AN ALGORITHMIC SOLUTION FOR A WORD PROBLEM IN GROUP THEORY

N. S. MENDELSOHN

1. Introduction. This paper describes a systematic procedure which yields in a finite number of steps a solution to the following problem. Let G be a group generated by a finite set of generators $g_1, g_2, g_3, \ldots, g_r$ and defined by a finite set of relations $R_1 = R_2 = \ldots = R_k = I$, where I is the unit element of G and $R_1, R_2, \ldots, R_k$ are words in the g_i and g_i^{-1}. Let H be a subgroup of G, known to be of finite index, and generated by a finite set of words, $W_1, W_2, \ldots, W_t$. Let W be any word in G. Our problem is the following. Can we find a new set of generators $W_1^*, W_2^*, \ldots, W_s^*$ for H, together with a set of representatives $h_1 = 1, h_2, \ldots, h_u$ of the right cosets of H (i.e. $G = H1 + Hh_2 + \ldots + Hh_u$) such that W can be expressed in the form $W = Uh_p$, where U is a word in W_i^*, W_i^{*-1}. In particular, a solution to this problem would yield an algorithm for deciding when a word in G lies in H and a method of writing this word as a word in W_i^*, W_i^{*-1}. The method will consist of an application of the Todd-Coxeter process of coset enumeration **(1, 6)**, together with the Schreier process as described by Marshall Hall in **(2, 5)**. It is clear that the Schreier process is algorithmic. With regard to the Todd-Coxeter process no proof has yet been given that the process stops after a finite number of steps. In **(3)** Leech gives an argument to make it appear plausible that the process must close after finitely many steps but this argument falls short of a formal proof. (Such a formal proof is given in Section 3 of this paper.) A second paper by Leech **(4)** does solve the word problem described above in the special case where the Todd-Coxeter enumeration is carried out without the introduction of redundant cosets, but fails when redundant cosets appear in the enumeration. When Leech's method does work, the word problem is solved in terms of the original generators W_i. In the method described here it is necessary first to express the W_i^* as words in the W_i if one insists on solving the problem in terms of the original generators. In this case Leech's method is superior to ours. However, our method works in all cases, and in Section 5 we give an example where Leech's method fails. Also, to be noted is that our algorithm is practical in the sense that it can be programmed readily on a computing machine and results can be obtained in reasonable time.

2. The Schreier process. This process is clearly set forth in **(2** and **5)** by Marshall Hall. We merely summarize the method here to make the paper self-contained.

Received August 16, 1963.

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

Suppose the decomposition of G into right cosets of H, $G = H1 + Hh_2 + \ldots + Hh_u$, is such that the representatives $1, h_2, \ldots, h_u$ are known. If the element g of G is in the coset Hh_i, put $\phi(g) = h_i$. Then, by Schreier's theorem, the elements $\alpha_{rs} = h_r g_s \{\phi(h_r g_s)\}^{-1}$ are generators of H. Suppose now we have a procedure for computing $\phi(k)$ for any element $k \in G$. Let $g \in G$ and write $g = a_1 a_2 \ldots a_m$ where each $a_i = g_j$ or $a_i = g_j^{-1}$ for some g_j. Put $k_0 = 1$, $k_1 = \phi(a_1)$, $k_2 = \phi(a_1 a_2), \ldots, k_m = \phi(a_1 a_2 \ldots a_m) = \phi(g)$. Then $g = \{(k_0 a_1 k_1^{-1})(k_1 a_2 k_2^{-1}) \ldots (k_{m-1} a_m k_m^{-1})\} k_m$. Now it is easily seen that each $k_i a_{i+1} k_{i+1}^{-1}$ is an α_{rs} or an α_{rs}^{-1} and $k_m = h_v$ for some v.

3. The Todd-Coxeter process. To complete the solution of our main problem, it suffices to have a method of computing the set $1, h_2, \ldots, h_u$ together with a method of computing $\phi(g)$. A modification of the Todd-Coxeter process can be mechanized completely to yield a solution to these parts. To be noted is the fact that although we prove that the process is finite whenever H is of finite index, we can give no estimate of any bound for the number of steps required for the process to stop. Thus we do not have a procedure for deciding the finiteness of the index of H. The process is as follows.

