PROC. SECOND INTERNAT. CONF. THEORY OF GROUPS, CANBERRA 1973, pp. 347-356.

20F05 (20-04)

A REIDEMEISTER-SCHREIER PROGRAM

George Havas

1. Introduction

The Reidemeister-Schreier method yields a presentation for a subgroup H of a group G when H is of finite index in G and G is finitely presented. This paper describes the implementation and application of a FORTRAN program which follows this method. The program has been used satisfactorily for subgroups of index up to several hundred.

Following the theory of Reidemeister and Schreier (see for example Magnus, Karrass, Solitar [4], 2.3, p. 86), we see that we require the coset table of H in G . The program described is implemented as a set of subroutines called by the Todd-Coxeter program described in Cannon, Dimino, Havas and Watson [1] and we shall consider it in that context.

2. The procedure in detail

The Reidemeister-Schreier program commences by finding the coset table of H in G . Directly following the theory of Reidemeister and Schreier, the program finds Schreier generators for H and next finds a set of Reidemeister relators in terms of the Schreier generators.

At this stage the presentation is usually not in a useful form. The number of Schreier generators is of the order of $n_g[G : H]$ where n_g is the number of generators of G ; the number of Reidemeister relators is of the order $n_r[G : H]$ where n_r is the number of relators in the presentation of G . In view of this, the program goes on further to improve the presentation using obvious but *ad hoc* techniques. In particular, this is done by using a canonical form for the relators, by eliminating redundant generators and by attempting relator simplification.

A somewhat more detailed description is given in the following paragraphs.

Supported by the Commonwealth Scientific and Industrial Research Organization.

2.1 Set up the coset table

Given $G = \langle g_1, \ldots, g_{n_g} \mid R_1 = R_2 = \ldots R_{n_r} = 1 \rangle$ and $H = \langle h_1, \ldots, h_{n_h} \rangle$ the coset table of H in G is computed. (The actual subgroup generators, the h_m, are required only by the Todd-Coxeter part of the program and are not required subsequently.) The subscript i will be used to run through the cosets of H in G.

2.2 Compute coset representatives

Minimal Schreier coset representatives are computed by constructing a minimal spanning tree for the coset table. For each coset i, C_i will represent the corresponding minimal Schreier coset representative. Further K will designate a coset representative function. This means that K maps words in the g_j onto a coset representative system for $G \bmod H$. In this case $K(w)$ will be the minimal Schreier representative of the coset of w.

2.3 Compute Schreier generators

The Schreier generators $S_{i,j}$ are computed using the formula

$$S_{i,j} = C_i g_j \left(K\left(C_i g_j\right)\right)^{-1} .$$

The Schreier generators are freely reduced by the program. Some may be trivial and any such are noted.

2.4 Compute Reidemeister relators

The Reidemeister relators $r_{i,k}$ are computed using a Reidemeister rewriting process t.

If $w = g_{j_1}^{\varepsilon_1} g_{j_2}^{\varepsilon_2} \ldots g_{j_n}^{\varepsilon_n}$ then

$$t(w) = S_{i_1,j_1}^{\varepsilon_1} S_{i_2,j_2}^{\varepsilon_2} \ldots S_{i_n,j_n}^{\varepsilon_n}$$

where i_k is the coset of the initial segment of w preceding $g_{j_k}^{\varepsilon_k}$ if $\varepsilon_k = 1$ and i_k is the coset of the initial segment of w up to and including $g_{j_k}^{\varepsilon_k}$ if $\varepsilon_k = -1$.

Using this rewriting we have

$$r_{i,k} = t\left(C_i R_k C_i^{-1}\right) .$$

The program eliminates all occurrences of trivial Schreier generators (previously noted for this purpose). Then the relators are freely and cyclically reduced.

All relators are converted to a canonical form as they are computed, and the relators are maintained in this form throughout. The canonical form is based on an ordering $<$ of the generators $S_{i,j}$. (The program uses the lexicographic ordering for the $S_{i,j}$.)

