DOUBLE-COSET ENUMERATION ALGORITHM FOR SYMMETRICALLY GENERATED GROUPS

MOHAMED SAYED

Received 25 March 2004 and in revised form 13 January 2005

A double-coset enumeration algorithm for groups generated by symmetric sets of involutions together with its computer implementation is described.

1. Introduction

The Todd-Coxeter algorithm described in [13] remains a primary reference for coset enumeration programs. It may be viewed as a means of constructing permutation representations of finitely presented groups. A number of effective computer programs for single-coset enumeration have been described, see, for example, [2, 7, 8].

Enumerating double cosets, rather than single cosets, gives substantial reduction in total cosets defined which leads to minimizing the storage (and time) needed. In [8, 9] Linton has developed two double-coset enumeration programs. The process in the earlier one, which is a corrected version of the algorithm proposed in [3], is viewed as a direct generalization of the ordinary single-coset enumeration. The later one is related to the present algorithm but with different calculations.

In this paper, we present a double-coset enumeration algorithm which is specially developed for groups *symmetrically generated* by involutions. Several finite groups, including all nonabelian finite simple groups, can be generated by symmetric sets of involutions, see, for example, [6, 10, 11]. The algorithm has been implemented as a Magma [1] program and this implementation has been used with success to check symmetric presentations for many finite simple groups.

2. Involutory symmetric generators of groups

Let G be a group and let $T = \{t_0, t_1, \ldots, t_{n-1}\}$ be a set of elements of order m in G. Defining $T_i = \langle t_i \rangle$ and $\overline{T} = \{T_0, T_1, \ldots, T_{n-1}\}$ allows us to define $N = \mathcal{N}_G(\overline{T})$, the set normalizer in G of $\overline{T}$. We say that T is a *symmetric generating set* for G if the following two conditions hold:

(i) $G = \langle T \rangle$, and
(ii) N permutes $\overline{T}$ transitively.

International Journal of Mathematics and Mathematical Sciences 2005:5 (2005) 699–715
DOI: 10.1155/IJMMS.2005.699

We call N the *control subgroup*. Conditions (i) and (ii) imply that G is a homomorphic image of the *progenitor*

$$m^{*n} : N, \tag{2.1}$$

where m^{*n} represents a free product of n copies of the cyclic group C_m and N is a group of automorphisms of m^{*n} which permutes the n cyclic subgroups by conjugation, see [5, 10].

Since in this paper we are only concerned with involutory symmetric generators we restrict our attention to the case $m = 2$ (while N will simply act by conjugation as permutations of the n involutory symmetric generators).

Theorem 2.1. *All nonabelian finite simple groups can arise as finite homomorphic images of progenitors of the form* $2^{*n} : N$.

Proof. Let H be a maximal subgroup of a finite simple group G. Suppose that $1 \neq \mathbf{t} \in G$, $\mathbf{t}^2 = 1$. Under the subgroup H, $\mathbf{t}^G$, the conjugacy class of $\mathbf{t}$ in G, splits into orbits as

$$\mathbf{t}^G = \mathcal{T}_1 \dot{\cup} \mathcal{T}_2 \dot{\cup} \cdots \dot{\cup} \mathcal{T}_r. \tag{2.2}$$

Without loss of generality, we may assume that $\mathcal{T}_1 = \{\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_{n-1}\}$ is not a subset of H. It is clear that

$$\mathcal{N}_G(\langle \mathcal{T}_1 \rangle) \geq \langle H, \mathcal{T}_1 \rangle = G, \tag{2.3}$$

since H is maximal in G and $\mathcal{T}_1$ is not a subset of H. Therefore,

$$1 \neq \langle \mathcal{T}_1 \rangle \lhd G, \tag{2.4}$$

and, since G is simple, we have

$$\langle \mathcal{T}_1 \rangle = G. \tag{2.5}$$

Moreover, if $\pi \in H$ and $\mathbf{t}_i^\pi = \mathbf{t}_i$ $(i = 0, 1, \ldots, n-1)$, then $\pi \in \mathfrak{F}(G)$ and so $\pi = 1$, that is, H permutes the elements of $\mathcal{T}_1$ faithfully (and transitively). Now, let 2^{*n} denote a free product of n copies of the cyclic group C_2 with involutory generators $t_0, t_1, \ldots, t_{n-1}$ and let $N \cong H$ consist of all automorphisms of 2^{*n} which permute the t_i as H permutes the $\mathbf{t}_i$:

$$\pi^{-1} t_i \pi = t_i^\pi = t_{\pi(i)} \quad \text{for } \pi \in N. \tag{2.6}$$

Then, clearly G is a homomorphic image of $2^{*n} : N$, a split extension of 2^{*n} by the permutation automorphisms N. □

Since the progenitor is a semidirect product (of $\langle T \rangle$ with N), it follows that in any homomorphic image G, we may use the equation

$$t_i \pi = \pi t_i^\pi = \pi t_{\pi(i)}, \tag{2.7}$$

or $i\pi = \pi i^\pi$ as we will more commonly write (see below), to gather the elements of N over to the left. Each element of the progenitor can be represented as πw, where $\pi \in N$

and w is a word in the symmetric generators. Indeed, this (symmetric) representation is unique provided w is simplified so those adjacent symmetric generators are distinct. Thus any additional relator by which we must factor the progenitor to obtain G must have the form

$$\pi w(t_0, t_1, \ldots, t_{n-1}), \tag{2.8}$$

where $\pi \in N$ and w is a word in T. Another consequence of this is that a relation of the form $(\pi t_i)^n = 1$ for some $\pi \in N$ in a permutation progenitor becomes

$$\pi^n = t_i t_{\pi(i)} \cdots t_{\pi^{n-1}(i)}. \tag{2.9}$$

In the next section we describe how a factor group

$$\frac{2^{*n} : N}{\pi_1 w_1, \pi_2 w_2, \ldots, \pi_s w_s} \tag{2.10}$$

may be identified.

