SCHREIER GENERATORS:
AN ALGORITHM FOR EXPRESSING THE SCHREIER GENERATORS OF A SUBGROUP AS WORDS IN THE GENERATORS USED TO DEFINE THE SUBGROUP

BY

FRANKLIN GEORGE REYNOLDS

A THESIS

PRESENTED TO THE

FACULTY OF GRADUATE STUDIES AND RESEARCH

OF THE

UNIVERSITY OF MANITOBA

IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

OCTOBER, 1965

THE WRITER WISHES TO EXPRESS SINCERE APPRECIATION TO DR. N. S. MENDELSOHN OF THE DEPARTMENT OF MATHEMATICS FOR HIS INTEREST AND DIRECTION.

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

In 1963, Mendelsohn presented an algorithmic solution for a word problem in Group Theory. The method is based on the Schreier process and the Todd-Coxeter process for coset enumeration. For even the simplest examples, the method is labourious, and Leech and Trotter have presented machine programs which will enumerate the cosets. This thesis describes a machine program which, given the coset enumeration, will express a word expressed in the generators of the group as a word in the generators of a subgroup times a coset representative. If this process is applied to the Schreier generators of the subgroup, a table results which enables one to carry out hand computations on word problems in a reasonable time. Such a table has been used to solve theoretical problems.

CHAPTER I

THE SCHREIER PROCESS

1.1 NOTATION AND TERMINOLOGY. Suppose a set of elements $S=(s_i)$, $i \in I$, where the index set I need not be finite but where its elements are assumed to be well ordered, are given. A "word" is a finite succession $a_1 a_2 \ldots\ldots a_k$ where each a_i is an s_i or its inverse, s_i^{-1}. The identity, 1, is the void word. A "reduced word" is a void word or one in which no pair $a_i a_{i+1}$ is the product of an element of S and its inverse (or vice-versa). Two words are "equivalent" if the reduced word obtained by multiplying one by the inverse of the other is the void word. Here, the inverse of $a_1 a_2 \ldots\ldots a_k$ is defined to be $a_k^{-1} a_{k-1}^{-1} \ldots\ldots a_2^{-1} a_1^{-1}$.

If S is a subset of a group G, then S "freely generates" G if and only if every $g \neq 1$ lying in G can be written in exactly one way in form $g=a_1^{n_1} a_2^{n_2} \ldots\ldots a_p^{n_p}$, a_i lying in S, $a_i \neq a_{i+1}$ and the n_i are integers, positive or negative. A group G is "free" if and only if some subset S freely generates it.

A "Schreier system" is a set B of elements of a group G in which each b lying in B has the following properties:

If 1) $b=a_1 a_2 \ldots\ldots a_n$ is reduced as written,

then 2) $a_1 a_2 \ldots\ldots a_{n-1}$ is an element of B.

G is said to have a two sided Schreier system if as well

3) $a_2 \ldots\ldots a_k$ is an element of B for every b lying in B.

1.2 SOME PRELIMINARY LEMMAS. In (2), Marshall Hall gives a proof of the following lemmas:

LEMMA 1.1. If the group G is generated by a set of elements Y: y_1, y_2, ... y_n, then if S: s_1, ..., s_n generates the free group F, there is a homomorphism of F onto G determined by $s_i \rightarrow y_i$ for i = 1, 2, ... ,n. A similar statement holds even if Y is not finitely generated.

LEMMA 1.2 If U is a subgroup of the free group F, it is possible to choose the representatives of the right cosets of U as a Schreier system. If U is normal, the Schreier system can be chosen to be two sided.

LEMMA 1.3 Every subgroup of a free group is free.

1.3 THE DEFINITION OF THE SCHREIER GENERATORS AND THE SCHREIER PROCESS.

DEFINITION: If U is a subgroup of any group F, if s runs over the generators of F and their inverses, g over the right coset representatives of U in F and if $\phi(gs)$ is the coset representative of the coset containing gs, then the elements $gs(\phi(gs))^{-1}$ are called the "Schreier generators" of U.

In (2), Marshall Hall shows that the Schreier generators generate U.

