Coset enumeration, permutation group algorithms, and applications to graphs and geometries

Leonard H. Soicher
School of Mathematical Sciences
Queen Mary and Westfield College
Mile End Road, London E1 4NS, U.K.

In these notes we discuss coset enumeration and basic permutation group algorithms. To illustrate some applications to graphs and finite geometries, we classify and study some graphs which are locally the incidence graph of the $2-(11,5,2)$ design.

1 Coset enumeration

Coset enumeration is one of the oldest and most useful methods of Computational Group Theory (see [32, 7, 14, 20, 13, 27]). Indeed, coset enumeration was first described in 1936 in a paper of Todd and Coxeter [32], and is often called the Todd-Coxeter method.

In these notes, we concentrate on what coset enumeration does, and how to apply coset enumeration. Coset enumeration and related algorithms for finitely presented groups are available in the group theory system GAP [23], and in the algebra system Magma [2].

Let

$$G = \langle X \mid R \rangle$$

be a finitely presented group; that is, G is generated by the finite set X, subject (only) to the finite set R of relators, which are words in $X \cup X^{-1}$ which evaluate to the identity in G ($X^{-1} = \{x^{-1} \mid x \in X\}$). Suppose also we are given a set Y of words in $X \cup X^{-1}$, and let $H = \langle Y \rangle$.

The input to coset enumeration is X, R, Y.

The coset enumeration process then attempts to construct a set Ω, with $1 \in \Omega$, and a transitive permutation representation

$$\rho : G \to \mathrm{Sym}(\Omega),$$

such that for this representation, H is the stabilizer in G of the point 1. Coset enumeration does this by using a trial and error process for constructing the permutations

$$\rho(X \cup X^{-1}).$$

The name coset enumeration comes from the fact that, if a coset enumeration is successful, there is a 1-1 correspondence between the elements of Ω and the cosets of H in G, with 1 corresponding to H.

We start the coset enumeration process with $\Omega = \{1\}$. During the "trial and error process" of coset enumeration new elements of Ω are defined as images of previously defined elements of Ω under the action of members of $X \cup X^{-1}$. The process must ensure that for every α defined to be in Ω, including $\alpha = 1$, the following holds after a finite number of steps:

1. α^x is defined for all $x \in X \cup X^{-1}$.
2. If $\alpha = 1$ then $\alpha^y = \alpha$ for all $y \in Y$.
3. $\alpha^r = \alpha$ for all $r \in R \cup \{xx^{-1}, x^{-1}x \mid x \in X\}$.

When applying conditions (2) and (3), it may be found that two previously defined elements of Ω represent the same coset of H in G. This is called a *coincidence*, and correct coincidence processing is usually the most delicate part of any coset enumeration program (see [20] for details). In practice, many more elements of Ω may be defined during the coset enumeration process than the final number of elements in this set, and even when G is the trivial group a coset enumeration may run out of store before it can finish.

For more complete and precise details of the coset enumeration process, the reader is urged to consult the excellent survey article by Neubüser [20].

If the index of H in G is infinite, then the coset enumeration process will not terminate. If this index is finite then the process will terminate, but no general bounds can be given on the time or store required for termination (this is due to the algorithmic unsolvability of certain questions about finitely presented groups).

There is an enormous amount of flexibility in the coset enumeration process, and many different approaches have been suggested and experimented with (see [7, 20]). Depending on the presentation and the method used there can be huge variations in the time and store taken by a coset enumeration. At present, the most advanced methods are due to George Havas, and these methods are available in the MAGMA [2] system.

Retaining the notation above, we observe that a successfully completed coset enumeration tells us that the index of H in G is finite of size $|\Omega|$. We also have permutation generators $\rho(X)$ for the (transitive) permutation group $\rho(G)$, and we know words in these generators for the stabilizer $\rho(H)$ of the point 1.

We remark that there are many variations on coset enumeration. For example, a so-called modified Todd-Coxeter gives us a presentation for H (see [20]). Also, Steve Linton has generalized the ordinary coset enumeration process to the enumeration of double cosets [17], and the calculation of linear representations of finitely presented algebras [18].

1.1 Applications of coset enumeration

Ordinary coset enumeration has many and varied applications in both group theory and geometry. For example, it has been used by many to study and construct group presentations (see [13, 28, 9]), to construct and characterize certain groups (including certain sporadic simple groups) (see [10, 19, 29]), and to classify certain finite geometries and graphs (see [1, 30, 33]).

