\ 45^4\ 60^{10}\ 90^{17}\ 120^{22}\ 180^{144}\ 360^{596} \tag{4.3.8}$$

However, H_9' has orbit shape:

$$15^2\ 30^2\ 45^6\ 60^{10}\ 90^{12}\ 120^{22}\ 180^{146}\ 360^{596} \tag{4.3.9}$$

so it must be of the third type. Thus H_9 is conjugate to K_9.

H_4 and H_5 can be distinguished because H_5 is soluble whereas H_4 is not (this can be checked on the faithful orbits of size 360).

4.3.2 HN.2

HN.2 has a permutation representation on 1140000 points. Table 4.11 gives the orbit shapes of the H_i.

For $i > 1$, we consider the action of H_i on the relevant orbit underlined in Table 4.11. For $i \neq 3$, the action on this orbit has order $|K_i|$. For $i = 3$, it has order $|K_i|/2$, but b^{20} is an element of order 2 in the kernel of the action homomorphism. This establishes (A2).

The orbit shapes and outer class argument are sufficient to establish (A3). The H_i can be distinguished by their orders alone.

4.3.3 Fi_{24}

Fi_{24} has a permutation representation on 306936 points. Table 4.12 gives the orbit shapes of the H_i.

For each i, we let H_i act on the underlined orbit in Table 4.12. The computed sizes of the images of these actions, together with the following subgroups of the action kernel:

$$\begin{aligned} H_2 :&\quad \langle (ab^4ab^3)^{13} \rangle &\simeq C_2 \\ H_7 :&\quad \langle (ab^{11}ab)^6 \rangle &\simeq C_3 \\ H_{10} :&\quad \langle (ab)^8 \rangle &\simeq C_2 \end{aligned} \tag{4.3.10}$$

establish that $|K_i|$ divides $|H_i|$ for $2 \leq i \leq 21$.

Group	Orbit shape	Putative structure
H_1	1140000^1	HN
H_2	1^1 $\underline{462^1}$ 5040^1 10395^1 16632^1 30800^1 69300^1 311850^1 332640^1 362880^1	S_{12}
H_3	1100^1 $\underline{7700^1}$ 154000^1 246400^1 268800^1 462000^1	$4 \cdot \mathrm{HS}.2$
H_4	$\underline{3648^1}$ 25536^1 153216^1 344736^1 612864^1	$U_3(8){:}6$
H_5	800^1 $\underline{1920^1}$ 9600^1 57600^1 61440^1 76800^1 102400^1 153600^1 307200^1 368640^1	$2_+^{1+8}.(A_5 \times A_5).2^2$
H_6	50^1 $\underline{5250^1}$ 17500^1 21000^1 25200^1 126000^1 210000^1 315000^1 420000^1	$5{:}4 \times U_3(5){:}2$
H_7	$\underline{2500^1}$ 62500^1 125000^1 200000^1 250000^1 500000^1	$5_+^{1+4}{:}2_-^{1+4}.5.4.2$
H_8	40^1 $\underline{72^1}$ 1920^1 2160^2 2304^1 5120^1 6480^1 11520^1 13824^1 17280^2 20736^1 23040^1 55296^1 69120^3 103680^1 138240^2 165888^1 207360^1	$2^6 \cdot U_4(2).2$
H_9	2^1 $\underline{72^1}$ 400^1 450^1 720^1 1296^1 1440^1 5400^1 8100^1 8640^1 10800^1 14400^3 16200^1 21600^1 32400^3 43200^1 64800^2 86400^1 103680^1 129600^1 172800^1 259200^1	$(S_6 \times S_6){:}2^2$
H_{10}	64^1 $\underline{448^1}$ 672^1 2688^1 10752^3 14336^1 21504^1 43008^1 57344^1 64512^1 86016^2 129024^1 258048^1 344064^1	$2^3.2^2.2^6.(3 \times L_3(2)).2$
H_{11}	$\underline{15000^1}$ 125000^1 250000^1 375000^2	$5^2.5.5^2.4A_5.2$
H_{12}	6^1 $\underline{72^1}$ 576^1 648^2 1458^1 2592^1 3888^1 5184^1 5832^2 7776^4 11664^3 15552^3 20736^1 23328^4 31104^1 46656^3 62208^1 93312^7	$3^4{:}2(S_4 \times S_4).2$
H_{13}	60^1 $\underline{540^1}$ 2160^1 3240^3 4860^1 9720^1 14580^1 19440^3 29160^5 38880^2 58320^8 116640^3	$3_+^{1+4}{:}4S_5$