3. Double-coset enumeration algorithm

If NxN is a double-coset of N in G, we have

$$NxN = N\pi wN = NwN, \tag{3.1}$$

where $x = \pi w \in G$, with $\pi \in N$, and w is a word in the symmetric generators. We denote this double-coset by $[w]$, for example, $[01]$ denotes the double-coset Nt_0t_1N. The double-coset $NeN = N$, where e is the identity element, is denoted by $[*]$.

We will allow i to stand for the coset Nt_i, ij for the coset Nt_it_j, and so on. We will also let i stand for the symmetric generator t_i when there is no danger of confusion. Thus we write, for instance, $ij \sim k$ to mean $Nt_it_j = Nt_k$ and $ij = k$ to mean $t_it_j = t_k$.

We define the subgroups N^i, N^{ij}, N^{ijk},… (for i, j, and k distinct) as follows:

$$\begin{aligned} N^i &= \mathscr{C}_N(\langle t_i \rangle), \\ N^{ij} &= \mathscr{C}_N(\langle t_i, t_j \rangle), \\ N^{ijk} &= \mathscr{C}_N(\langle t_i, t_j, t_k \rangle), \end{aligned} \tag{3.2}$$

or, more generally,

$$N^{i_1 i_2 \cdots i_m} = \mathscr{C}_N(\langle t_{i_1}, t_{i_2}, \ldots, t_{i_m} \rangle), \tag{3.3}$$

for $i_1, i_2, \ldots, i_m$ distinct.

Let g be an element of G. Then we define the *coset stabilizing subgroup* (of Ng in N) by

$$N^{(g)} = \{\pi \in N \mid Ng\pi = Ng\}. \tag{3.4}$$

Clearly $N^w \leq N^{(w)}$ for w, a word in the symmetric generators.

It is sometimes useful to have the notation of a *length* of a coset. The fact that for $\pi \in N$ we have

$$Nw(t_i)\pi = N\pi^{-1}w(t_i)\pi = Nw\left(t_i^{\pi}\right) = Nw'(t_i) \tag{3.5}$$

shows that all N cosets have a representative in $\langle T_0 \cup T_1 \cup \cdots \cup T_{n-1}\rangle$. We are in a position to define the length $L(Nw) = L(w)$ of a coset Nw. Firstly, we have $L(N) = 0$. If Nw has length n and $t \in T$, then Nwt has length at most $n+1$ and has length precisely $n+1$ if it does not have (or has not been proved to have) length at most n. We specify that all cosets of length $n+1$ have the form Nwt where $L(Nw) = n$ and $t \in T$.

If Nw_1 and Nw_2 are cosets in NwN, then we have that $L(w_1) = L(w_2)$, that is, all cosets of NwN have the same length. Thus, we can precisely compute the number of (right) cosets in NwN by using the simple but important following lemma.

Lemma 3.1. *The number of cosets in the double-coset* $[w] = NwN$ *is* $|N|/|N^{(w)}|$.

Proof. For $\pi_1, \pi_2 \in N$,

$$\begin{aligned} &Nw\pi_1 \neq Nw\pi_2 \Longleftrightarrow Nw\pi_1\pi_2^{-1} \neq Nw \Longleftrightarrow \pi_1\pi_2^{-1} \notin N^{(w)} \\ &\Longleftrightarrow N^{(w)}\pi_1\pi_2^{-1} \neq N^{(w)} \Longleftrightarrow N^{(w)}\pi_1 \neq N^{(w)}\pi_2. \end{aligned} \tag{3.6}$$

Thus, $Nw\pi_1$ and $Nw\pi_2$ are distinct cosets in NwN if and only if $N^{(w)}\pi_1$ and $N^{(w)}\pi_2$ are distinct cosets of $N^{(w)}$ in N. □

This last lemma makes double-coset enumeration over N to obtain the index of N in G a practical proposition. In order to obtain the index of N in G, we will find all double cosets $[w]$ and work out how many (single) cosets each of them contains. We will know that we have completed the double-coset enumeration when the set of right cosets obtained is closed under right multiplication. Moreover, the completion test is best performed by obtaining the orbits of $N^{(w)}$ on the symmetric generators. We need only identify, for each $[w]$, the double-coset to which Nwt_i belongs for one symmetric generator t_i from each orbit.

It is easy to see that if $N^{(w)}$ is a transitive subgroup of N, then Nw does not extend to a longer coset. Indeed, if the t_i are involutions and $N^{(w)}$ is a transitive subgroup of N, then all the cosets Nwt_i are shorter than Nw except when $Nw = N$, see [6, 10, 11].

With the observations of this section, we are in a position to carry out simple double-coset enumeration.

Example 3.2. Consider the group

$$G \cong \frac{2^{*4} : S_4}{(2,3) = \left[t_0 t_1\right]^2}, \tag{3.7}$$

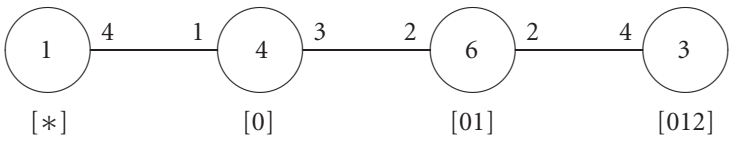
which means that the progenitor $2^{*4} : S_4$ is quotiented out by the relation $(2,3) = [t_0t_1]^2$.