Marshall Hall sets forth the Schreier process in (2). Let $e_{xy} = g_x s_y (\phi(g_x s_y))^{-1}$ be a Schreier generator. Suppose a procedure is given for determining $\phi(h)$ for any element h lying in F. Let $g = a_1 a_2 \ldots a_n$ lie in F where each a_r is a generator (or the inverse of a generator) of F. Set $h_0 = 1$, $h_1 = \phi(a_1)$, $h_2 = \phi(a_1 a_2)$, ... ,$h_n = \phi(a_1 a_2 \ldots a_{n-1} a_n) = \phi(g)$

Obviously, $g = ((h_0 a_1 h_1^{-1})(h_1 a_2 h_2^{-1}) \ldots\ldots (h_{n-1} a_n h_n^{-1}))h_n^{-1}$ where each $h_{i-1} a_i h_i^{-1}$ is an e_{xy} or an e_{xy}^{-1} and each h_j is a coset representative.

CHAPTER II

MENDELSOHN'S ALGORITHM

2.1 INTRODUCTION. The principle of coset enumeration is stated. The Todd-Coxeter process for coset enumeration is stated, as is Mendelsohn's algorithm. The Todd-Coxeter coset enumeration process will be described by a more rigid set of rules than originally presented by the authors in order to make it more amenable to machine computation.

2.2 COSET ENUMERATION. Let a group G with a subgroup H be given. Let G be freely generated by a set S of generators which satisfy a set $R_n = I$ (n = 1, 2, ... ,h) of relations which are chosen in such a way that each element of S appears as the first letter of some R_n and as the last letter of some R_n. Assign a positive integer to each right coset Hg of H in G, and take as the name of the coset that number. Assign the number 1 to the coset H·1. Let V_r (r = 1, 2, ... , l) be a set of words which generate H. Suppose $R_n = a_1 a_2 \ldots\ldots a_t$ where a_i lies in S for all i. For each R_n, build a table with t + 1 columns and infinitely many rows. Place the a_i, successively, between the columns of the array. In the i^{th} row of the table, the first and last columns contain the integer i. A similar set of tables, based on the V_r's, but containing only one row, namely that starting and ending in 1, is constructed. If d, e and f are cosets, and h is a generator, the equations $d\cdot h = f$ and $e\cdot h = f$ imply $d = e$ and $d = f\cdot h^{-1}$. The places of the tables are filled with integers according to the following rules. To start with, the first and last entry of the n^{th} row of each table R_n is taken to be n. Integers are now entered into the table so that if m and p fill consecutive places in a row of one of the tables and the letter a_j straddles the columns containing m and p, then $m \cdot a_j = p$. The places in the tables are assumed to be ordered and a new integer is introduced at the first available place.

Two special situations arise. Closure occurs if, after j and the smaller integers have been entered as often as possible, the first j rows of every table are filled with numbers less than or equal to j. In this case, the process is complete. In the second case, redundancy, entering i and the smaller integers results in the appearance of equations of the form $m \cdot a = r$ and $n \cdot a = r$ or $s \cdot a = n$ and $s \cdot a = m$, with $m \neq n$. In this case m and n represent the same coset. If $m > n$, replace every appearance of m by n and delete the m^{th} row (i.e. the row whose first integer is m). Next, replace every integer w which is greater than n by w-1 wherever it appears in the table. If this produces further redundancies, repeat the process until no further redundancies appear. If this results in closure, discontinue; otherwise, proceed as before until closure results. It has been shown by Mendelsohn (1) that if H is of finite index, closure must occur after some finite number of steps.

2.3 THE TODD-COXETER PROCESS. This process consists of entering a new integer (defining a new coset) at the place where the maximum number of subsidiary equations will result;

or more generally, according to any strategy intended to speed up the process. This procedure is convenient for hand computation but is too vague for machine programming.

2.4 MENDELSOHN'S ALGORITHM. In (1), Mendelsohn formulates the following problem. Find an algorithm for deciding when a word in a group G lies in a subgroup H and a method of expressing this word in terms of the generators of H. More generally, the problem is to find a method of expressing any word in G in terms of the generators of H and a coset representative of H. An algorithmic solution of this problem in terms of the Schreier generators of H is given.

2.5 PERMUTATION REPRESENTATION. When closure is affected, the tables yield a representation of G as a group of permutations of its cosets. If there are n cosets, each generator appears as a permutation of the integers 1, 2, ... ,n.