If x_k, y_k are Schreier generators we define our ordering $<$ on the relators by

$$x_1^{\varepsilon_1} < y_1^{\delta_1} \text{ if } x_1 < y_1 \text{ or } x_1 = y_1 , \; \varepsilon_1 = 1 , \; \delta_1 = -1 ;$$

$$x_1^{\varepsilon_1} x_2^{\varepsilon_2} \dots x_m^{\varepsilon_m} < y_1^{\delta_1} y_2^{\delta_2} \dots y_n^{\delta_n} \text{ if } m < n ;$$

and inductively

$$x_1^{\varepsilon_1} x_2^{\varepsilon_2} \dots x_m^{\varepsilon_m} < y_1^{\delta_1} y_2^{\delta_2} \dots y_m^{\delta_m}$$

if

$$x_1^{\varepsilon_1} x_2^{\varepsilon_2} \dots x_{m-1}^{\varepsilon_{m-1}} < y_1^{\delta_1} y_2^{\delta_2} \dots y_{m-1}^{\delta_{m-1}}$$

or

$$x_1^{\varepsilon_1} x_2^{\varepsilon_2} \dots x_{m-1}^{\varepsilon_{m-1}} = y_1^{\delta_1} y_2^{\delta_2} \dots y_{m-1}^{\delta_{m-1}} \text{ and } x_m^{\varepsilon_m} < y_m^{\delta_m} .$$

We choose as our canonical relator the least (with respect to this ordering) of the set of relators made up of all cyclic rotations of the given relator and its formal inverse.

As each relator is computed its canonical representative is inserted into a relator list, provided it is not a repeat of a relator already there. The canonical form notion substantially reduces the number of relators in the presentation.

2.5 Eliminate redundant generators

When we have completed the above steps we have a presentation for H , namely

$$H = \langle S_{i,j} \mid r_{i,k} = 1 \rangle ,$$

with duplicate relators removed.

This presentation is still highly redundant. In particular there are usually many redundant generators in the sense that there are many relators containing exactly one occurrence of a particular generator. Such generators can be removed by substitution.

The most obvious approach is to remove one redundant generator at a time from each relator. However this turns out to be very time-consuming in computer implementation. Three possible techniques are discussed below.

2.5.1 ELIMINATION TECHNIQUE 1

This is the technique actually implemented for regular use in the Reidemeister-Schreier program. The eliminations are batched together to save computing time.

Each relator in the relator list is examined for any generator occurring exactly once (provided the relator is independent of previously discovered but not yet eliminated redundant generators). The shorter relators are examined first so that the generators selected for elimination will tend to have short generator strings as their equivalents.

Whenever a redundant generator is found the associated relator is removed from the relator list, the generator is marked redundant, and the value of the generator and its inverse computed. On the completion of such a pass through the relator list all remaining relators are examined for occurrences of redundant generators. Each such occurrence is eliminated by substituting for the redundant symbol its computed value. The new relators are again freely and cyclically reduced and a new relator list is formed.

On completing the redundant generator elimination we repeat this step (for new redundancies may now have appeared in the relators). We continue repeating this step till no further redundancies are found, when we go to the last step, 2.6.

2.5.2 ELIMINATION TECHNIQUE 2

Eliminate one generator at a time. Heuristically this seems superior in that we may select for elimination the generator with shortest equivalent string at each stage. Unfortunately the Reidemeister-Schreier method frequently gives us presentations with hundreds of generators and relators, and hundreds of redundancies, and this approach takes too long.

2.5.3 ELIMINATION TECHNIQUE 3

A compromise between techniques 1 and 2 is to eliminate redundant generators with values of the same length at the one time, doing this elimination for increasing length. This has the advantage of ensuring that all length zero and one eliminations are done as soon as possible. Length zero and one eliminations are most desirable for they do not increase relator lengths, whereas higher length eliminations may well increase relator lengths, and usually do.

In §3 comparisons of the above three techniques in terms of execution times and "niceness" of ensuing presentation are given.

2.6 Simplify the presentation

When no obviously redundant generators remain we resort to other attempts to simplify the presentation. There are an unlimited number of possibilities for this. The most obvious is looking for relator substrings which have shorter equivalent strings. At this stage only the following technique is included in the program.