We shall give some of the flavour of these applications in the extended example, with exercises, below.

Let Γ and Δ be simple graphs (i.e. undirected, with no loops and no multiple edges). Then Γ is said to be *locally* Δ if for every vertex v of Γ, the induced subgraph on the neighbours of v is isomorphic to Δ.

We shall use coset enumeration to study some presentations which arise in the classification of the connected, ordered-triangle-transitive graphs which are locally the incidence graph of the (unique) $2-(11,5,2)$ design. In the process, we obtain a presentation and a flag-transitive diagram geometry for $M_{12}{:}2$, the automorphism group of the sporadic simple Mathieu group M_{12}.

Let Δ be the incidence graph of the unique $2-(11,5,2)$ design $\mathcal{D}$ (see [12]). Thus Δ has exactly 22 vertices: 11 corresponding to the points of $\mathcal{D}$ and 11 corresponding to the blocks of $\mathcal{D}$, with $\{v,w\}$ an edge of Δ precisely when $\{v,w\}$ is an incident point-block pair.

Let $H = \mathrm{Aut}(\Delta)$. Then $H \cong L_2(11){:}2$ (we use ATLAS notation [8] for group structures), and we can show that H is a homomorphic image of the group

$$\tilde{H} = \langle a,b,c,d \mid a^3, b^2, c^2, d^2, (ab)^3, a(cd)^4, (bc)^3, (bd)^2 \rangle.$$

Indeed, suppose $t, \{t,u\}, \{t,u,v,w\}$ is an incident vertex-edge-quadrangle triple in Δ (H is transitive on such triples), with respective H-stabilizers T, U, V. Then we have a homomorphism of $\tilde{H}$ onto H in which $\langle a,b,c\rangle$ maps onto $T \cong A_5$, $\langle a,b,d\rangle$ maps onto $U \cong S_4$, and $\langle a,c,d\rangle$ maps onto $V \cong D_{24}$.

Now the presentation for $\tilde{H}$ tells us that c, d are elements of order (at most)

2, whose product has order (at most) 12 (since $a^3 = 1$, $a^{-1} = (cd)^4$). Thus $\langle a, c, d\rangle$ is visibly (a quotient of) the dihedral group D_{24}, of order 24, in which the relations $1 = (ac)^2 = (ad)^2$ hold.

Exercise 1 *Use coset enumeration to determine the index of $\langle a, c, d\rangle$ in $\tilde{H}$. Deduce that $\tilde{H} \cong H$.*

Exercise 2 *Find an explicit homomorphism from*

$$A = \langle a, b, c \mid a^3, b^2, c^2, (ab)^3, (ac)^2, (bc)^3\rangle$$

onto the alternating group A_5. Prove that $A \cong A_5$. Show that $\langle a, b, c\rangle$ has index 22 in $\tilde{H}$, and again deduce that $\tilde{H} \cong H$.

If the connected graph Γ is locally Δ, and $\mathrm{Aut}(\Gamma)$ is transitive on the ordered-triangles of Γ, then it can be shown that the stabilizer in $\mathrm{Aut}(\Gamma)$ of a vertex v of Γ acts faithfully on the neighbourhood of v, and is isomorphic either to $L_2(11){:}\,2$ or its Sylow-11 normalizer $11{:}\,10$.

We now consider the case where Γ is a connected, ordered-triangle-transitive, locally Δ graph, with vertex stabilizer $L_2(11){:}\,2$ in $\mathrm{Aut}(\Gamma)$.

Suppose such a Γ exists, and let $G = \mathrm{Aut}(\Gamma)$. Let $\{x, y, z\}$ be a triangle of Γ, and X, Y, Z be the G-stabilizers of $x, \{x, y\}, \{x, y, z\}$, respectively. Analysis of Δ and $L_2(11){:}\,2$ shows that X, Y, Z must be respectively isomorphic to

$$L_2(11){:}\,2, \quad S_5, \quad (A_4 \times 3){:}\,2.$$

Furthermore, we must have $X \cap Y \cong A_5$, $X \cap Z \cong S_4$, and $Y \cap Z \cong S_4$. Given this information, we can then show that G is a homomorphic image of the group

$$\tilde{G} =$$

$$\langle a, b, c, d, e \mid a^3, b^2, c^2, d^2, e^2, (ab)^3, a(cd)^4, (ae)^2, (bc)^3, (bd)^2, (be)^2, (ce)^2, (de)^3\rangle, \tag{1}$$

such that $\tilde{X} = \langle a, b, c, d\rangle$ maps bijectively to X, $\tilde{Y} = \langle a, b, c, e\rangle$ maps bijectively to Y, and $\tilde{Z} = \langle a, b, d, e\rangle$ maps bijectively to Z.