Table 4.11: Orbit shapes of H_i for HN.2

Group	Orbit shape	Putative structure
H_1	306936^1	Fi_{24}'
H_2	1^1 $\underline{31671^1}$ 275264^1	$\mathrm{Fi}_{23} \times 2$
H_3	2^1 3510^1 $\underline{56320^1}$ 247104^1	$(2 \times 2 \cdot \mathrm{Fi}_{22}){:}2$
H_4	$\underline{3^1\ 3240^1}$ 85293^1 218400^1	$S_3 \times O_8^+(3){:}S_3$
H_5	$\underline{528^1}$ 104448^1 201960^1	$O_{10}^-(2){:}2$
H_6	$\underline{1134^1}$ 30240^1 275562^1	$3^7 \cdot O_7(3){:}2$
H_7	$\underline{1485^1}$ 128304^1 177147^1	$3_+^{1+10}{:}(2 \times U_5(2){:}2)$
H_8	24^1 $\underline{24288^1}$ 282624^1	$2^{12} \cdot \mathrm{M}_{24}$
H_9	3^1 693^1 $\underline{8448^1}$ 76032^1 221760^1	$(2 \times 2^2 \cdot U_6(2)){:}S_3$
H_{10}	$\underline{504^1}$ 48384^1 258048^1	$2_+^{1+12} \cdot 3U_4(3).(2^2)_{122}$
H_{11}	27^1 81^1 $\underline{2106^1}$ 9477^1 39366^1 255879^1	$3^3.[3^{10}].(L_3(3) \times 2^2)$
H_{12}	270^1 $\underline{4860^1}$ 65610^1 236196^1	$3^2.3^4.3^8.(S_5 \times 2S_4)$
H_{13}	6^1 360^1 $\underline{9450^1}$ 11520^1 134400^1 151200^1	$S_4 \times O_8^+(2){:}S_3$
H_{14}	120^1 $\underline{2688^1}$ 107520^1 196608^1	$2^{3+12}.(L_3(2) \times S_6)$
H_{15}	8^1 48^1 $\underline{960^1}$ 8960^1 43008^1 122880^1 131072^1	$2^{7+8}.(S_3 \times A_8)$
H_{16}	6^1 $\underline{2106^1}$ 4368^1 25272^1 117936^1 157248^1	$(S_3 \times S_3 \times G_2(3)){:}2$
H_{17}	10^1 36^1 $\underline{420^1}$ 1200^1 3150^1 5600^1 16800^1 30240^1 37800^1 60480^1 151200^1	$S_5 \times S_9$
H_{18}	15^1 $\underline{216^1}$ 840^1 945^1 2520^1 7560^1 15120^1 30240^1 45360^2 68040^1 90720^1	$S_6 \times L_2(8){:}3$
H_{19}	21^1 $\underline{294^1}$ 441^1 735^1 1470^1 2205^1 4410^3 8820^1 13230^2 15120^1 17640^4 26460^3 35280^1 52920^1	$7{:}6 \times S_7$
H_{20}	$\underline{294^1}$ 2058^2 4116^3 6174^3 12348^{10} 24696^6	$7_+^{1+2}{:}(6 \times S_3).2$
H_{21}	$\underline{406^{36}}$ 812^{360}	$29{:}28$

Table 4.12: Orbit shapes of H_i for Fi_{24}

The outer class argument holds for these groups, and the orbit shapes in Table 4.12 show that no other putative inclusions are possible.

To distinguish H_{14} from H_{15}, note that H_{15} acts on its orbit of size 8 like A_8, and $K_{14} \simeq 2^{3+12}.(L_3(2) \times S_6)$ does not have any quotient isomorphic to A_8 (by the Jordan-Hölder theorem), and so we must have that H_{14} is conjugate to K_{14} and H_{15} is conjugate to K_{15}.