2.6.1 SIMPLIFICATION TECHNIQUE

All relators are checked to see if they are of the form $S_{i,j}^{n}$. If any such relators are found then all other relators are processed for strings in the $S_{i,j}$ of known order. All long strings (that is of length exceeding $n/2$) are replaced by their shortest equivalent counterparts. Further, for even n , strings $S_{i,j}^{-n/2}$ are replaced by $S_{i,j}^{n/2}$.

After this simplification the program returns to step 2.5, and if no further redundancies appear, the program terminates. Otherwise steps 2.5 and 2.6 are repeated till no further improvement is obtained.

3. Examples

In this section some applications in determining previously unknown Macdonald groups (see [3]) are presented. Also some other test examples are presented to give an indication of the performance of the different possible techniques and the program as a whole.

Macdonald group coset enumerations are notoriously difficult (see [1]), and this forced us to resort to the techniques described here to determine them by coset enumeration based methods.

Wamsley [5] describes a technique for constructing the largest finite nilpotent p-factors of groups. Such an approach is in fact much more suitable for determining Macdonald groups and enables their determination more easily.

3.1 Determination of a Macdonald group

First let us consider in detail a reasonably easy application of the Reidemeister-Schreier program. The Macdonald group $G(-2, -2)$ is defined

$$G(-2, -2) = \langle a, b \mid b^{-1}a^{-1}bab^{-1}aba^{2} = a^{-1}b^{-1}aba^{-1}bab^{2} = 1 \rangle .$$

In practice it is impossible to do the coset enumeration $G|\langle 1\rangle$ using Todd-Coxeter programs. This is not surprising for $[G : \langle a\rangle] = 729$ (by Todd-Coxeter), and it is easy to prove that the order of a is either 27 or 81 , making the order of G either 19683 or 59049 .

The subgroup $H = \langle [a, b], [a^{-1}, b], [b, a], [b^{-1}, a] \rangle$ is of index 9 in G

(hence H is the commutator subgroup), and the enumeration of cosets is easy. Using the Reidemeister-Schreier program the following presentation for H was found

$$H = \langle x, y, z \mid x^2y^{-1}xy^{-2} = y^2z^2yz = xz^2yxz^{-1}xz^2y^{-1} = xyz^{-2}x^{-1}(yz^{-2})^2 = xz^2y^{-1}xz^3y^{-1}xz^4y^{-1} = 1 \rangle .$$

Further, the enumeration $H|\langle 1 \rangle$ was done, revealing that H has order 6561, whence G has order 59049 and a has order 81.

3.2 Test examples

A set of test examples was run using each of the three techniques described in 2.5, and also with the final program. The following groups and subgroups (taken from Coxeter and Moser [2]) were used.

(a) $G_1 = \langle a, b \mid a^3 = b^6 = (ab)^4 = (ab^2)^4 = (ab^3)^3 = a^{-1}b^{-2}a^{-2}b^{-2}a^{-1}b^{-2}ab^2a^2b^2ab^2 = 1 \rangle .$

(G_1 is a presentation for PSL(3, 3), of order 5616.)

$H_1 = \langle a, b^2 \rangle$ is the Hessian group of order 216, $[G_1 : H_1] = 26$.

(b) $G_2 = \langle a, b \mid a^4 = b^4 = (ab)^4 = (a^{-1}b)^4 = (a^2b)^4 = (ab^2)^4 = (a^2b^2)^4 = [a, b]^4 = (a^{-1}bab)^4 = 1 \rangle .$

(G_2 is a presentation for $B_{2,4}$, of order 4096.)

$H_2 = \langle a, b^2 \rangle$, $|H_2| = 64$, $[G_2 : H_2] = 64$.

(c) $G_3 = \langle a, b, c \mid a^{11} = b^5 = c^4 = (bc^2)^2 = (abc)^3 = (a^4c^2)^3 = b^2c^{-1}b^{-1}c = a^4b^{-1}a^{-1}b = 1 \rangle .$

(G_3 is a presentation for M_{11}, of order 7920.)

$H_3 = \langle a, b, c^2 \rangle$ is PSL(2, 11) of order 660, $[G_3 : H_3] = 12$.