We shall classify the graphs Γ by first classifying the homomorphisms of $\tilde{G}$, such that these homomorphisms are isomorphisms when restricted to each of $\tilde{X}, \tilde{Y}, \tilde{Z}$. An image G of such a homomorphism is a candidate automorphism group for one of the graphs Γ we seek, such that the G-images X, Y of $\tilde{X}$,$\tilde{Y}$ should be the respective stablizers of a vertex and an edge on that vertex in Γ. This determines Γ completely, and Γ can be constructed and studied using G, X and Y (see Section 2.5).

Exercise 3 *Apply coset enumeration, and find that $\langle a, b, c, d\rangle \cong L_2(11){:}\,2$ has index 432 in $\tilde{G}$. What is the order of $\tilde{G}$?*

Exercise 4 *Use coset enumeration to show that $\langle (bcde)^{11}\rangle$ is a normal subgroup of order 3 of $\tilde{G}$.*

From the information in [22], we see that $M_{12}{:}\,2$ acts transitively on the ordered-triangles of a connected, locally Δ graph Γ_{144}, having 144 vertices, with vertex stabilizer $L_2(11){:}\,2$. It follows that $\mathrm{Aut}(\Gamma_{144}) \cong M_{12}{:}\,2$,

$$\tilde{G}/\langle (bcde)^{11}\rangle \cong M_{12}{:}\,2,$$

and

$$\tilde{G} \cong (3 \times M_{12}){:}\,2$$

(note that $\tilde{G}$ has the symmetric group S_3 as a homomorphic image).

Now $(3 \times M_{12}){:}\,2$ is also the automorphism group of a connected, ordered-triangle-transitive, locally Δ graph, Γ_{432} say, having 432 vertices and vertex stabilizer $L_2(11){:}\,2$ (see Exercise 7). We conclude that Γ_{144} and Γ_{432} are the only connected, ordered-triangle-transitive, locally Δ graphs whose vertex stabilizer is $L_2(11){:}\,2$.

Note that we obtain a presentation for $M_{12}{:}\,2$ from the presentation (1) by adjoining the relator $(bcde)^{11}$.

We obtain diagram geometries (see [5]) from the above graphs by applying a theorem of A. Neumaier. It follows from [21, Theorem 7] that Γ_{144} and Γ_{432} give rise to residually connected, flag-transitive geometries with diagram

for $M_{12}{:}\,2$ and $(3 \times M_{12}){:}\,2$, respectively. The objects (from left to right in the diagram) are the vertices, edges, triangles, and octahedra of the corresponding graph.

2 Permutation group algorithms

Permutation group algorithms is the most developed area of Computational Group Theory, and is still being actively developed (see [25, 26, 15, 16, 11]). In this section, we describe some of the basic permutation group algorithms, and indicate the capabilities of more advanced algorithms.

Throughout this section, G is a permutation group, with generating set $X = \{X_1, \ldots, X_d\}$, acting on a finite set Ω of size n.

In most practical implementations, the set Ω is taken to be $\{1, \ldots, n\}$. In a language like PASCAL, a permutation a is usually stored as an array of integers of length n, with a[i] storing the image i^a of i under a. Then given two permutations a, b, the image i^{ab} of i under the product ab is simply b[a[i]]. Thus, we can determine $c = ab$ in PASCAL as follows:

```
for i:=1 to n do c[i] := b[a[i]];
```

In the GAP and MAGMA systems, objects such as permutations and matrices are basic data types, and we can compute a product simply by writing

```
c:=a*b;
```

Also, in these languages, the image i^a is just i^a.

2.1 Orbits and Schreier vectors

The most basic structural information about the permutation group $G = < X >$ on Ω are the orbits $\{\alpha^G \mid \alpha \in \Omega\}$ of G on Ω. These can be computed in $O(n|X|)$ time, as shown below.