Let $R = I$ be any relation. Then $R = a_1 a_2 \ldots a_f$, where each $a_i = g_j$ or g_j^{-1} for some j. Now $R = I$ is equivalent to $S = I$ and to $T = I$, where $S = a_i a_{i+1} \ldots a_f a_1 a_2 \ldots a_{i-1}$ and $T = a_i^{-1} a_{i-1}^{-1} \ldots a_1^{-1} a_f^{-1} a_{f-1}^{-1} \ldots a_{i+1}^{-1}$. We call $S = I$ and $T = I$ trivially equivalent to $R = I$. It is clear that if g_i or g_i^{-1} appears in R, then there exist S and T such that g_i is both the first letter of S and the last letter of T. Also, if a generator g_i is free, we can write down two relations $g_i g_i^{-1} = I$ and $g_i^{-1} g_i = I$. Hence, by adding to the set of defining relations for G a number of trivially equivalent relations together with a number of relations of the forms $g_i g_i^{-1} = I$, $g_i^{-1} g_i = I$, we can obtain a set of defining relations for G, possibly redundant, such that each generator of G appears as the first and as the last letter of some relation. We call such a set of relations, for want of a better term, an algorithmic set. Suppose now, we imagine that a positive integer is assigned to each right coset Hg of H, the integer being taken as the name of the coset. To the coset H is assigned the integer 1. Now if i, j, and k are cosets and g is a generator, the equations $i.g = j$ and $k.g = j$ imply $i = k$ and $j.g^{-1} = i$. The equations $i.g = j$ are to be entered into a set of tables as follows. Let $R_u = I (u = 1, 2, \ldots, k)$ be an algorithmic set of relations for G, and let W_v $(v = 1, 2, \ldots, t)$ be a set of words in G which generate H. Suppose that $R_u = a_1 a_2 \ldots a_q$, where each a_i or its inverse is a generating element of G. With R_u we build a table with $q + 1$ columns and infinitely many rows. At the top of the table will appear the letters $a_1 a_2 \ldots a_q$ in succession but these will straddle the columns, i.e. a_1 will have the first column on its left and the second on its right, a_2 will have the second column on its left and the third column on its right, etc. Also in the ith row of the table, the first and last entries are to be the integer i. (See Table I.)

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

TABLE I

R_u

a_1	a_2	a_3	a_4	...	a_q
1					1
2					2
3					3
4					4
.					.
.					.
.					.
i					i
.					.
.					.
.					.

With each of the words W_v a similar table is built with the exception that such a table is to consist of exactly one row, its beginning and end entries being 1. Suppose that the R_u and W_v are ordered and the tables are placed side by side, the W-tables coming after the R-tables. The unfilled places in the sets of tables are then ordered from left to right and each place in the ith row of any table precedes each place in the jth row of any other table if $i < j$. The object is to enter integers into the tables in such a way that if r and s fill consecutive places in a row of one of the tables and the letter g straddles the columns in which r and s appear, then $r.g = s$. The construction proceeds as follows.

If any of the words W_v are of length 1, say $W_v = g_i$, then the table for W_v will be ${}_1g_{i1}$. Now in any of the tables if 1 is to the left or right of g_i or g_i^{-1}, a 1 is to be placed on the other side of this letter in the same row, in accordance with the equations $1.g_i = 1$ and $1.g_i^{-1} = 1$. When all possible 1's are entered, the integer 2 is placed in the first empty space. This will yield an equation of the type $1.a = 2$, where $a = g_j$ or g_j^{-1}. Fill all possible places with the digit 2 consistent with the equations $1.a = 2$ and $2.a^{-1} = 1$. Now suppose this has been continued with the integers $1, 2, \ldots, i-1$. The integer i is then placed in the first available place. This induces an equation $i.b = j$ and $j.b^{-1} = i$. In placing the integer i in the tables consistent with these equations, other relations of the type $i.c = k$ and $k.c^{-1} = i$ may be induced. One then adds at all possible places the integers i and k and if other equations are induced by these entries, the appropriate integers are added at all appropriate places until no further equations are induced. One then normally proceeds to the integer $i + 1$. There are two possible situations in which one does not proceed to the integer $i + 1$. The first is *closure*. Closure occurs if after all possible additions of the integer i and smaller integers the first i rows of every table are completely filled. In this case we stop the process entirely and consider the tables for each R_u to have exactly i rows, all of whose entries are from the integers $1, 2, 3, \ldots, i$. The second case in which one does not