Appendix A

GAP code for the toolkit

A.1 Global variables

A.1.1 User-defined global variables

Before using the toolkit, the following global variables need to be defined:

- `G` is the almost simple group G under consideration.
- `C` and `D` are its standard generators c and d.
- `CT` is the character table of G.
- `omega0` is some subset of the set on which G acts. Ideally, it should be a base of G.

A.1.2 Toolkit global variables

The toolkit itself defines the following global variables:

- `TKatoms` is an array containing the elements cd^i for $1 \leq i < o(d)$.
- `TKatomsinv` contains the inverses of the elements in `TKatoms`.
- `TKpossorders` is a list of the allowable values for $o(g)$, $g \in G$.

They are defined as follows:

```
TKatoms := [C*D];;
for i in [2..Order(D)-1] do
    TKatoms[i] := TKatoms[i-1]*D;
od;
TKatominvs := List(TKatoms, x->x^-1);
TKpossorders := Set(OrdersClassRepresentatives(CT));;
```

A.2 Functions for prewords

A.2.1 `TKevalpreword`

Purpose Evaluate a preword w with the function η.

Parameter `preword` is the preword w to evaluate

Returns The value of $\eta(w)$.

```
TKevalpreword := function(preword)

    local atomnum, elt;
    elt := ();

    for atomnum in preword do
        elt := elt * TKatoms[atomnum];
    od;

    return elt;

end;
```

A.2.2 `TKlistprewords`

Purpose List prewords lexicographically (as per the successor function σ).

Parameters

- `maxlength` is the length of the longest preword to be produced by the function.
- `numatomtypes` is the number of different values that are put into the tuple. Typically, this will be the $o(d) - 1$.
- `startwith` is a tuple (represented by a sequence in GAP) which is the first preword to produce. This will usually be `[1]`.
- `callbackfunction` is a function which takes a preword as an argument and returns either `true` or `false`. This function is called for every preword listed. If the function returns `false`, then `TKlistprewords` aborts the listing.

Returns `true` iff all the prewords were evaluated.

Example To print a list of all prewords of length ≤ 6 when d has order 3:

```
gap> TKlistprewords(6, 2, [1],
>                    function(w)
>                        Print(w, "\n");
>                        return true;
>                    end);
```

```
TKlistprewords := function(maxlength, numatomtypes,
                           startwith, callbackfunction)

    local list, size, i, j, incremented;

    list := startwith;
    size := 1;

    repeat
        if not callbackfunction(list) then
            return false;
```

```
        fi;
        incremented := false;
        size := Size(list);
        for i in [1..size] do
            if (list[i] < numatomtypes) then
                list[i] := list[i] + 1;
                for j in [1..i-1] do
                    list[j] := 1;
                od;
                incremented := true;
                break;
            fi;
        od;
        if not incremented then
            for i in [1..size] do
                list[i] := 1;
            od;
            Add(list,1);
            size := size + 1;
        fi;
    until size = maxlength+1;

    return true;

end;
```

A.2.3 TKprintpreword

Purpose Print the word in c and d associated with a preword w.

Parameter The preword w.

```
TKprintpreword := function(preword)
    local i;

    for i in [1..Size(preword)] do
        if i > 1 then
            Print("*");
        fi;
```

```
        Print("C*D");
        if (preword[i] > 1) then
            Print("^", preword[i]);
        fi;
    od;
end;
```

A.3 Functions for conjugacy classes

A.3.1 `TKmakecycletypetable`

Purpose Compile a table of the cycle types of each conjugacy class in G.

Parameters

- `chartable` is the character table of the group G.
- `permchar` is the permutation character corresponding to the representation of G we are using.