The double cosets of length up to 3 are

$$[*], [0], [01], [010], [012]. \tag{3.8}$$

Next, we observe that $N = N(2,3) = Nt_0t_1t_0t_1$, which we write as $* \sim 0101$ in our notation. By postmultiplying both sides by t_1, we deduce that $Nt_1 = Nt_0t_1t_0$, that is, $010 \sim 1$. Furthermore, postmultiplying both sides by t_0 yields $Nt_1t_0 = Nt_0t_1$ which is $01 \sim 10$. We have thus shown the double-coset equality $[010] = [1] = [0]$. Also, we have shown that $Nt_1t_0 = N(t_0t_1)^{(0,1)} = Nt_0t_1$ and thus that $(0,1) \in N^{(01)}$. Since we also have $N^{(01)} \geq N^{01} = \langle (2,3) \rangle$, we have that $N^{(01)} \geq \langle (0,1),(2,3) \rangle$. Also $N^{(*)} = N = \langle (0,1,2,3),(2,3) \rangle$ and $N^{(0)} \geq N^0 = \langle (1,2,3),(2,3) \rangle$. Thus, we find that the double cosets $[*]$, $[0]$, and $[01]$ contain at most 1, 4, and 6 cosets, respectively. We have now shown that any length-3 coset is in $[012]$. However, we have that $102 \sim 012 = 01212 \cdot 21 = 0(0,3)21 \sim 321$. But $t_3t_2t_1 = (t_1t_0t_2)^{(0,2,1,3)}$ and so $N^{(012)} = N^{(102)} \geq \langle (0,1),(0,2,1,3) \rangle \cong D_8$, and thus $[012]$ has at most 3 cosets. Moreover, $N^{(012)}$ is a transitive subgroup of N, so cosets in this double-coset extend no further and we have completed the coset enumeration.

Thus, $|G:N| \leq 14$, so $|G| \leq 336 = |\mathrm{PGL}_2(7)|$, and the (relatively) easy task of finding generators for $\mathrm{PGL}_2(7)$, see [4], satisfying the required relations completes the identification of G with $\mathrm{PGL}_2(7)$. The correspondence between the 14 cosets and the 7 points and the 7 lines of the projective plane of order 2 is given in [11]. The Cayley diagram of G over N is given below:



(3.9)

where each circle is labeled with a double-coset $[w]$, with the number in the circle being the number of cosets in this double-coset. The numbers around the circles indicate the number of symmetric generators i such that $[wi]$ is a particular double-coset $[w']$.

4. Implementation

In this implementation we try to codify the technique that has been developed for hand working. A more detailed description and improvements are provided. Given a control subgroup N (of a group G) as permutations on n letters together with some relations, in which elements of N are written in terms of n (involutory) symmetric generators (of G), the program performs a double-coset enumeration for G over N. The program also returns what is essentially a Cayley graph of the action of G on the cosets of N. Each element of G is represented by a permutation of N followed by a word in the symmetric generators. The procedures for multiplying and inverting elements represented in this manner are also described. Indeed, the program allows the user readily to pass between the symmetric representation of an element of G and its action on the cosets of N. If the index $|G:N|$ is finite, the procedure does finish and succeed in finding the permutation representation of the group G. The proof is almost identical to that of Suzuki [12, pages 178–180].

4.1. Data structure. The control subgroup N is defined as a permutation group of degree n. The two sequences

$$\pi = [\pi_1, \pi_2, \ldots, \pi_m], \qquad w = [w_1, w_2, \ldots, w_m], \tag{4.1}$$

where $\pi_i \in N$ and w_i are words in the symmetric generators, represent the left- and the right-hand side of the problem relations. Also, ss and sg are defined as two sequences whose terms are the double-coset representative words and the coset stabilizing subgroups, respectively.

4.2. The algorithm design. The system contains a set of routines (procedures) such as *dcoset*, *eqper*, *reduce*, *main*, *names*, *x2per*, *t2per*, *sym2per*, *per2sym*, *mult*, and *invert*. In the remainder of this paper, we give a detailed description and an outline of some difficulties which arise with the implementation.

4.2.1. dcoset procedure. The procedure identifies, for each $[w]$, the double-coset to which Nwt_i belongs for one of the symmetric generators t_i from each orbit of the coset stabilizing subgroup $N^{(w)}$. Also, it produces

$$N^{(w)i} = \mathcal{C}_{N^{(w)}}(t_i), \tag{4.2}$$

a single-point stabilizer in $N^{(w)}$.

dcoset 1. Repeat 2 for each element ss$[i]$ (in the double-coset representative sequence ss) which has the maximal length.

dcoset 2. Identify the orbits of the coset stabilizing subgroup sg$[i]$. For each orbit which does not contain the last element of ss$[i]$, establish a new double-coset representative, ss$[j]$, as ss$[j]$ = Append(ss$[i]$, orbit representative), define sg$[j]$, the single-point stabilizer in sg$[i]$ by the same orbit representative.

4.2.2. eqper procedure. When the relations have been applied to a double-coset, permutations of the control subgroup N which fix that coset can be added as new generators to the corresponding coset stabilizing subgroup. In order to examine each double-coset $[w]$, each relation must be studied to establish whether a part of it is equivalent to this coset and so we can easily deduce a permutation of N which fixes that coset.

Given any two sequences (each represents a word in the symmetric generators)

$$e_1 = [a_1, a_2, \ldots, a_r], \quad e_2 = [b_1, b_2, \ldots, b_r], \quad a_i, b_i \in \{1, 2, \ldots, n\}; \tag{4.3}$$

the procedure checks the equivalence ($e_2^p = e_1$, for some $p \in N$) between them. If they are equivalent, it determines the permutation p of N such that $e_2^p = e_1$.

We know that

$$N \geq N^{b_1} \geq N^{b_1 b_2} \geq \cdots \geq N^{b_1 b_2 \cdots b_r}. \tag{4.4}$$

Assume that

$$n_1 = \left| N : N^{b_1} \right|, \qquad n_i = \left| N^{b_1 \cdots b_{i-1}} : N^{b_1 \cdots b_i} \right|, \quad i \in \{2, 3, \ldots, r\}. \tag{4.5}$$