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

proceed to $i + 1$ is *redundancy*. Redundancy occurs as follows: In entering the integer i into the table and then entering smaller integers because of induced equations, the entries at different points in the table may imply equations of the type $k.a = n$, $m.a = n$ with $k \neq m$. This means, of course, that the integers m and k represent the same coset. Suppose that $m > k$. We then replace every appearance of m by k. Then we delete from each table the row whose first entry is m. Next, every integer n which is greater than m is replaced by $n - 1$ at all of its appearances. If no new equations are induced by the altered entries, one fills in further entries in accordance with the new equations. If a new redundancy appears, we alter the tables accordingly, and if closure results, we stop. If closure does not result and there are no further redundancies or entries possible because of induced equations, we continue as follows. Go to the first available place in the tables and enter into it the smallest integer which does not appear in any previous place and continue as before.

We now prove that if H is of finite index, closure must be reached after a finite number of steps. We first prove a lemma.

LEMMA 1. *After a finite number of steps the first r rows of all the tables are stabilized, i.e. none of the entries are further altered because of redundancy.*

Proof. Suppose that at a given stage S is the sum of the *distinct* integers which appear in the first r rows of the tables. If a redundancy alters the first r rows, its effect is to replace all occurrences of a certain integer by a smaller integer. Hence S is decreased. But $S \geqslant \frac{1}{2}r(r + 1)$. Then, after a finite number of redundancies, either S is reduced to $\frac{1}{2}r(r + 1)$ in which case closure occurs, or else redundancies no longer affect the first r rows.

Now suppose that the process continues indefinitely without closure. We form a permutation representation of G on the set of all positive integers as follows. Let g_i be a generator and j be any positive integer. Let R_v be one of the relations which start with g_i and R_u one of the relations which end with g_i. After the jth row has become stable, let k be the second entry of the jth row in the table for R_v and let m be the second last entry in the jth row of the table for R_u. Then $j.g_i = k$ and $m.g_i = j$. Now with g_i associate the permutation P_{g_i}, where $P_{g_i}: j \to k$ and $m \to j$. Since j is an arbitrary positive integer and appears both as an image and a pre-image, P_{g_i} is, as claimed above, a permutation of all the positive integers. Let P_G be the group generated by all the P_{g_i}. Then the mapping $g_i \to P_{g_i}$ is a homomorphism of G onto P_G. We now show that P_G is a transitive group. In fact, we show that every integer is in the orbit of 1. If this were not so, let M be an orbit (set of transitivity) and let u be its smallest member. If $u \neq 1$, then its first appearance in the tables is to the right of an integer $v < u$. This means that there is a g_i (or g_i^{-1}) in G such that P_{g_i} (or $P_{g_i}^{-1}$): $v \to u$. Hence v is in M, a contradiction. Now from the tables corresponding to W_v we see that for every generator of H the corresponding element of P_G fixes 1. It is clear that no element outside H fixes 1.

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

We can thus interpret P_G as the representation of G by the permutations induced in the cosets of H by right multiplication by elements of G. But since P_G is transitive on infinitely many elements, the order of P_G is infinite, and hence the index of H is infinite, a contradiction.