Returns A table of records; one for each conjugacy class of G. The `name` part of the record gives the ATLAS name of the conjugacy class, and the `cycles` part gives the cycle type in the same format as the `CycleStructurePerm` GAP function (*i.e.* the ith element gives the number of $(i+1)$-cycles).

```
TKmakecycletypetable := function(chartable, permchar)

    local i, j, table, numcycles, numfixedpoints,
          orders, numclasses;

    numfixedpoints := ValuesOfClassFunction(permchar);
    orders := OrdersClassRepresentatives(chartable);

    table := [];
```

```
numcycles := function(class, cyclelength, recursedepth)
    local d, fixedpointsaccountedfor, cycles;

    fixedpointsaccountedfor := 0;

    if IsBound(table[class].cycles[cyclelength-1]) then
        return table[class].cycles[cyclelength-1];
    else
        for d in DivisorsInt(cyclelength) do
            if (not d in [1, cyclelength]) then
                fixedpointsaccountedfor :=
                    fixedpointsaccountedfor +
                    d * numcycles(class, d,
                        recursedepth+1);
            fi;
        od;

        cycles := (numfixedpoints[PowerMap(chartable,
                    cyclelength)[class]]
                    - numfixedpoints[class]
                    - fixedpointsaccountedfor) / cyclelength;

        table[class].cycles[cyclelength-1] := cycles;
        return cycles;
    fi;
end;

numclasses := Size(ClassNames(chartable));

for i in [1..numclasses] do
    table[i] := rec(
        name := ClassNames(chartable, "ATLAS")[i],
        cycles := [ ]
    );
od;

for i in [1..numclasses] do
    for j in DivisorsInt(orders[i]) do
        if j > 1 then
            numcycles(i, j, 1);
        fi;
    od;
```

```
    od;

    for i in [1..numclasses] do
        for j in [1..Size(table[i].cycles)] do
            if IsBound(table[i].cycles[j]) and
                table[i].cycles[j] = 0 then
                Unbind(table[i].cycles[j]);
            fi;
        od;
    od;

    return table;

end;
```

A.3.2 TKccl

Purpose Find the conjugacy class of an element of G.

Parameters

- g is the element whose conjugacy class we want to know
- table is the table of cycle types produced by the function TKmakecycletypetable.

Returns An array containing the names of all the possible ATLAS classes to which the element could belong, on the basis of the cycle type of the element. Note that this array may contain more than one element.

```
TKccl := function(g, table)
    local list, row, cycletype;

    list := [];
    cycletype := CycleStructurePerm(g);

    for row in table do
        if row.cycles = cycletype then
            Add(list, row.name);
```

```
        fi;
    od;

    return list;
end;
```

A.3.3 TKfindcclrep

Purpose Find a conjugacy class representative for a given ATLAS class as a word in the standard generators of G.

Parameters

- `atlasname` is the ATLAS name for the class to be sought
- `chartable` is the character table of the group G.
- `cycletable` is the table of cycle types produced with `TKmakecycletypetable`.
- `startwith` is the preword to start with (usually `[1]`).

Returns The conjugacy class representative found (as a permutation). A word in c and d giving this element is output to the screen.

Note Classes are distinguished by their cycle types. By powering up, it is sometimes possible to ensure that an element is in the correct class even if there is more than one class with a given cycle type. If the function cannot find a strategy which guarantees an element in the correct class, then it prints a warning.

```
TKfindcclrep := function(atlasname, chartable, cycletable, startwith)

    local i, j, classnames, targetclassnum,
        orders, targetorder, powerableclasses,
        powerableclassnums, unambiguousclasses,
        searchclasses, atoms, elt, eltccls, ambiguous,
```

```
    gottoend, preword, found;

classnames := ClassNames(chartable, "ATLAS");
orders := OrdersClassRepresentatives(chartable);

targetclassnum := Position(classnames, atlasname);
targetorder := orders[targetclassnum];

unambiguousclasses := [];
powerableclasses := [];
powerableclassnums := [];

# Find classes which power up to the class we want
for i in [1..Size(classnames)] do
    if orders[i] mod targetorder = 0 then
        if PowerMap(chartable, orders[i]/targetorder, i)
            = targetclassnum then

            Add(powerableclassnums, i);
            Add(powerableclasses, classnames[i]);

        fi;
    fi;
od;

# Of these, find a subset B such that if an element
# has the same cycle type as a class in B, then it
# powers up to an element in the class we want.
for i in powerableclassnums do
    if IsSubset(powerableclasses,
        List(Filtered(cycletable,
        x->x.cycles = cycletable[i].cycles),
        x->x.name)) then

        Add(unambiguousclasses, classnames[i]);

    fi;
od;

# If this subset is empty, we will look for the any of the
# 'powerable' classes, and simply accept that the ccl rep
# we find might lie in the wrong class. Otherwise, we
# have a set of classes which are easy to distinguish and
```

```
    # definitely power up to the correct class.
    if IsEmpty(unambiguousclasses) then
        Print("Warning: cannot distinguish class ", atlasname,
            " by cycle type and powering up alone.\n",
            "Representative found may lie in wrong class.\n");
        searchclasses := powerableclasses;
    else
        searchclasses := unambiguousclasses;
    fi;


    # List prewords and until we find an element in one of the
    # search classes
    gottoend := TKlistprewords(1000, Size(TKatoms), startwith,
        function(testpreword)
            local i;

            elt := TKevalpreword(testpreword);
            eltccls := TKccl(elt, cycletable);

            if not IsEmpty(Intersection(eltccls,
                searchclasses)) then

                # Print out the preword prettily
                if not Order(elt) = targetorder then
                    Print("(");
                fi;
                TKprintpreword(testpreword);
                if not Order(elt) = targetorder then
                    Print(")^", Order(elt)/targetorder);
                fi;
                Print("\n");
                return false;

            fi;
            return true;
        end);

    if gottoend then
return fail;
    fi;

    return elt^(Order(elt)/targetorder);
```

```
end;
```