The calculation of these orbits consists of repeated calculation of single orbits until all points of Ω have been encountered. The computation of a single orbit $\alpha^G = \{\alpha^g \mid g \in G\}$ essentially consists of a breadth-first-search to compute a (directed) spanning tree, rooted at α, of the (strongly) connected component containing α in the directed graph with vertex-set Ω, and edge-set $\{(\beta, \gamma) \mid \beta^x = \gamma \text{ for some } x \in X\}$. These spanning trees are described by a Schreier vector, usually implemented as an array.

A *Schreier vector*, *sch*, for a set Λ of orbits of G on Ω, with respect to the generating sequence $[X_1, \ldots, X_d]$ for G, is a map

$$sch : \Omega \to \{-1, 0, \ldots, d\},$$

with the property that $sch[\gamma] = -1$ if γ is not in an element of Λ, $sch[\gamma] = 0$ if γ is the root of the spanning tree for an orbit in Λ, and otherwise $sch[\gamma] = i > 0$ if (β, γ) is an edge of the spanning tree for an orbit in Λ, such that $\beta^{X_i} = \gamma$. Usually, Λ consists either of a single orbit or all orbits of G on Ω.

We give below straightforward GAP code to determine all the orbits of the permutation group generated by $X = \{X_1, \ldots, X_d\}$, acting on $\{1, \ldots, n\}$. Upon exit, *sch* will be a Schreier vector for these orbits (w.r.t. the generating sequence $[X_1, \ldots, X_d]$), and the orbits will be specified by the array *orbnum*, indexed by the points of Ω, such that $orbnum[\alpha] = i$ will mean that point α is in the i-th orbit.

```
#
# Computing the orbits of  <X>  on  {1,...,n},
# and a Schreier vector  sch.
#
norb:=0; # number of orbits found so far
for alpha in [1..n] do # initialize
 sch[alpha]:=-1;
od;
for alpha in [1..n] do
 if sch[alpha]=-1 then
  # new orbit
  norb:=norb+1;
  orbnum[alpha]:=norb;
  sch[alpha]:=0;
  orb:=[alpha]; # orbit of alpha so far
  for beta in orb do
   for i in [1..Length(X)] do
    gamma:=beta^X[i];
    if sch[gamma]=-1 then
     # new point in orbit of alpha
     orbnum[gamma]:=norb;
     sch[gamma]:=i;
     Add(orb,gamma); # add gamma to orbit of alpha
    fi;
   od;
  od;
 fi;
od;
```

Given a Schreier vector *sch*, for certain orbits of G on Ω, and a point β in one of these orbits, we can determine a word $w(\beta)$ in X, such that $\alpha^{w(\beta)} = \beta$, where α is the orbit representative (root) for the orbit (spanning tree) containing β. This calculation proceeds as follows:

Set $i := sch[\beta]$. If $i = 0$ then return the empty word as $w(\beta)$. Otherwise apply this algorithm (recursively) to obtain $W := w(\beta^{X_i^{-1}})$, and then return WX_i as $w(\beta)$.

2.2 Point stabilizers

It is often important to know the stabilizer G_α in G of a point $\alpha \in \Omega$. Given a Schreier vector for the (single) orbit α^G, we have shown how to compute, for each $\beta \in \alpha^G$, an element $g(\beta) \in G$ such that $\alpha^{g(\beta)} = \beta$ ($g(\beta)$ is just the word

$w(\beta)$ evaluated on X). We note that $g(\alpha) = 1$, and that for each $t \in G$, we have $tg(\alpha^t)^{-1} \in G_\alpha$. A famous result of Schreier [13, p.18,Lemma 3] implies the following result.

Lemma 1 *The group G_α is generated by*

$$\{g(\beta)xg(\beta^x)^{-1} \mid \beta \in \alpha^G, x \in X\}, \tag{2}$$

where X generates G.

The elements of (2) are called Schreier generators, and usually a small subset of the Schreier generators suffice to generate G_α.

2.3 Bases and strong generating sets

C. Sims [25, 26] introduced the fundamental concept of a base and strong generating set for computing with the permutation group G on Ω. A base and strong generating set is required by most advanced permutation group algorithms.