Consequently, there exist transeversals

$$\{\tau_1,\dots,\tau_{n_1}\},\{\sigma_1,\dots,\sigma_{n_2}\},\{\rho_1,\dots,\rho_{n_3}\},\dots,\{\phi_1,\dots,\phi_{n_r}\} \tag{4.6}$$

such that

$$\begin{aligned} N &= N^{b_1}\tau_1 \dot\cup N^{b_1}\tau_2 \dot\cup \cdots \dot\cup N^{b_1}\tau_{n_1},\\ N^{b_1} &= N^{b_1b_2}\sigma_1 \dot\cup N^{b_1b_2}\sigma_2 \dot\cup \cdots \dot\cup N^{b_1b_2}\sigma_{n_2},\\ N^{b_1b_2} &= N^{b_1b_2b_3}\rho_1 \dot\cup N^{b_1b_2b_3}\rho_2 \dot\cup \cdots \dot\cup N^{b_1b_2b_3}\rho_{n_3},\\ &\vdots\\ N^{b_1\cdots b_{r-1}} &= N^{b_1\cdots b_r}\phi_1 \dot\cup N^{b_1\cdots b_r}\phi_2 \dot\cup \cdots \dot\cup N^{b_1\cdots b_r}\phi_{n_r}. \end{aligned} \tag{4.7}$$

Now, if $e_2^p = e_1$, $p \in N$, then we can find the permutation $p = \phi'\cdots\rho'\sigma'\tau'$, where

$$\tau' \in \{\tau_1,\dots,\tau_{n_1}\},\sigma' \in \{\sigma_1,\dots,\sigma_{n_2}\},\rho' \in \{\rho_1,\dots,\rho_{n_3}\},\dots,\phi' \in \{\phi_1,\dots,\phi_{n_r}\}, \tag{4.8}$$

as follows. Since $\sigma',\rho',\dots,\phi'$ fix b_1, the equation $b_1^p = a_1$ can be reduced to $b_1^{\tau'} = a_1$. Also, the permutations $\rho',\dots,\phi'$ fix b_2, so the equation $b_2^p = a_2$ can be reduced to $b_2^{\sigma'\tau'} = a_2$. Similarly, we have $b_3^{\rho'\sigma'\tau'} = a_3,\dots,b_r^{\phi'\cdots\rho'\sigma'\tau'} = a_r$. Thus, we can easily identify (in a recursive manner) the permutations $\tau',\sigma',\rho',\dots,\phi'$ and consequently p.

eqper 1. Set p as the identity element of S_n.

eqper 2. For $i \in \{1,2,\dots,r\}$ do
If $i = 1$, then trans = Transversal(N,N^{b_1}),
else trans = Transversal$(N^{b_1b_2\cdots b_{i-1}},N^{b_1b_2\cdots b_i})$.
If there exists a permutation trans$[j]$ such that $a_i^{(p^{-1}\cdot \text{trans}[j]^{-1})} = b_i$, then set $p =$ trans$[j]\cdot p$,
otherwise e_1 is not equivalent to e_2, leave the loop and return with a proper prompt.

4.2.3. Reduce procedure. Any word w in the symmetric generators is put by the procedure into its canonically shortest form. No other representations of group elements are used; words in the symmetric generators are simply shortened by application of the relations (and their conjugates under N). The relation

$$w_i = \pi_i, \tag{4.9}$$

where $w_i = t_{i_1}t_{i_2}\cdots t_{i_r}$ and $\pi_i \in N$, can be written as

$$t_{i_1}t_{i_2}\cdots t_{i_k} = \pi_i t_{i_r}t_{i_{r-1}}\cdots t_{i_{k+1}}, \tag{4.10}$$

where k is equal to $r/2$ or $(r+1)/2$ according to whether r is even or odd, respectively.

The procedure checks if a part of any given word in the symmetric generators of length equal to k is equivalent to $t_{i_1}t_{i_2}\cdots t_{i_k}$, the left-hand side of one of the previous relations using the *eqper* procedure, if so, the procedure replaces this part by $t_{i_r}t_{i_{r-1}}\cdots t_{i_{k+1}}$ after permuting by p^{-1} (a permutation obtained from the *eqper* procedure) and moves the permutation $\pi^{p^{-1}}$ over to the left of the whole word.

Reduce 1. For any sequence in the symmetric generators, set the pointer at the first letter.

Reduce 2. Check if the first k elements starting from the pointer position are equivalent to the left-hand side of the given relation, replace them by the right-hand side of the same relation after permuting by p^{-1}, and conjugate the preceding elements by the permutation $\pi_i^{p^{-1}}$.

Reduce 3. Shift the pointer one position and go to 2.

Moreover, in order to put the word w in the canonically shortest form, we may need to insert the identity element t_i^2 in a particular position (in this word). Assume that $w = [x_1, x_2, \ldots, x_{k-1}, y_1, y_2, \ldots, y_{k-1}]$, length$(w_i) = 2k-2$. The procedure checks if there exists an element s from the symmetric generators such that either $sy_1y_2\cdots y_{k-1} \sim t_{i_1}t_{i_2}\cdots t_{i_k}$ or $x_1x_2\cdots x_{k-1}s \sim t_{i_1}t_{i_2}\cdots t_{i_k}$ (or both). If so, the procedure replaces any of these two words (or both) by the right-hand side of the relation after permuting by a suitable permutation of N.

4.2.4. Main processing. In this section, we show how the program generates the double cosets and give an efficient method for handling the collapses. It is useful to note that the principle of termination will always be reached for any symmetric presentation of a finite group.

Input and initialization. Let $N \leq S_n$ be a permutation group of cardinality n. The problem relations are given as $w_i = \pi_i$, $i \in \{1, 2, \ldots, m\}$, where w_i are words in the symmetric generators and $\pi_i \in N$. The relations are sorted in ascending order according to the length of w_i, the variable *level* is set equal to $|L(w_1)/2 + 1|$, the double-coset representative words up to length equal to, *level* are obtained, and the corresponding stabilizing subgroups are defined.

Reduction. The same double-coset will often have many names and the purpose of the procedure is to find these coincidences by using the relations $w_i = \pi_i$. For every element sw (in the symmetric generators) in the ss sequence and every additional relation $w_i = \pi_i$, call the *reduce* procedure. Having obtained a new word of length less than the length of sw; the procedure deletes the double-coset of representative word sw and records that sw is equivalent to this new word. If a new word of length equal to the length of sw and equivalent to sw is obtained, then the procedure adds a permutation—which sw should be conjugated by to obtain this new word—to the coset stabilizing subgroup in the sg sequence.