A.4 Functions for large degree permutation groups

A.4.1 TKprintorbitshape

Purpose Print the orbit shape of a subgroup H

Parameter The subgroup H of G

Note This function, unlike the built-in function in GAP, can count fixed points because it knows the size of the set on which G acts.

```
TKprintorbitshape := function ( subgp )
    local orbits, sort, set, t, u, count, fixpts;

    orbits := OrbitLengths( subgp );
    sort := AsSortedList( orbits );
    set := AsSet( sort );

    fixpts := NrMovedPoints(G) - Sum(sort);
    if fixpts > 0 then
        Print("1^", fixpts, " ");
    fi;
    for t in set  do
        count := 0;
        Print( t, "^" );
        for u  in sort  do
            if u = t  then
                count := count + 1;
            fi;
        od;
        Print( count, " " );
    od;
end;
```

A.4.2 TKorderword

Purpose Estimate the order of a word in g and $h^{\eta(w)}$, where $g, h \in G$ and $w \in W$.

Parameters

- `elt1` is the element g
- `elt2` is the element h
- `conjpreword` is the preword w
- `word` is an array representing the word we are calculating. Each element of the array is 1 or 2: 1 represents multiplication by g, 2 represents multiplication by $h^{\eta(w)}$.
- `require` is the order we are looking for, if any. A value of zero for this parameter means we are not hoping for a particular value for the order.

Note Choosing a non-zero value for `require` will generally make the function run faster, as it may abort earlier in the case of an element with the wrong order.

Returns The best estimate of the order of our word in g and $h^{\eta(w)}$, or possibly 0 if `require` is non-zero and the order is definitely not the order we require. If the returned value is non-zero, then it divides the true order.

```
TKorderword := function(elt1, elt2, conjpreword,
                        word, require, pointset)

    local basecur, i, pt, origpt, ptorder, curorder,
        gen, pos, sizeconjpreword, possorders;

    sizeconjpreword := Size(conjpreword);

    possorders := ShallowCopy(TKpossorders);
```

```
    curorder := 1;

    for origpt in pointset do
        pt := origpt;
        ptorder := 0;
        repeat
            for gen in word do
                if gen = 1 then
                    pt := pt^elt1;
                else
                    for pos in Reversed([1..sizeconjpreword]) do
                        pt := pt^(TKatominvs[conjpreword[pos]]);
                    od;
                    pt := pt^elt2;
                    for pos in [1..sizeconjpreword] do
                        pt := pt^(TKatoms[conjpreword[pos]]);
                    od;
                fi;
            od;
            ptorder := ptorder + 1;
            if require > 0 and ptorder > require then
                return 0;
            fi;
        until pt = origpt;
        curorder := Lcm(ptorder, curorder);
        possorders := Filtered(possorders, x->x mod curorder = 0);
        if Size(possorders) = 1 then
            return possorders[1];
        fi;
        if require > 0 and not require in possorders then
            return 0;
        fi;
    od;

    return curorder;

end;
```

A.4.3 `TKcentralizertest`

Purpose Given $w \in W$, $g \in G$ test whether $\eta(w) \in C_G(g) - \{1\}$.