A sequence $B = (\beta_1, \ldots, \beta_m)$ of points of Ω is called a *base* for G if the (pointwise) stabilizer G_B of B is the trivial group. Note that $g \in G$ is uniquely determined by the base image B^g. A base for G can often be of much smaller size than Ω, and so a base can give us a very compact way to represent elements of G.

A base B defines a chain

$$G = G^{(1)} \geq G^{(2)} \geq \cdots \geq G^{(m)} \geq G^{(m+1)} = 1$$

of subgroups of G, such that $G^{(i)}$ is the (pointwise) stabilizer of $(\beta_1, \ldots, \beta_{i-1})$. We observe that

$$|G| = \prod_{i=1}^{m} |G^{(i)} : G^{(i+1)}|,$$

and that

$$|G^{(i)} : G^{(i+1)}| = |\beta_i^{G^{(i)}}|.$$

A *strong generating set* (SGS) for G with respect to B is a generating set S for G with the property that

$$\langle S \cap G^{(i)} \rangle = G^{(i)}, \quad \text{for } 1 \leq i \leq m+1.$$

Given an SGS, the orbits $\beta_i^{G^{(i)}}$ can be easily computed, and we can thus obtain the order of G. Also note that $(\beta_i, \ldots, \beta_m)$ is a base for $G^{(i)}$, with associated strong generating set $S \cap G^{(i)}$.

Exercise 5 *Let A_5 be the alternating group on $\{1,2,3,4,5\}$. Show (by hand) that $(5,4,3)$ is a base for A_5, and $\{(123),(12)(34),(12)(45)\}$ is an SGS with respect to this base.*

Exercise 6 *Explain why in its action on the incidence graph of the $2-(11,5,2)$ design, the group*

$$L_2(11){:}\,2 \cong \langle a,b,c,d \mid a^3, b^2, c^2, d^2, (ab)^3, a(cd)^4, (bc)^3, (bd)^2 \rangle$$

has as a base a vertex, together with any three of its neighbours. Show that $\{a,b,c,d\}$ is a strong generating set with respect to some such base.

Suppose we are given a permutation p of Ω, and we wish to determine if $p \in G$. There is an efficient algorithm to do this, given a base $B = (\beta_1, \ldots, \beta_m)$ and associated SGS S for G, as above, together with the orbits $\beta_i^{G^{(i)}}$ and their corresponding individual Schreier vectors. Indeed, if $p \in G$, this algorithm determines a factorization

$$g = r_m r_{m-1} \cdots r_1,$$

where r_i is a coset representative of $G^{(i+1)}$ in $G^{(i)}$. This algorithm, called "stripping" or "sifting" p, proceeds as follows:

1. If p is the identity permutation then return the empty factorization for $p \in G$. On the other hand, if G is trivial and p is not the identity, then return that $p \notin G$.

2. If $\beta_1^p \notin \beta_1^G$, then return that $p \notin G$. Otherwise, use the Schreier vector for the orbit β_1^G to determine $r_1 \in G$ such that $\beta_1^{r_1} = \beta_1^p$. Set $q := pr_1^{-1}$ (so q fixes β_1). We then apply the algorithm recursively to the permutation q and the group $G^{(2)}$. If $q \notin G^{(2)}$ then return that $p \notin G$. Otherwise we have a factorization $r_m r_{m-1} \cdots r_2$, say, for $q \in G^{(2)}$, and we return the factorization $r_m r_{m-1} \cdots r_1$ for $p \in G$.

2.4 The Schreier-Sims algorithm

C. Sims [25] devised an algorithm, now called the Schreier-Sims algorithm, to construct a base and associated SGS for the permutation group $G = \langle X \rangle$. His main theoretical tool is Lemma 1 above, but he avoids the inclusion of many redundant Schreier generators in a strong generating set.

The Schreier-Sims algorithm takes as input a partial base $B = (\beta_1, \ldots, \beta_k)$, and a partial strong generating set S, such that $G = \langle S \rangle$, and no element of S fixes each element of B. The algorithm then attempts to verify that B is a base for

G, and S an SGS with respect to B. If this is not the case, then the algorithm adds points to B and group elements to S, as necessary, so that on termination, B is a base for G, and S an SGS with repsect to B.