Now that we have proved that closure occurs after finitely many steps, the group P_G is of finite order. Coset representatives for the different cosets are obtained as follows. Since P_G is transitive, then corresponding to any integer i there is an element $x \in P_G$ such that $x: 1 \to i$. As P_G is finite, such an x is easily determined. In the mapping $G \to P_G$ let x_i be any element of G such that $x_i \to x$. Take x_i to be the coset representative of the coset i. Finally, let g be any element of G. Then $g = b_1 b_2 \ldots b_m$, where each $b_i = g_j$ or g_j^{-1}. Let $P_{b_1} P_{b_2} \ldots P_{b_m}: 1 \to k$; then $\phi(g) = x_k$. We have now completed a solution to our original problem.

4. Remarks on the solution. In this section we remark on the role played by the redundant relations and give a heuristic discussion of why one should not expect closure to occur always if one does not start with an algorithmic set of relations. In setting up our tables, in what follows, we shall not adhere strictly to the prescription of entering each new integer at the first available space. We shall follow the practice of Todd and Coxeter of introducing each new integer at a convenient place to induce as many equations as possible and to avoid redundancy if it can be avoided. Where we get closure, anyway, without the introduction of the redundant relations, we shall not introduce them.

First, we show that the introduction of a redundant (non-trivial) relation can change the formation of the tables in an essential way. Consider the following example. G is generated by $\{A, X\}$ subject to the relations $X^4 = X^2AXA^{-2} = I$. Let H be the subgroup generated by X. The tables form as follows (we omit obviously unnecessary parts):

X	X	X	X
1	1	1	1	1
2	3	4	5	2
6	7	8	9	6

X	X	A	X	A^{-1}	A^{-1}
1	1	1	2	3	2	1
2	3	4	6	7	3	2
3	4	5			7	3
4	5	2	3	4	6	4
5	2	3	7	8		5
6	7	8	9	6	4	6
7	8	9	5	2	1	7
8	9	6	4	5	9	8
9	6	7	1	1	7	9

X	
1	1

We get redundance from the underlined parts of the tables. Adjusting for this redundance produces closure and we ultimately get the representation $X \to (1)(2345)$ and $A \to (1235)(4)$. It can easily be proved that *any* way of filling the tables yields at least three redundant cosets, so that redundancy is

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

essential. Now consider the following implication of $X^2AXA^{-2} = X^4 = I$. We have $X^2AX = A^2$ or $AX = X^2A^2$, from which

$$AXAX = X^2A^2AX = X^2A^3X = X^2AX^2AX^2 = A^2XAX^2.$$

The relation $AXAX = A^2XAX^2$ implies $XA = AXAX$ or $AXAXA^{-1}X^{-1} = I$. If this is added to the set of relations used in the formation of the tables, redundancy is avoided.

Our second example will illustrate the role played by the addition of trivially redundant relations. Let G be the group generated by A, B, C, with a single definining relation $ABC = I$. If we do not add redundant relations, the table appears as follows:

A	B	C	
1	2	3	1
2	4	5	2
3	6	7	3
4	.	.	4
.			.
.			.
.			.

The table is, of course, infinite. Note that this table does not yield a permutation representation of G. In fact, the mapping induced by A maps all the integers onto the even integers; the mapping induced by B maps the even integers onto the odd integers, while that induced by C maps the odd integers onto all the integers. Note, particularly, that no matter how one tries to fill in the tables, it is impossible for the same integer to appear in the columns both to the left and to the right of B. If, on the other hand, we use the algorithmic set $ABC = BCA = CAB = I$, the tables do yield a permutation representation of G.

It is to be noted here that the necessity for adding the redundant relations is due to our insistence that the first and last entries of a row be filled in first. Todd and Coxeter did not use this condition. We added it to make our proofs technically simpler. It also makes machine programming easier.

5. Two examples. We close with two examples to illustrate the effectiveness of the techniques used. In this connection the technique has actually been used for subgroups of index as high as 600.

Let G be the group generated by A and X with the sole definining relation $X^2AX = A^2$. Let H be the subgroup generated by

$$X, \qquad Y = AX^4A^{-1}, \qquad Z = A^{-1}X^4A.$$

Since this example is essentially that given in Section 4, coset enumeration yields that H is of index 5 and also the following permutation representation: $X \to (1)(2345)$, $A \to (1235)(4)$.