Collapses. It was mentioned before that it is convenient to have some way of recording, in a sequence, all the new relations that were obtained during the reduction step. From time to time we pack the sequences of double-coset representative elements and coset stabilizing subgroups, reclaiming the space that was occupied by the redundant elements and

this might lead to the collapse of part of or the entire double-coset diagram. Rather than starting collapses at the end of each level or delaying them to the end of the processing, they may be initiated at specific levels. The process of determining when the collapses should be started is demonstrated as follows: suppose $w_i = \pi_i$, $i \in \{1,2,\ldots,m\}$, are the problem relations such that length$(w_i) = 2k_i + 1$ or $2k_i$. During the processing, the suitable levels to check the collapses are k_i.

Collapse 1. Set level 1 = level.

Collapse 2. If new relations have been defined during level 1 which is equal to n, say, try to reduce again the double cosets of length equal to $n - 1$ using these relations, together with the problem relations, otherwise go to 4.

Collapse 3. Set level $1 = n - 1$ and go to 2.

Collapse 4. Call the *dcoset* procedure l times, where l = level – level 1, starting from level 1 and modify ss and sg sequences according to the new data.

Termination. The double-coset enumeration is complete when the set of right cosets obtained is closed under right multiplication. Since N is a finitely generated subgroup of countable index in a finitely presented group G, the point of termination will always be reached.

Termination 1. Set l = length(ss).

Termination 2. Call the *dcoset* procedure. If l = length(ss), then call the *output* process, else call the reduction process.

Output. The information contained in the double-coset representatives and the coset stabilizing subgroups sequences may be portrayed graphically in the form called a Cayley diagram. Unfortunately this is not suitable to be produced using the current Magma programming language. It is, however, possible to display the output in a tabular form. The use of this algorithm makes it possible to show the following: double-coset labels together with their representative words, orbit representatives and the number of elements in each orbit, right multiplication images, coset stabilizing subgroups and their orders, the number of cosets in each double-coset, and the order of the group G.

4.2.5. Names procedure. When the control subgroup N is large, but the number of the cosets of N in G is small, the action of the elements of G on these cosets can be easily obtained. This procedure builds cst, a sequence of length equal to the number of the cosets of N, whose terms represent words in the symmetric generators. These words form a complete set of coset representatives for N in G. The first element of the cst sequence is the empty word followed by $n = |N : N^{(i)}|$ words of length one, and so forth.

Names 1. Set cst = $[\cdot]$.

Names 2. Construct two sequences, the first one contains an empty word and the second one contains words of the form $[i]$, $i = 1,2,\ldots,$Degree(N). Append them to the sequence of sequences cst.

Names 3. For each double-coset representative word ss[i] of length $l > 1$, apply the elements of the right transversal of sg[i] in N to this coset to obtain a sequence of all cosets of length l. Append this sequence to cst.

4.2.6. x2per procedure. As mentioned earlier, each element of the group G can be represented by a permutation on n letters; n is the cardinality of the permutation group N, followed by a word in the n involutory symmetric generators. Given a permutation $x \in N$, the procedure constructs a permutation, xp, say, which gives the action of x on the cosets of N in G.

x2per 1. Initialize xs as a sequence of integers of length equal to the number of the cosets of N.

x2per 2. For each i, j such that length(cst[i]) = length(cst[j]), if $(\text{cst}[i])^x = \text{cst}[j]$, then set $xs[i] = j$.

x2per 3. Convert the sequence of integers xs to xp, a permutation on the cosets of N.

4.2.7. t2per procedure. This procedure gives the action of the symmetric generators on the cosets of N in G. In general t_i in its action on the cosets of N has the form

$$(*, i)(j, ji)\cdots(jk, jki)\cdots(jkl, jkli)\cdots, \quad \text{for } i, j, k, l \text{ distinct.} \tag{4.11}$$

In practice our symmetric presentation is given in terms of a set of generators of N together with one of the symmetric generators. So, if the action of one of the symmetric generators, t_i, on the cosets of N is known, we can obtain the action of the others on the cosets of N by permuting this symmetric generator by NN (the control subgroup N in its action on the cosets of N).

t2per 1. Initialize ts as a sequence of integers of length equal to the number of the cosets of N.

t2per 2. For each i, j such that length(cst[i]) = length(cst[j]) + 1, if (cst[i] cat 1) = cst[j], then set $ts[i] = j$ and $ts[j] = i$.

t2per 3. Convert the sequence of integers ts into $tp[1]$, a permutation on the cosets of N.

t2per 4. Construct $tp[i]$, $i = 2, 3, \ldots, n$ by permuting $tp[1]$ by NN.

4.2.8. sym2per procedure. This procedure converts a symmetrically represented element x of G into a permutation acting on the cosets of N in G. Let $x = \pi w$, where $\pi \in N$ and $w = [i, j, k, \ldots]$, a word in the symmetric generators. Then $p = x2per(\pi) \cdot tp[i] \cdot tp[j] \cdot tp[k]\ldots$, where p is a permutation acting on the cosets of N.

4.2.9. per2sym procedure. The procedure converts a permutation p (acting on the cosets of N) of G into its symmetric representation. The image of one under p gives the coset representative for Np as a word w in the symmetric generators. Multiplication of p by the symmetric generators in w, in reverse order, yields a permutation which can be identified with an element of N by its action on cosets of length one.

per2sym 1. Assume $p = \pi w$, obtain w as $w = \mathrm{cst}[1^p]$.

per2sym 2. Obtain π as a permutation on the cosets of N, using the equation $\pi = pw^{-1}$.

per2sym 3. Identify the action of π on the cosets of length one as $\pi = [j \mid (1^{tp[i]})^{\pi} = (1^{tp[j]}),$ for all $i, j \in \{1,2,\ldots,n\}]$.

Finally write π as a permutation of S_n.