We give below an outline of a simplified version of the Schreier-Sims algorithm:

1. If $S = \{\}$, then return B, S.

2. At this point we have a non-empty partial base $B = (\beta_1, \ldots, \beta_k)$, say, and partial SGS S, such that $G = \langle S \rangle$, and no element of S fixes each element of B. Set $C := (\beta_2, \ldots \beta_k)$, $T := S \cap G_{\beta_1}$, and apply this algorithm (recursively) with input C, T, so that they are modified to be a base $C = (\beta_2, \ldots, \beta_k, \ldots, \beta_l)$, say, and associated strong generating set T, for $H = \langle T \rangle$.

3. Set $B := (\beta_1, \beta_2, \ldots, \beta_l)$ and $S := S \cup T$. Now we can do membership testing in $H \leq G_{\beta_1}$, using the "sifting" algorithm above, and we test each Schreier generator s for G_{β_1} to see if $s \in H$. If all such Schreier generators are in H then we are done, and return B, S.

4. Otherwise we have a Schreier generator $s \in G_{\beta_1}$, but $s \notin H$. If s fixes all points of B, we append to B a point of Ω which is moved by s. We then set $S := S \cup \{s\}$, and goto Step 2.

In practice, a good implementation of the Schreier-Sims algorithm can be used to compute bases and strong generating sets for permutation groups of degree up to about 10,000. There are many variations and improvements to the basic Schreier-Sims algorithm, which can extend this range (see [15, 11]).

One very interesting aspect of the Schreier-Sims algorithm is that it takes polynomial-time in the size of the input (see [6]). Much work has recently been done on polynomial-time algorithms for permutation groups (see the articles in [11]). One important result is that a composition series for G can be computed in polynomial-time.

Some permutation group algorithms, such as determining the G-stabilizer of a set of points of Ω and determining the centralizer of a subgroup H of G, use at present a backtrack search (of base images of G), and so are not polynomial-time algorithms (but it would be extremely intersesting if polynomial-time algorithms could be found for these tasks). These backtrack algorithms can, however, be quite efficient in practice, even when Ω has size 100,000 or more (see [16]).

2.5 Permutation groups and graphs

As well as his fundamental work on permutation group algorithms, C. Sims [24] also pioneered the powerful graph-theoretical approach to the study of permutation groups. On the other hand, permutation groups can be used to study graphs, and the GRAPE [31] share library package which comes with GAP exploits the use of groups to study graphs, especially those related to permutation groups and finite geometries.

In this section we concentrate on transitive permutation groups, and the study of the related vertex-transitive graphs on which they act (as groups of automorphisms). For geometers, these graphs include the collinearity graphs of point-transitive geometries.

Let G be a transitive permutation group on a finite set Ω. Then G has a natural action on $\Omega \times \Omega$, defined by

$$(\alpha, \beta)^g = (\alpha^g, \beta^g) \qquad (\alpha, \beta \in \Omega, g \in G).$$

The orbits of G on $\Omega \times \Omega$ are called *orbitals*, and the orbits of the stabilizer G_α of a point $\alpha \in \Omega$ are called *suborbits*. It is well-known that the orbitals for G are in 1-to-1 correspondence with these suborbits: this correspondence maps an orbital E to the suborbit $\{\beta \mid (\alpha, \beta) \in E\}$. The *orbital digraph* for G associated with an orbital E is simply the directed graph (digraph) (Ω, E), having vertex-set Ω and edge-set E.

Let $\alpha \in \Omega$, and suppose that $\Omega_1 = \{\alpha\}, \Omega_2, \ldots, \Omega_r$ is an ordering of the orbits of G_α, with respective representatives $\alpha_1 = \alpha, \alpha_2, \ldots, \alpha_r$. Let $\Gamma = (\Omega, E)$ be a digraph on which G acts as a vertex-transitive group of automorphisms, and so E is a union of orbitals, and the neighbourhood of α is the union Λ of the corresponding suborbits contained in $\{\Omega_1, \ldots, \Omega_r\}$. For $i, j = 1, \ldots, r$ define

$$a_{ij} = |\{(\alpha_i, \beta) \in E \mid \beta \in \Omega_j\}|.$$

Note that a_{ij} does not depend on the choice α_i of suborbit representative, and that a_{ij} can be calculated by

$$a_{ij} = |\Lambda^{g(\alpha_i)} \cap \Omega_j\}|,$$

where $g(\alpha_i)$ is an element of G taking α to α_i. The $r \times r$ integer matrix $A = (a_{ij})$ is called the *collapsed adjacency matrix* for Γ (with respect to G and the ordering of the suborbits).