(d) $G_4 = \langle a, b, c \mid a^{11} = b^5 = c^4 = (ac)^3 = b^2c^{-1}b^{-1}c = a^4b^{-1}a^{-1}b = 1 \rangle .$

(G_4 is a "better" presentation for M_{11}.)

$H_4 = \langle c, b, c^2 \rangle$ is again PSL(2, 11) of order 660, $[G_4 : H_4] = 12$.

(e) $G_5 = \langle a, b, c \mid a^3 = b^7 = c^{13} = (ab)^2 = (bc)^2 = (ca)^2 = (abc)^2 = 1 \rangle$.

(G_5 is $G^{3,7,13}$, a presentation for PSL(2, 13) , of order 1092 .)

$H_5 = \langle ab, c \rangle$ is dihedral of order 26 , $[G_5 : H_5] = 42$.

(f) $G_6 = \langle a, b, c \mid a^3 = b^7 = c^{14} = (ab)^2 = (bc)^2 = (ca)^2 = (abc)^2 = 1 \rangle$.

(G_6 is $G^{3,7,14}$, of order 2184 .)

$H_6 = \langle ab, c \rangle$ is dihedral of order 28 , $[G_6 : H_6] = 78$.

In fact it is easy to read presentations for H_5 and H_6 off from the presentations for G_5 and G_6 , but it is still interesting to observe the behaviour of the algorithm in these cases.

Each of the three elimination techniques was used on each of the test groups. These runs were actually made at an early stage of program development, before the introduction of the canonical form or the simplification technique. The following table indicates the nature of the presentations obtained at that time.

Column a indicates the number of generators in the presentation,
Column b indicates the number of relators, and
Column c indicates the length of the longest relator.

	Technique 1			Technique 2			Technique 3		
Subgroup	a	b	c	a	b	c	a	b	c
H_1	3	35	80	3	37	114	3	37	114
H_2	2	85	264	2	76	160	2	81	132
H_3	4	27	157	3	19	42	4	25	115
H_4	3	13	84	3	11	24	3	11	24
H_5	3	12	182	3	13	134	3	13	142
H_6	3	16	56	3	16	80	3	16	80

We might say one presentation is better than another if it has fewer generators, fewer relators and/or shorter relators. The table indicates that sometimes the heuristically better technique produces a "worse" presentation.

The fact that the better technique sometimes leads to a worse presentation is troubling. The reason for this is that redundant generators may be eliminated in different orders when different techniques are used, leading to different presentations.

Other factors may also affect the final presentation. Obviously the nature of the initial group presentation is very relevant to the nature of the subgroup presentation deduced. The presentation for H_4 is much "nicer" than the presentation for H_3. This is because the presentation for G_4 is better than the presentation for G_3.

It is interesting to note that if we take different generators for the subgroup we may get significantly better or worse presentations.

The reason for this is that the coset table may be generated in a different order, leading to a different ordering on the Schreier generators. When given a choice of a number of redundant subgroup generators to eliminate in one relator, the elimination procedure uses the rule of selecting the largest with respect to the canonical ordering. Thus different Schreier generators are eliminated under different conditions.

Sample timings for each of the three techniques are as follows. For H_1 technique 3 took 1.5 times as long as technique 1 while technique 2 took 2.4 times as long. For H_5 technique 3 took 1.6 times as long as technique 1 while technique 2 took 8.5 times as long. These timing considerations justify the selection of technique 1 for the final program implementation.

It is also interesting to consider the nature of the presentation yielded by the original Reidemeister-Schreier method and the nature of the presentation obtained by the final program implementation. Columns a, b and c have the same significance as before.

	Original Reidemeister-Schreier Presentation			Final output Presentation		
Subgroup	a	b	c	a	b	c
H_1	27	156	14	2	29	168
H_2	65	576	16	2	67	204
H_3	25	96	15	4	22	111
H_4	25	72	11	3	10	21
H_5	157	546	14	3	13	65
H_6	85	294	13	2	12	66

At first sight, it may seem that neither of these presentations is of much use. The former has too many generators and too many relators while the latter has relators which are too long, and perhaps too many relators. But looking at the final output presentation shows us otherwise. Of the examples given, the presentation for

H_2 looks the worst. However the first four relations in the presentation produced for H_2 are $b^2 = a^4 = (ab)^4 = (a^2b)^4 = 1$ and these alone suffice for a presentation of H_2. In a similar fashion three of the first four relations of the presentation produced for H_6 are $a^2 = (ab)^2 = b^{14} = 1$, indeed a presentation for H_6.