4.2.10. mult procedure. Suppose that πw and σu are two symmetrically represented elements in G. We have that $\pi w \cdot \sigma u = \pi\sigma w^{\sigma} u = \rho v$, where $\pi, \sigma, \rho \in N$ and w, u, v are words in the symmetric generators. Elements represented as above are transformed into permutations on the cosets of N and the procedure performs the multiplication operation before it transforms the result back into the symmetric representation.

mult 1. Multiply w^{σ} and u after converting them into permutations on the cosets of N using the *t2per* procedure and store the result in a temporary variable called *temp*.

mult 2. Transform *temp* to its symmetric representation $\rho' v$ using the *per2sym* procedure.

mult 3. Identify ρ as $\rho = \pi\sigma\rho'$.

On the other hand, the procedure *reduce* can be used to put the word $w^{\sigma} u$ in its canonically shortest form.

4.2.11. Invert procedure. The procedure gives the inverse of any symmetrically represented element πw in the group G. We have

$$(\pi w)^{-1} = w^{-1}\pi^{-1} = \pi^{-1}\pi w^{-1}\pi^{-1} = \pi^{-1}\left(w^{-1}\right)^{\pi^{-1}} = \pi' w', \tag{4.12}$$

where $\pi' w'$ is a symmetrically represented element of G.

5. Examples

In order to illustrate the process, we consider the following small but interesting examples.

Example 5.1.

$$G \cong \frac{2^{*4} : S_4}{(3,4) = \left(t_1 t_2\right)^2} \cong \left\langle N, T \mid N \cong S_4 \cong \langle x, y\rangle,\ t_i^{\pi} = t_{\pi(i)},\ (3,4) = \left(t_1 t_2\right)^2 \right\rangle, \quad x : (1,2,3,4), \quad y : (3,4). \tag{5.1}$$

The input consists of the control subgroup $N \cong S_4$ acting on $\{1,2,3,4\}$, together with the two sequences $\pi = [(3,4)]$ and $w = [[1,2,1,2]]$. The program, therefore, has found all the double cosets (see the manual double-coset enumeration in Section 3) as shown in Table 5.1.

Table 5.1. Result of the double-coset enumeration of $\mathrm{PGL}_2(7)$ over S_4.

Double-coset label	Coset representative word	Orbit representative (number)	Right mult image	Coset stabilizing subgroup (order)	Number of single cosets
1	[∗]	1 (4)	2	$\langle(1,2,3,4),(3,4)\rangle$ (24)	1
2	[1]	1 (1)	1	$\langle(2,4),(3,4)\rangle$	4
		2 (3)	3	(6)	
3	[12]	1 (2)	2	$\langle(1,2),(3,4)\rangle$	6
		3 (2)	4	(4)	
4	[123]	1 (4)	3	$\langle(1,2),(1,4)(2,3)\rangle$ (8)	3

The list of the 14 single cosets together with their equivalent names is [[[]], [[1], [2], [3], [4]], [[[12], [21]], [[23], [32]], [[34], [43]], [[24], [42]], [[14], [41]], [[13], [31]]], [[[123], [213], [432], [342], [124], [214], [431], [341]], [[143], [413], [234], [324], [142], [421], [231], [321]], [[132], [312], [423], [243], [134], [314], [421], [241]]]]. The action of the element $x = (1,2,3,4)$ on single cosets is given by

$$xp = (2,3,4,5)(6,7,8,10)(9,11)(12,13). \tag{5.2}$$

Also, the action of our 4 symmetric generators is given as follows:

$$\begin{aligned} t_1 &: (1,2)(3,6)(4,11)(5,10)(7,13)(8,12)(9,14),\\ t_2 &: (1,3)(2,6)(4,7)(5,9)(8,12)(10,13)(11,14),\\ t_3 &: (1,4)(2,11)(3,7)(5,8)(6,12)(9,14)(10,13),\\ t_4 &: (1,5)(2,10)(3,9)(4,8)(6,12)(7,13)(11,14). \end{aligned} \tag{5.3}$$

The second example shows the ability of the process to perform harder reductions and to handle collapses.

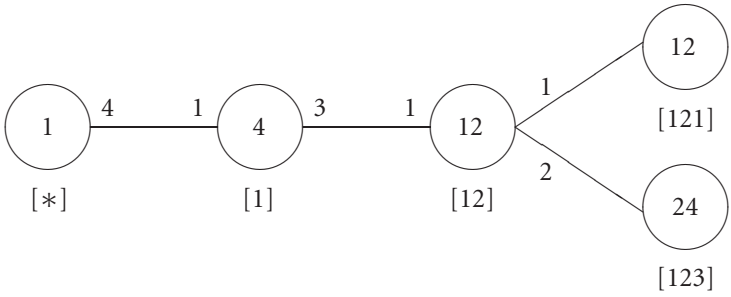
Example 5.2.

$$G \cong \frac{2^{*4} : S_4}{\left[(1,2)(3,4)t_1\right]^5, \left[(1,2,3)t_1\right]^{11}}. \tag{5.4}$$

The input consists of the control subgroup $N \cong S_4$ acting on the set $\{1,2,3,4\}$, together with the two sequences $\pi = [(1,2)(3,4),(1,2,3)]$ and $w = [[1,2,1,2,1],[2,1,3,2,1,3,2,1,3,2,1]]$.

We now commence a double-coset enumeration of G over N. Using the *dcoset* procedure, the program finds all double cosets of length up to level = 3. So the Cayley graph

has the following diagram:

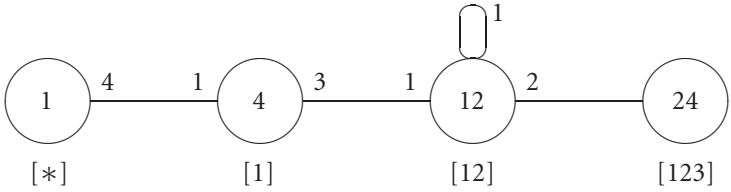


(5.5)