Parameters

- `preword` is the preword w;
- `elt` is the element g;
- `pointset` is a subset of Ω, preferably a base.

Returns Either `true` or `false`. The result will be correct if `pointset` is a base.

```
TKcentralizertest := function(preword, elt, pointset)

    local i, omega, image1, image2,
          mightbeidentity, mightcommute;

    mightbeidentity := true;
    mightcommute := true;

    for omega in pointset do

        image1 := omega^elt;
        image2 := omega;

        for i in preword do
            image1 := image1^TKatoms[i];
            image2 := image2^TKatoms[i];
        od;

        if not image2 = omega then
            mightbeidentity := false;
        fi;

        image2 := image2^elt;

        if not image1 = image2 then
            mightcommute := false;
            break;
        fi;
    od;
```

```
    return (mightcommute and not mightbeidentity);

end;
```

A.4.4 TKcentralizerccltest

Purpose Given $w \in W$, $g, h \in G$, test whether $h^{\eta(w)} \in C_G(g)$.

Parameters

- cclrep is the element h
- preword is the preword w
- elt is the element g
- pointset is a subset of Ω, preferably a base.

Returns Either true or false. The result will be correct if pointset is a base.

```
TKcentralizerccltest := function(cclrep, preword, elt, pointset)

    local i, omega, image1, image2, revpreword, mightcommute;

    mightcommute := true;
    revpreword := Reversed(preword);

    for omega in pointset do

        image1 := omega^elt;
        image2 := omega;

        for i in revpreword do
            image1 := image1^TKatominvs[i];
            image2 := image2^TKatominvs[i];
        od;

        image1 := image1^cclrep;
        image2 := image2^cclrep;
```

```
        for i in preword do
            image1 := image1^TKatoms[i];
            image2 := image2^TKatoms[i];
        od;

        image2 := image2^elt;

        if not image1 = image2 then
            mightcommute := false;
            break;
        fi;
    od;

    return mightcommute;

end;
```

A.4.5 `TKnormalizertest`

Purpose Given $K \leq G$, $w \in W$, test whether $\eta(w) \in N_G(K)$.

Parameters

- `preword` is the preword w
- `subgp` is the subgroup K
- `gens` is a list of elements of G generating K
- `pointset` is a subset of Ω
- `pointorbits` is a list of K-orbits corresponding to the points in `pointset`

Returns Either `true` or `false`. The result `false` always indicates non-membership; this may not apply with the result `true`.

Note Some of the parameters are redundant, but are included in the parameter list to keep the function fast.

```
TKnormalizertest := function(preword, subgp, gens, pointset, pointorbits)

    local i, j, s, image, gen, revpreword, mightnormalize;

    mightnormalize := true;

    revpreword := Reversed(preword);

    s := Size(pointset);

    for gen in gens do
        for j in [1..s] do
            image := pointset[j];

            for i in revpreword do
                image := image^TKatominvs[i];
            od;
            image := image^gen;
            for i in preword do
                image := image^TKatoms[i];
            od;

            if not image in pointorbits[j] then
                mightnormalize := false;
                break;
            fi;
        od;
    od;

    return mightnormalize;

end;
```

A.4.6 TKnormalizerccltest

Purpose Given $K \leq G$, $h \in G$, $w \in W$, test whether $h^{\eta(w)} \in N_G(K)$.

Parameters

- cclrep is the element h;
- cclrepinv is the element h^{-1};

- `preword` is the preword w;
- `subgp` is the subgroup K;
- `gens` is a set of elements of G generating K;
- `pointset` is a subset of Ω
- `pointorbits` is a list of K-orbits corresponding to the points in `pointset`

Returns Either `true` or `false`. The result `false` always indicates non-membership; this may not apply with the result `true`.

