Generating maximal subgroups of sporadic almost-simple groups

Simon Nickerson

11 April 2003

There are 26 sporadic simple groups. 14 of these groups are complete:

$$\begin{array}{ccccc} M_{11} & M_{23} & M_{24} & Co_3 & Co_2 \\ Co_1 & Fi_{23} & Th & J_1 & J_4 \\ Ly & Ru & B & \mathbb{M} & \end{array}$$

The remaining 12 sporadic groups have automorphism group isomorphic to C_2:

$$\begin{array}{ccccc} M_{12} & M_{22} & HS & McL & J_2 \\ Suz & Fi_{22} & Fi'_{24} & He & HN \\ O'N & J_3 & & & \end{array}$$

For these 12 groups, a complete list of the maximal subgroups for both G (the simple group) and $G.2$ (the automorphism group) is known. We seek generators for the maximal subgroups of these groups $G.2$.

(The case $G = J_3$ has already been considered by Suleiman and Wilson in *Experimental Math.* **4** (1995), 11-18.)

R. A. Wilson has defined *standard generators* for a number of finite groups, including the sporadic simple groups and their automorphism groups. Standard generators are defined in terms of conjugacy classes and orders of words in these generators.

Examples:

- Standard generators for $L_2(7)$ are elements a, b where a has order 2, b has order 3 and ab has order 7.

- Standard generators for $McL.2$ are c and d where c is in class $2B$, d is in class $3B$, cd has order 22 and $(cd)^2(cdcdd)^2cdd$ has order 24.

By convention, standard generators for a simple group G are called a and b, and those for $G.2$ are called c and d.

The crucial property of standard generators

If G_1 is isomorphic to G_2, and $(x_1^{(1)}, \ldots x_1^{(m)})$ and $(x_2^{(1)}, \ldots x_2^{(m)})$ are standard generators for G_1 and G_2, then the group homomorphism induced by $x_1^{(i)} \mapsto x_2^{(i)}$ for $1 \leq i \leq m$ is an isomorphism.

In other words:

If you and I both have copies of G, and we both find standard generators for G, then we have the same generators (up to automorphisms of G). Any computation I do in G with my generators, you can do with yours.

Thus finding standard generators allows us to construct an explicit isomorphism between groups which are known to be isomorphic.

For example, let G be a group with standard generators $g_1, \ldots g_n$ defined. If we find generators $h_1, \ldots h_m$ for a (maximal) subgroup H of G, and write each h_i as words in the $\{g_j : 1 \leq j \leq n\}$, then it allows us to find this subgroup easily in any group isomorphic to G.

Definitions of standard generators and representations for many simple and almost-simple groups can be found at:

`http://www.mat.bham.ac.uk/atlas/`

Outline of the main search methods

- Naive random search
- Random search through the right conjugacy classes
- Special methods for involution centralizers
- Stepwise construction from other subgroups

The algebra system GAP was used for most of the programming.

Naive random search

It is sometimes possible to find some or all of the maximal subgroups by searching for generating sets of the form:

$$\langle g, w(c,d) \rangle$$

where $g \in \{c, d\}$ and $w(c,d)$ is a short word in the standard generators.

Random searching in the right conjugacy classes

Assume that M can be 2-generated.

- Determine in which conjugacy classes of G.2 generators for M might live.

- Find representatives t, u for these G.2 classes.

- Search through groups:

  $$\langle t, v = u^{w(c,d)} \rangle$$

  where $w(c, d)$ is a word in c and d as before.

We will usually also impose some condition on words in t and v (e.g. the order of tv, the order of tvv and so on) using our knowledge of M. Thus we can often get standard generators for M (if they are defined).

Example: the maximal subgroups of He.2

$He.2$ is a group of order 8060774400, and has a permutation representation on 2058 points, corresponding to the cosets of its maximal subgroup $S_4(4) : 4$. Its standard generators are $c \in 2B, d \in 6C$ with cd having order 30.

We found generators for the following subgroups by a naive search.

$S_4(4) : 4$	$2^2.L_3(4).D_{12}$	$S_4 \times L_3(2) : 2$
$2^{4+4}.(S_3 \times S_3).2$	$2^{1+6}.L_3(2).2$	$3 \cdot S_7 \times 2$
$(S_5 \times S_5).2$		

This leaves:

$2^2.L_3(4).D_{12}$	$7^2 : 2.L_2(7).2$	$7^{1+2} : (6 \times S_3)$
$7 : 6 \times L_3(2)$	$5^2 : 4S_4$	He

To find $7^2 : 2.L_2(7).2 < He.2$

Standard generators of $L_2(7).2$ are:

$$c \in 2B,\ o(d) = 3,\ o(cd) = 8,\ o(cdcdd) = 4$$

Standard generators of $2.L_2(7).2$ are preimages C, D such that D has order 3. The corresponding conditions for $7^2 : 2.L_2(7).2$ are:

$$e \in 14C,\ f \in 3B,\ o(ef) = 16,\ o(efeff) = 8$$

Found ccl reps:

$$cdcdcd^3 \in 14C; \qquad (cdcddcdd)^8 \in 3B$$

Searching through conjugates of the second word gives subgroup:

$$\langle cdcdcd^3, [(cdcddcdd)^8]^{d^2cd^4cdcd^2cd^3} \rangle$$

which is the maximal $7^2 : 2.L_2(7).2$ by its order and because it contains outer elements.