Since

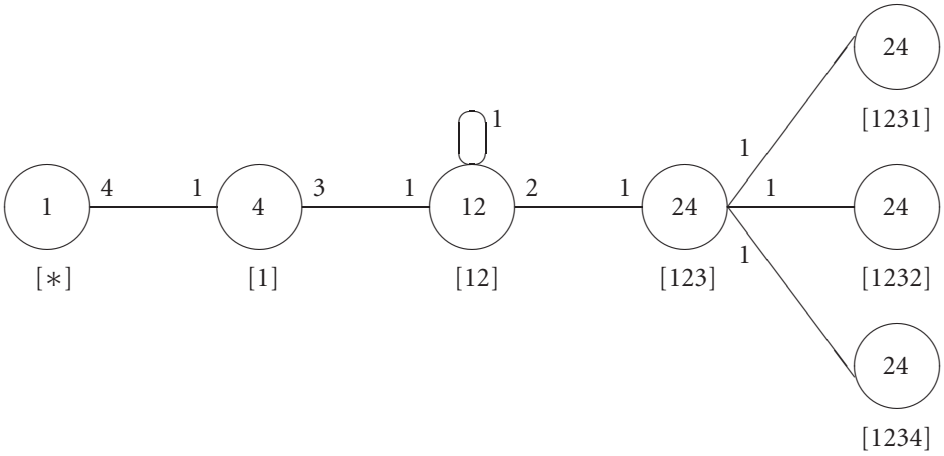
$$12121 \sim * \Longrightarrow 121 \sim 12, \tag{5.6}$$

then we have the double-coset equality $[121] = [12]$. So the Cayley diagram is reduced to



(5.7)

Then, the coset enumeration is extended to level four as

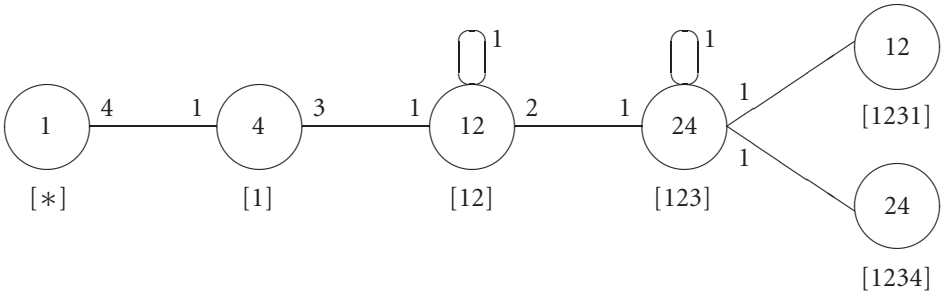


(5.8)

The above Cayley diagram can be reduced by repeated use of the first relation. We have that

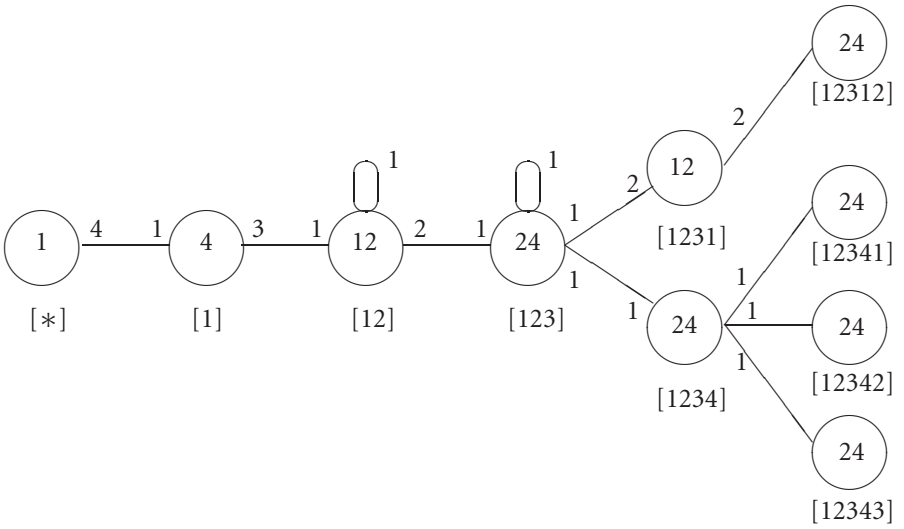
$$\begin{aligned} &1231 = 121\cdot 131 = 12(1,3)(2,4)13 \sim (12)^{(1,3)(2,4)}13 \sim 3413 \Longrightarrow (1,3)(2,4) \in N^{(1231)},\\ &1232 = 1(1,4)(2,3)23 \sim 1^{(1,4)(2,3)}23 \sim 423 \sim 123 \Longrightarrow [1232] = [123], \end{aligned} \tag{5.9}$$

so the Cayley diagram becomes



(5.10)

At this time the check for collapses fails and the Cayley diagram at the beginning of level five has the following shape:



(5.11)

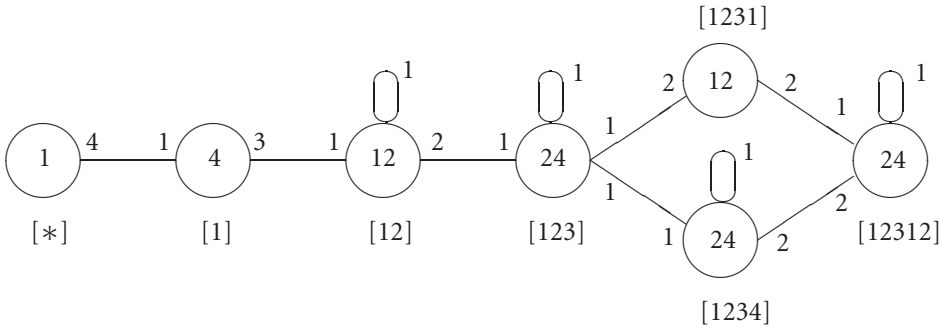
Now

$$\begin{aligned}
12341 &= 1231 \cdot 141 \sim 3413(1,4)(2,3)14 \sim 214214 \\
&\sim 12412 \sim 12312 \Longrightarrow [12341] = [12312], \\
12342 &= 1232 \cdot 242 \\
&= 1(2,3)(1,4)23242 \sim 423(1,3)(2,4)24 \\
&\sim 24124 \sim 12312 \Longrightarrow [12342] = [12312], \\
12343 &= 12(1,2)(3,4)34 \sim 2134 \sim 1234 \Longrightarrow [12343] = [1234].
\end{aligned} \tag{5.12}$$