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

From these we find as a possible set of coset representatives the following table:

Coset	*Representative*
1	I
2	A
3	AX
4	AX^2
5	AX^3

It will actually be convenient to equate the integers to the coset representative and write $1 = I$, $2 = A$, $3 = AX$, etc. Corresponding to these representatives, we obtain in the table below the Schreier generators for H (omitting the occurrences of I) together with their expressions in terms of X, Y, Z.

SCHREIER TABLE

Schreier representative	X, Y, Z *equivalent*
X	X
$A^2X^{-1}A^{-1}$	X^2
AX^4A^{-1}	Y
$AXAX^{-3}A^{-1}$	ZXY^{-1}
AX^3A	ZXZ
$AX^2AX^{-2}A^{-1}$	X^2ZXY^{-1}

We note, in passing, that the Schreier representatives are redundant and that their expression in terms of X, Y, Z is certainly not unique, since $AX^3A = ZXZ = YZX$, etc.

We now show how, in a purely mechanical way, to express any word in G in terms of X, Y, Z and a coset representative. A simple example will suffice. Consider the word $AXAX$. We first write a line of an enumeration table using our permutation representation:

A	X	A	X	
1	2	3	5	2

We then use this to write an equation

$$AXAX = (1A2^{-1})(2X3^{-1})(3A5^{-1})(5X2^{-1})2,$$

where now the integers are interpreted as the coset representatives. Each bracketed term is a Schreier generator (or its inverse). Consulting the Schreier table, we obtain $AXAX = (XZ)A$.

Since the expression of a word in H in terms of any set of generators is not uniquely determined, one might inquire whether a word in H reduces to the identity. In general, this is a futile problem since it is known that such a general word problem is unsolvable. There is, however, a heuristic technique for obtaining relations in H. One simply starts with any word W_1 in G. In our example, one would then use the relation $X^2AX = A^2$ to obtain a word

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press

W_2 equivalent to W_1. One then applies the above process to W_1 and W_2. In this way we have obtained, for example, the relation $ZXZ = YZX$. Incidentally, this relation implies $Y = ZXZX^{-1}Z^{-1}$, so that Y is a redundant generator.

As a second example we shall show how to represent the free group on n generators as a permutation group on the set of all positive integers. Here, of course, it is mandatory to use redundant relations. Let $g_1, g_2, \ldots, g_n$ be the generators. Apply the enumeration to $g_i g_i^{-1} = I$, $g_i^{-1} g_i = I$ $(i = 1, 2, \ldots, n)$. In this case, the enumeration proceeds smoothly and no redundant cosets appear. In particular, for the cyclic free group ($n = 1$, g the generator) one obtains the following representation:

$$g\colon\ 2i + 1 \to 2i - 1 \text{ for } i = 1, 2, 3, \ldots$$
$$g\colon\ 2i \to 2i + 2 \text{ for } i = 1, 2, 3, \ldots$$
$$g\colon\ 1 \to 2.$$

It is instructive to work out a similar representation for two generators.

References

1. H. S. M. Coxeter and W. O. J. Moser, *Generators and relations for discrete groups*, Ergebnisse der Mathematik und ihrer Grenzgebiete, Chapter 2, pp. 12–18.
2. John Leech, *Coset enumeration on digital computers*, Proc. Camb. Phil. Soc., *59* (1963), 257–267.
3. ——— *Some definitions of Klein's simple group of order 168 and other groups*, Proc. Glasgow Math. Assoc., *5*, Part 4, 166–175 (1962).
4. Marshall Hall, *The theory of groups* (New York, 1959), Chapter 7, pp. 94–106.
5. John Todd, *Survey of numerical analysis* (Mew York, 1962), Chapter 15 by Marshall Hall, pp. 534–538.
6. J. A. Todd and H. S. M. Coxeter, *A practical method for enumerating cosets of a finite abstract group*, Proc. Edinburgh Math. Soc. (2), *5* (1936), 34–36.

University of Manitoba, Winnipeg

https://doi.org/10.4153/CJM-1964-052-3 Published online by Cambridge University Press