Note Some of the parameters are redundant, but are included in the parameter list to keep the function fast.

```
TKnormalizerccltest := function(cclrep, cclrepinv, preword, subgp, gens,
                                pointset, pointorbits)

    local i, j, s, image, gen, revpreword, mightnormalize;

    mightnormalize := true;

    revpreword := Reversed(preword);

    s := Size(pointset);

    for gen in gens do
        for j in [1..s] do
            image := pointset[j];

            for i in revpreword do
                image := image^TKatominvs[i];
            od;
            image := image^cclrepinv;
            for i in preword do
                image := image^TKatoms[i];
            od;
            image := image^gen;
```

```
            for i in revpreword do
                image := image^TKatominvs[i];
            od;
            image := image^cclrep;
            for i in preword do
                image := image^TKatoms[i];
            od;

            if not image in pointorbits[j] then
                mightnormalize := false;
                break;
            fi;
        od;
    od;

    return mightnormalize;

end;
```

A.5 Functions for constructing subgroups

A.5.1 `TKtrawl`

Purpose Perform trawling as described in section 2.4.1.

Parameters

- `fixedgen` is the first generator for all the candidate subgroups
- `targetorbitshapes` is a list of orbit shapes. If the list is not empty, then only groups with orbit shapes in this list will be printed.
- `targetorders` is a list of subgroup orders. If the list is not empty, then only subgroups with orders in this list will be printed.
- `maxlength` is the maximum length of prewords to search for the second generator.

Returns A subgroup with the right properties, if any were found. Words producing this subgroup are printed out.

```
TKtrawl := function(fixedgen, targetorbitshapes,
                    targetorders, maxlength)

    local transorbitshape, orbitshapes, subgp;

    orbitshapes := List(targetorbitshapes, x->SortedList(x));
    transorbitshape := SortedList(OrbitLengths(G));

    TKlistprewords(maxlength, Size(TKatoms), [],
        function(preword)
            local iscandidate, orbitshape;

            subgp := Group(fixedgen, TKevalpreword(preword));
            orbitshape := SortedList(OrbitLengths(subgp));

            if not orbitshape = transorbitshape then
                if not targetorbitshapes = [] then
                    if not SortedList(OrbitLengths(subgp))
                           in orbitshapes then

                         return true;
                    fi;
                fi;
                if not targetorders = [] then
                    if not Order(subgp) in targetorders then
                        return true;
                    fi;
                fi;
                Print("Group(fixedgen, ");
                TKprintpreword(preword);
                Print(");\n");
                Print("Orbit shape: ");
                TKprintorbitshape(subgp);
                Print("\n");
            fi;

            return true;
```

```
        end);
end;
```

A.5.2 TKstdgensearch

Purpose

Parameters

- `rep1` and `rep2` are elements of G in the required conjugacy classes.
- `conditions` is a list of records. Each record has two fields:
  - `prodtype` is a list containing 1s and 2s representing a word in the two generators of the desired subgroup.
  - `order` is the order that this word should have.
- `targetorbitshape` is a list of integers. If it is not an empty list, then only subgroups with this list as orbit shape will be considered.
- `targetorder` is an integer. If it is non-zero, then only subgroups with this as order will be considered.
- `maxlength` is the maximum length of prewords to consider for the word by which the second generator is conjugated.

Returns A subgroup with the right properties, if any were found. Words producing this subgroup are also printed out.

```
TKstdgensearch := function(rep1, rep2, conditions, targetorbitshape,
                           targetorder, maxlength)

    local orbitshape, subgp;
    orbitshape := SortedList(targetorbitshape);
```

```
    TKlistprewords(maxlength, Size(TKatoms), [],
        function(preword)
            local row, order, iscandidate;

            iscandidate := true;
            for row in conditions do
                if not TKorderword(rep1, rep2, preword,
                    row.prodtype, row.order, omega0)
                    = row.order then

                    iscandidate := false;
                    break;
                fi;
            od;

            if iscandidate then
                subgp := Group(rep1, rep2^TKevalpreword(preword));
                if not targetorbitshape = [] then
                    if not SortedList(OrbitLengths(subgp))
                        = orbitshape then

                        return true;
                    fi;
                fi;
                if not targetorder = 0 then
                    if not Order(subgp) = targetorder then
                        return true;
                    fi;
                fi;
                Print("Group(rep1, rep2^(");
                TKprintpreword(preword);
                Print("));\n");
                return false;
            fi;

            return true;
        end);
    return subgp;
end;
```

A.5.3 `TKinvcentelt`

Purpose Calculate an element in $C_G(t)$ for an involution t using Lemma 2.15.