Applying the process again, level six has been reached which will cause a part of the Cayley diagram to collapse to its final form. We have

$$\begin{aligned}
123121 &= 123(1,2)(3,4)12 \sim 21412 \\
&\sim 2(1,4)(2,3)142 \sim 3142 \Longrightarrow [123121] = [1234], \\
123123 &\sim 21321 \sim 12312 \Longrightarrow [123123] = [12312].
\end{aligned} \tag{5.13}$$

The Cayley diagram takes the following shape:



(5.14)

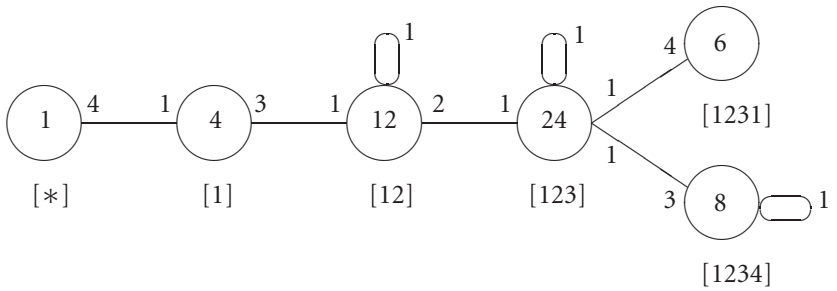
Finally

$$\begin{aligned}
1234 &= 12341 \cdot 1 \sim 124121 \sim 124(1,2)(3,4)12 \sim 21312 \\
&\sim 2(1,3)(2,4)132 \sim 4132 \Longrightarrow (1,4,2) \in N^{(1234)},
\end{aligned} \tag{5.15}$$

also

$$\begin{aligned}
1231 &= 1234 \cdot 41 \sim (1234)^{(1,2,4)}41 \sim 243141 \sim 243(1,4)(2,3)14 \sim 31214 \\
&\sim 3(1,2)(3,4)124 \sim 4124 \Longrightarrow (1,4,3,2) \in N^{(1231)}.
\end{aligned} \tag{5.16}$$

The Cayley diagram now terminates as



(5.17)

We may readily construct our symmetric generators as permutations on $1+4+12+24+6+8=55$ letters and verify that they do indeed satisfy the relations we assumed. It is easy to recognize the group G—in this case the projective general linear group $\mathrm{PGL}_2(11)$ of order $55\times 24=1320$—and check that it does contain such a symmetric generating set.

6. Conclusion

The present double-coset enumeration algorithm is different from the existing coset enumeration techniques in the way it handles the elements of the group and for telling the user more about the structure of the group. For instance, the operations of inversion and multiplication can be performed manually (or mechanically) by means of short algorithms. Also, it is helpful to define the group in terms of generators and relations in the standard way; we can find a symmetric generating set for the group and from this generating set we can determine the relations which we need to add to our progenitor presentation.

It should be emphasized that the performance of our enumerator depends largely on the defining set of additional relations. Although the enumerator seems to be slow, it works well with $\pi_{ij}=iji$ relations. The other factor which particularly affects the efficiency of the implementation is that it is heavily Magma dependent. Our program is run on a Sun Sparc Station 5 with CPU clock rate 110 MHz. For example, the groups $G_2(4):2$ and $\mathrm{Suz}:2$ (see symmetric presentations in [11]) take about 0.2 second and 0.35 second of CPU time, respectively. Single-coset enumerations of these groups (using the same machine) take about 0.04 second and 0.11 second of CPU time, respectively. However, the time required using our implementation for the example $\mathrm{PGL}_2(11)$ shown above is relatively high comparing with the time required using single-coset enumeration.

References

[1] W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265.

[2] 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.

[3] J. H. Conway, *An algorithm for double coset enumeration?*, Computational Group Theory (Durham, 1982) (M. D. Atkinson, ed.), Academic Press, London, 1984, pp. 33–37.
[4] J. H. Conway, R. T. Curtis, S. P. Norton, R. A. Parker, and R. A. Wilson, *Atlas of Finite Groups*, Oxford University Press, Eynsham, 1985.
[5] R. T. Curtis, *Symmetric presentations. I. Introduction, with particular reference to the Mathieu groups M_{12} and M_{24}*, Groups, Combinatorics & Geometry (Durham, 1990), London Math. Soc. Lecture Note Ser., vol. 165, Cambridge Press, Cambridge, 1992, pp. 380–396.
[6] R. T. Curtis and Z. Hasan, *Symmetric representation of the elements of the Janko group J_1*, J. Symbolic Comput. **22** (1996), no. 2, 201–214.
[7] J. Leech, *Coset enumeration*, Computational Group Theory (Durham, 1982) (M. D. Atkinson, ed.), Academic Press, London, 1984, pp. 3–18.
[8] S. A. Linton, *The maximal subgroups of the sporadic groups* **Th**, **Fi**24 *and* **Fi**24′ *and other Topics*, Ph.D. thesis, University of Cambridge, London, 1989.
[9] ———, *Double coset enumeration*, J. Symbolic Comput. **12** (1991), no. 4-5, 415–426.
[10] M. Sayed, *Computational methods in symmetric generation of groups*, Ph.D. thesis, University of Birmingham, Birmingham, 1998.
[11] ———, *Nested symmetric representation of elements of the Suzuki chain groups*, Int. J. Math. Math. Sci. **2003** (2003), no. 62, 3931–3948.
[12] M. Suzuki, *Group Theory. I*, Grundlehren der Mathematischen Wissenschaften, vol. 247, Springer, Berlin, 1982.
[13] J. A. Todd and H. S. M. Coxeter, *A practical method for enumerating cosets of a finite abstract group*, Proc. Edinb. Math. Soc. **5** (1936), 26–34.

Mohamed Sayed: Department of Mathematics and Computer Science, Faculty of Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait
E-mail address: msayed@mcs.sci.kuniv.edu.kw