Many properties of a vertex-transitive (di)graph Γ can be easily deduced from a collapsed adjacency matrix for Γ. For example, the collapsed adjacency matrix can be used to determine if Γ is (strongly) connected, and if so, its diameter; if Γ is simple, and if so its girth; whether Γ is distance-regular, and whether G acts distance-transitively on Γ. See [3] for the definitions of these properties, and [22] for more details.

Exercise 7 *Using the presentation (1) and coset enumeration, calculate generators for the degree 432 permutation group* $\tilde{G} \cong (3 \times M_{12}){:}\,2$ *acting on the* *cosets of $H = \langle a,b,c,d\rangle$. Calculate the orbits of H, and the collapsed adjacency matrix for the orbital digraph Γ whose edge-set is $(\alpha,\alpha^e)^{\tilde{G}}$, where $\tilde{G}_\alpha = H$. (You might try to use some appropriate* GRAPE *functions.) Use this matrix to show that Γ is locally the incidence graph of the $2-(11,5,2)$, and that Γ has diameter 4.*

Exercise 8 (Challenging) *As usual, let Δ be the incidence graph of the $2-(11,5,2)$. It can be shown that if G is the automorphism group of a connected, ordered-triangle-transitive, locally Δ graph Γ, such that the G-stabilizer of a vertex is $11{:}10$, then G is a quotient of*

$$\tilde{G} = \langle a,b,c \mid a^5, b^2, c^2, (ac)^2, (bc)^3, [a,b]^{11}, [b,a]^a[a,b]^3\rangle,$$

such that for some triangle $\{x,y,z\}$ of Γ, the respective G-stabilizers of $x, \{x,y\}, \{x,y,z\}$ are the images of $\langle a,b\rangle$, $\langle a,c\rangle$, $\langle b,c\rangle$. (It can be shown that replacing the relator $(ac)^2$ by $[a,c]$ in the presentation above does not lead to such a graph.)

Study the permutation group $\tilde{G}$ acting on the cosets of $\langle a,b\rangle$, using the permutation group algorithms in GAP *or* MAGMA. *Determine the order and structure of $\tilde{G}$.*

Prove that $\tilde{G}$ is the automorphism group of an ordered-triangle-transitive, connected, locally Δ graph, such that the $\tilde{G}$-stabilizer of a vertex is $11{:}10$. Prove that no proper quotient of $\tilde{G}$ is the automorphism group of such a graph.

References

[1] J. van Bon, Some extended generalized hexagons, in *Finite Geometry and Combinatorics* (F. De Clerck *et al.*, eds.), L.M.S. Lecture Notes **191**, Cambridge University Press, 1993, pp. 395–403.

[2] W. Bosma, J. Cannon and G. Matthews, Programming with algebraic structures: the design of the Magma language, in *Proceedings of ISSAC '94*, Assoc. Comp. Mach., 1994, 52–57.

[3] A.E. Brouwer, A.M. Cohen and A. Neumaier, *Distance-Regular Graphs*, Springer, Berlin and New York, 1989.

[4] A.E. Brouwer, D.G. Fon-Der-Flaass and S.V. Shpectorov, Locally co-Heawood graphs, in *Finite Geometry and Combinatorics* (F. De Clerck *et al.*, eds.), L.M.S. Lecture Notes **191**, Cambridge University Press, 1993, pp. 59–68.

[5] F. Buekenhout, Diagram geometries for sporadic groups, in *Finite Groups – Coming of Age* (J. McKay ed.), A.M.S. Contemporary Mathematics **45**, 1985, pp. 1–32.

[6] G. Butler, *Fundamental Algorithms for Permutation Groups*, Lecture Notes in Computer Science **559**, Springer, Berlin and New York, 1991.

[7] J.J. Cannon, L.A. Dimino, G. Havas and J.M. Watson, Implementation and analysis of the Todd-Coxeter algorithm, *Math. Comp.* **27** (1973), 463–490.

[8] J.H. Conway, R.T. Curtis, S.P. Norton, R.A. Parker and R.A. Wilson, *An ATLAS of Finite Groups*, Clarendon Press, Oxford, 1985.

[9] J.H. Conway, S.P. Norton, and L.H. Soicher, The Bimonster, the group Y_{555}, and the projective plane of order 3, in *Computers in Algebra* (M.C. Tangora, ed.), Marcel Dekker, New York, 1988, pp. 27–50.