Generators for the maximal subgroups of $He.2$

$$
\begin{aligned}
S_4(4):4 &= \langle c, (ddcd)^{cd} \rangle \\
2^2.L_3(4).D_{12} &= \langle d^3, dcdcd^5cd^3cdc \rangle \\
S_4 \times L_3(2):2 &= \langle c, d^5cdd(cd)^4cd^2cd^3cd \rangle \\
2^{4+4}.(S_3 \times S_3).2 &= \langle c, (d^4c)^2ddcdcdcdcddc \rangle \\
2^{1+6}.L_3(2).2 &= \langle c, (d^3c)^2(dc)^2d^3c \rangle \\
3^{\cdot}S_7 \times 2 &= \langle cdcddcdc, d \rangle \\
(S_5 \times S_5).2 &= \langle d, (cd^3)^3d(cd^3cd)^2c \rangle \\
2^2.L_3(4).D_{12} &= \langle d^3, dcdcd^5cd^3cdc \rangle \\
7^2:2.L_2(7).2 &= \langle cdcdcd^3, [(cdcddcdd)^8]^{ddcd^4cdcddcd^3} \rangle \\
7^{1+2}:(6 \times S_3) &= \langle cd^3cdcddcd, (cd^3cdcddcd)^{cdcd^{-1}cd^{-1}cdcdcd^{-1}} \rangle \\
7:6 \times L_3(2) &= \langle d^3, [(cddcddcd)^8]^{dcd^4cd^3c} \rangle \\
5^2:4S_4 &= \langle (cdd)^2(cd)^3, [(cddcddcd)^8]^{ddcd^3(cdd)^3c} \rangle \\
He &= \langle [(cd)^2(cddcd)^2]^{14}, [(cdcdcdd)^3]^{cddcdd} \rangle
\end{aligned}
$$

The two methods discussed so far are sufficient to find all maximal subgroups of the groups:

$$M_{12}.2 \quad M_{22}.2 \quad HS.2 \quad McL.2 \quad J_2.2$$
$$Suz.2 \quad Fi_{22}.2 \quad He.2$$

with the exception of:

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

$$U_4(3).2.2 \times S_3 < Fi_{22}.2$$

which cannot be 2-generated (they both have quotients isomorphic to 2^3). For these groups, we search in the usual way for two generators which generate an index 2 subgroup and then search for an element normalizing this subgroup.

The remaining cases are:

$$HN.2 \quad O'N.2 \quad Fi_{24}$$

Issues of group degree

These last three groups have quite large permutation representations.

Group	Degree
$O'N.2$	245520
Fi_{24}	306936
$HN.2$	1140000

Many group algorithms — such as calculating the order of a subgroup — are thus either very time-consuming or very costly in terms of storage space (or both), and need to be avoided as much as possible.

Example: testing the order of an element

Suppose we wish to test whether

$$cdcd^2cd^3cd^4cd^3cd^2cdcd^2cd^3cd^4 \in Fi_{24}$$

has order 3.

- Naive way - 1.3s

```
gap> Order(C*D*C*D^2*C*D^3*C*D^4*C*D^3*C*D^2*
           C*D*C*D^2*C*D^3*C*D^4);
```

- Use precomputed cd^j - 430ms

```
gap> Order(CD*CD2*CD3*CD4*CD3*CD2*CD*CD2*CD3*CD4);
```

- Calculate order from point images - 100μs

```
gap> eltorderwd([1,2,3,4,3,2,1,2,3,4], 0, f242gpinfo);
```

- Test order from point images - 10μs

```
gap> eltorderwd([1,2,3,4,3,2,1,2,3,4], 3, f242gpinfo);
```

eltorderwd :=

```
function ( wordlist, require, gpinfo )
    local  basecur, i, pt, origpt, ptorder, curorder,
         wd, pos, sizewordlist, possorders;
    sizewordlist := Size( wordlist );
    possorders := ShallowCopy( gpinfo.possorders );
    curorder := 1;
    for origpt  in gpinfo.base  do
        pt := origpt;
        ptorder := 0;
        repeat
            for wd  in wordlist  do
                pt := pt ^ gpinfo.atoms[wd];
            od;
            ptorder := ptorder + 1;
            if require > 0 and ptorder > require  then
                return 0;
            fi;
        until pt = origpt;
        curorder := Lcm( ptorder, curorder );
        possorders := Filtered( possorders, function ( x )
                return x mod curorder = 0;
            end );
        if Size( possorders ) = 1  then
            return possorders[1];
        fi;
        if require > 0 and not require in possorders  then
            return 0;
        fi;
    od;
    return curorder;
```

Special methods for involution centralizers

Let $x \in G$ with $o(x) = 2$. The following procedure gives us an element $z \in C_{G.2}(x)$. 3 or 4 iterations usually suffice to give a generating set. Let $g \in G.2$ be arbitrary, and let $n := o(xx^g)$.

- If $n = 2m$, then $H := \langle x, x^g \rangle \cong D_{4m}$ and $z := (xx^g)^m$ is a central element of H of order 2.

- If $n = 2m + 1$, we let $z := g(xx^g)^m$. Then:

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

An example of ad hoc methods

$HN.2$ has a maximal subgroup

$$M = (S_6 \times S_6) : 2^2$$

- Find standard generators for $S_{12} < HN.2$.
- Find $H = S_6 \wr S_2 \cong (S_6 \times S_6) : 2 < S_{12}$.
- Find an involution commuting with the wreathing $2A$-element of H and normalizing $S_6 \times S_6$. (Look inside $C_{HN.2}(2A)$)

This gives us 3 generators for M. We cannot do better than this because

$$M/(A_6 \times A_6) \cong 2 \times D_8$$

and $2 \times D_8$ cannot be 2-generated.

The last hurdle

The final group to be considered, Fi_{24}, is a very large group:

$$|Fi_{24}| = 251041141838132344258560 0.$$

Many of its conjugacy classes are very large, making many random searches rather impractical.