If we can successfully select a likely presentation for H from the relators, it is an easy matter, using coset enumeration, to show whether the remaining relators are consequences of the selected set. So even when we get an apparently ungainly presentation, we may well be able to extract a useful presentation from it.

The final programmed algorithm, including the coset enumeration, took 6.5 seconds CPU time on a CDC 6600 to find the presentation for H_1.

3.3 The Reidemeister-Schreier abelianized

In certain cases (the Macdonald groups are again a case in point) there are abelian subgroups of a group of which we do not know the structure. In such cases it is useful to perform an abelianized Reidemeister-Schreier. What we do is abelianize each relator at each stage of the computation, and this greatly simplifies the ensuing presentation.

Let us consider the use of the abelianized Reidemeister-Schreier in the context of two other previously unknown Macdonald groups. Consider $G = G(3, 5)$. Again the enumeration $G|\langle 1\rangle$ is too difficult. However, the enumeration $G|H$ where $H = \langle b\rangle$ can be done easily enough, to get $[G : H] = 8$.

Unfortunately we do not know the order of b.

If we try to use the straight Reidemeister-Schreier program to find a presentation for H we get a nasty presentation (3 generators, 10 relators, longest relator length 66). However, H is abelian of course. Using the abelianized Reidemeister-Schreier we obtain the one relator presentation $H = \langle b \mid b^{16}\rangle$, whence $|G| = 128$.

The second example is $G(-3, -5)$. $[G : H = \langle b\rangle] = 32$, and a presentation for H given by the abelianized Reidemeister-Schreier is $H = \langle b \mid b^{12}\rangle$, whence $|G| = 384$. However, using the straight Reidemeister-Schreier program we get a presentation with 3 generators, 34 relators and longest relator length 684.

4. Data structures and computation techniques

For those interested in the actual computer implementation, brief details of some of the methods used are given in this section.

After computing the coset table we need to compute coset representatives, Schreier generators and Reidemeister relators for the subgroup. All of these are of unpredictable length so a list structure is most appropriate. In order to compute Schreier coset representatives we need two locations per coset of the table. In addition we need n_g locations per coset for pointers to each of the n_g Schreier generators associated with each coset.

Thus, in addition to the data structures used in [1], we must augment the coset table by n_g columns and we require a reasonably large amount of list space. The list processing is done in-line in the FORTRAN code.

In order to perform the conversion of relators to canonical form, the relators are made into circular lists after free and cyclic reduction. This facilitates the examination of all rotations of a relator without the use of additional storage.

Finally, the relator list, in which all the canonical form relators for the subgroup are stored, is kept sorted according to the previously defined order. This makes the search for duplication easier and enables the program to simply process the relators in relator list order when applying generator elimination and relator simplification techniques.

References

[1] John J. Cannon, Lucien A. Dimino, George Havas and Jane M. Watson, "Implementation and analysis of the Todd-Coxeter algorithm", *Math. Comp.* 27 (1973), 463-490.

[2] H.S.M. Coxeter and W.O.J. Moser, *Generators and relations for discrete groups* (Ergebnisse der mathematik und ihrer Grenzgebiete, Band 14. Springer-Verlag, Berlin, Göttingen, Heidelberg, New York; 1st edition, 1957; 2nd edition, 1965). MR19,527 (1st edition); MR30#4818 (2nd edition).

[3] I.D. Macdonald, "On a class of finitely presented groups", *Canad. J. Math.* 14 (1962), 602-613. MR25#3992.

[4] Wilhelm Magnus, Abraham Karrass, Donald Solitar, *Combinatorial group theory* (Pure and Appl. Math., 13. Interscience [John Wiley & Sons], New York, London, Sydney, 1966). MR34#7617.

[5] J.W. Wamsley, "Computation in nilpotent groups (Theory)", these Proc.

Canberra College of Advanced Education,
Canberra City, ACT 2601.