[10] H. Cuypers and J.I. Hall, The classification of 3-transposition groups with trivial centre, in *Groups, Combinatorics and Geometry* (M.W. Liebeck and J. Saxl, eds.), L.M.S. Lecture Notes **165**, Cambridge University Press, 1992, pp. 121–138.

[11] L. Finkelstein and W.M. Kantor, eds., *Groups and Computation*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **11**, A.M.S., 1993.

[12] D.R. Hughes and F.C. Piper, *Design Theory*, Cambridge University Press, 1985, reprinted in paperback, 1988.

[13] D.L. Johnson, *Presentations of Groups*, Cambridge University Press, 1990.

[14] J. Leech, Coset enumeration, in *Computational Group Theory* (M.D. Atkinson, ed.), Academic Press, London, 1984, pp. 3–18.

[15] J.S. Leon, On an algorithm for finding a base and strong generating set for a group given by generating permutations, *Math. Comp.* **35** (1980), 941–974.

[16] J.S. Leon, Permutation group algorithms based on partitions, I: theory and algorithms, *J. Symb. Comput.* **12** (1991), 533–583.

[17] S.A. Linton, Double coset enumeration, *J. Symb. Comput.* **12** (1991), 415–426.

[18] S.A. Linton, Constructing matrix representations of finitely presented groups, *J. Symb. Comput.* **12** (1991), 427–438.

[19] J. McKay, Computing with finite simple groups, in *Proceedings, Second International Conference on the Theory of Groups, Canberra, 1973*, Lecture Notes in Mathematics **372**, Springer, New York and Berlin, 1974, pp. 448–452.

[20] J. Neubüser, An elementary introduction to coset table methods in computational group theory, in *Groups – St. Andrews 1981* (C.M. Campbell and E.F. Robertson, eds.), L.M.S. Lecture Notes **71**, Cambridge University Press, 1982, pp. 1–45.

[21] A. Neumaier, Rectagraphs, diagrams, and Suzuki's sporadic simple group, in *Algebraic and Geometric Combinatorics* (E. Mendelsohn, ed.), Ann. Discrete Math. **15**, 1982, pp. 305–318.

[22] C.E. Praeger and L.H. Soicher, Low rank representations and graphs for sporadic groups, manuscript, 115 pages.

[23] M. Schönert, *et al.*, GAP – Groups, Algorithms and Programming, version 3, release 4, Lehrstuhl D für Mathematik, RWTH Aachen, 1994.

[24] C.C. Sims, Graphs and finite permutation groups, *Math. Z.* **95** (1967), 76–86.

[25] C.C. Sims, Computation with permutation groups, in *Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation* (S.R. Petrick, ed.), Assoc. Comp. Mach., 1971, pp. 23–28.

[26] C.C. Sims, Determining the conjugacy classes of a permutation group, in *S.I.A.M.-A.M.S. Proceedings* **4**, Amer. Math. Soc., 1971, pp. 191–195.

[27] C.C. Sims, *Computation with Finitely Presented Groups*, Cambridge University Press, 1994.

[28] L.H. Soicher, Presentations for some groups related to Co_1, in *Computers in Algebra* (M.C. Tangora, ed.), Marcel Dekker, New York, 1988, pp. 151–154.

[29] L.H. Soicher, A new existence and uniqueness proof for the O'Nan group, *Bull. London Math. Soc.* **22** (1990), 148–152.

[30] L.H. Soicher, On simplicial complexes related to the Suzuki sequence graphs, in *Groups, Combinatorics and Geometry* (M.W. Liebeck and J. Saxl, eds.), L.M.S. Lecture Notes **165**, Cambridge University Press, 1992, pp. 240–248.

[31] L.H. Soicher, GRAPE: a system for computing with graphs and groups, in *Groups and Computation* (L. Finkelstein and W.M. Kantor, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science **11**, A.M.S., 1993, pp. 287–291.

[32] J.A. Todd and H.S.M. Coxeter, A practical method for enumerating cosets of a finite abstract group, *Proc. Edinburgh Math. Soc.* **5** (1936), 26–34.

[33] S. Yoshiara, A classification of flag-transitive classical $c.C_2$-geometries by means of generators and relations, *Europ. J. Combinatorics* **12** (1991), 159–181.