Parameters

- `centelt` is the involution t to be centralized
- `conjelt` is another element u of G

Returns An element x which is a word in t and u which commutes with t. Some information about how x was calculated is output.

```
TKinvcentelt := function(centelt, conjelt)

    local x, y, o, retelt;

    x := centelt;
    y := conjelt;
    o := Order(x*x^y);

    if (o mod 2 = 0) then
        Print("Even case: (x*x^y)^", o/2);
        retelt := (x*x^y)^(o/2);
    else
        Print("Odd case: y*(x*x^y)^", (o-1)/2);
        retelt := y*(x*x^y)^((o-1)/2);
    fi;

    Print(" - order ", Order(retelt), "\n");
    return retelt;

end;
```

List of References

[1] Michael Aschbacher. *3-Transposition groups*, volume 124 of *Cambridge Tracts in Mathematics*. Cambridge University Press, 1997.

[2] John N. Bray. An improved method for generating the centralizer of an involution. *Arch. Math.*, 74(4):241–245, 2000.

[3] John N. Bray, Ibrahim A. I. Suleiman, Peter G. Walsh, and Robert A. Wilson. Generating maximal subgroups of sporadic simple groups. *Comm. Algebra*, 29(3):1325–1337, 2001.

[4] J. H. Conway. A construction for the smallest Fischer group F_{22}. In *Finite groups '72 (Proceedings of the Gainesville Conference, University of Florida)*, pages 27–35. North-Holland, 1973.

[5] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson. *An* ATLAS *of Finite Groups*. Oxford University Press, 1985.

[6] John D. Dixon and Brian Mortimer. *Permutation Groups*, volume 163 of *Graduate Texts in Mathematics*. Springer, 1996.

[7] Bernd Fischer. Finite groups generated by 3-transpositions. I. *Inv. Math.*, 13:232–246, 1971.

[8] The GAP Group, Aachen, St Andrews. *GAP — Groups, Algorithms and Programming Version 4.2*, 1999.

[9] Dieter Held. The simple group related to M_{24}. *J. Algebra*, 13:253–79, 1969.

[10] Donald G. Higman and Charles C. Sims. A simple group of order $44,352,000$. *Math. Zeitschr.*, 105:110–113, 1968.

[11] G. Higman and J. MacKay. On Janko's simple group of order 50,232,969. *Bull. London Math. Soc.*, 1:89–94, 1969.

[12] Christopher Jansen, Klaus Lux, Richard Parker, and Robert Wilson. *An Atlas of Brauer Characters*. Number 11 in LMS Monographs. Oxford Science Publications, 1995.

[13] Jack MacLaughlin. A simple group of order 898,128,000. In Richard Brauer and Chih-Han Sah, editors, *Theory of Finite Groups*, pages 109–111. W. A. Benjamin Inc., 1969.

[14] Simon Nickerson. Straight line programs for some maximal subgroups. Available online at `http://web.mat.bham.ac.uk/S.Nickerson/maxlsubs/`.

[15] S. P. Norton and R. A. Wilson. Maximal subgroups of the Harada-Norton group. *J. Algebra*, 103:362–376, 1986.

[16] Ákos Seress. *Permutation Group Algorithms*. Number 152 in Cambridge Tracts in Mathematics. Cambridge University Press, 2003.

[17] Ibrahim A. I. Suleiman and Robert A. Wilson. Standard generators for J_3. *Experiment. Math.*, 4(1), 1995.

[18] R. A. Wilson. The maximal subgroups of the O'Nan group. *J. Algebra*, 97(2):467–473, December 1985.

[19] R. A. Wilson. Maximal subgroups of sporadic groups. In E. F. Robertson and C. M. Campbell, editors, *Proceedings of groups — St. Andrew's 1985*, pages 352–357, 1986.

[20] Robert Wilson, Peter Walsh, Jonathan Tripp, Ibrahim Suleiman, Stephen Rogers, Richard Parker, Simon Norton, Steve Linton, and John Bray. Atlas of Finite Group Representations. Available online at `http://web.mat.bham.ac.uk/atlas/`.

[21] Robert A. Wilson. Standard generators for sporadic simple groups. *J. Algebra*, 184(2):505–515, 1996.

[22] Satoshi Yoshiara. The maximal subgroups of the sporadic simple group of O'Nan. *J. Fac. Sci. Univ. Tokyo*, 32:105–141, March 1985.