2.6 COMPUTATIONS. In deriving the tables mentioned in section 2.2, it is important to list the induced equations and note the place where, and the order in which, the induced equations appear. An equation is induced whenever the introduction of a integer completes a row of a table. If a word is given in terms of the generators of G, it can be expressed in terms of the generators of H and a coset representative of H in the following manner. Write down the word. Say it is $a_1 a_2 \ldots\ldots a_m$. Using the permutation representation determine where a_1 sends 1; then where $a_1\ a_2$ send 1; and so forth. Write this in the same form as one of the one line arrays used in deriving the equations. Look at the last induced equation. Does it appear anywhere in the line? If not, consider the second last induced equation and so on. When an induced equation does appear, substitute for the generator of G, its equivalent in terms of the relation in which the induced equation first appeared. Do this for every appearance of the induced equation, cancelling where possible. Then proceed backwards through the table of induced equations. The final result is the required expression.

LEMMA 2.1 Once an induced equation has been used at all occurrences in a line, it cannot appear in subsequent places.

PROOF. If a substitution is to be made involving the "ith" induced equation, the line of the table from which the substitution was made can only involve other induced equations of lower rank. Hence, on making the replacement, only induced equations of lower rank are introduced. Any cancellations which now occur cannot introduce any places where an induced equation of higher rank occurs.

CHAPTER III

THE PROGRAM AND THE RESULTS

3.1 INTRODUCTION. The program is briefly described and the results of two examples shown.

3.2 THE PROGRAM. Input consists of the complete set of equations, the induced equations in the order in which they were derived, the expressions to be used to substitute for a generator and the word to be translated. Up to nine generators of G and H are allowed while the number of cosets is limited to forty-nine and the number of letters in the word must not exceed 999 initially. Intermediate (i.e. after some substitutions have been made) words may safely run over 12,000 letters.

The program proceeds by checking each letter in turn, determining the coset before and after the multiplication by this letter and checking to see if this is the current induced equation. If not, it proceeds to the next letter and so on. If it is the current induced equation, the necessary substitution is made and the program continues with the letter immediately following the substitution. After checking all the letters, all cancellations are made by inserting zero in the place of all cancelling letters. Then, the remaining letters are moved together. The next induced equation is then initialized and the process repeated until no induced equations remain. At the end of the job the final result is typed.

3.3 AN EXAMPLE. Consider the group G generated by A and X subject to the relation $X^2AX=A^2$. Let H be the subgroup generated by $X, Z = A^{-1}X^3A$, and $Y = AX^3A^{-1}$. The permutation representation of the cosets of H is X = (1)(234)(576) and A = (1236475) and the induced equations and their substitutes (in order of derivation) are:

1. $\quad {}_2A_3 \rightarrow A = A^{-1}X^2AX$
2. $\quad {}_4X_2 \rightarrow X = X^{-2}A^{-1}YA$
3. $\quad {}_6A_4 \rightarrow A = A^{-1}X^2AX$
4. $\quad {}_7X_6 \rightarrow X = A^{-1}X^{-2}A^2$
5. $\quad {}_5X_7 \rightarrow X = A^2X^{-1}A^{-1}X^{-1}$
6. $\quad {}_6X_5 \rightarrow X = X^{-2}AZA^{-1}$
7. $\quad {}_7A_5 \rightarrow A = A^{-1}X^2AX$

The problem was to express $AXAX^2A^2$ in terms of X, Y and Z. The solution is $YX^2YZ(Y^{-1}X^{-2})^2Y^2X^2YZ$

3.4 A SECOND EXAMPLE. Consider the group G generated by A and X subject to the relation $X^3AX = A^2$. Let H be the subgroup of G generated by A, $Y = X^{-1}A^3X$ and $Z = XA^3X^{-1}$. The permutation representation of the cosets of H is X = (1,2,3,4,10,8,11,6,12, 7,9,5) and A = (1)(456)(3)(278)(9)(10)(11)(12) and the induced equations, in order, and their substitutes are as in table I.

The problem was to take the Schreier generators of H expressed in terms of A and X and express them in terms of A, Y and Z. The generators in their two forms are presented in table II.

TABLE I

THE INDUCED EQUATIONS AND THEIR SUBSTITUTES

Number	Equation	Generator Replaced	Substitute
1	${}_4A_5$	A	$X^{-3}A^2X^{-1}$
2	${}_6A_4$	A	$A^{-1}XYX^{-1}A^{-1}$
3	${}_3A_3$	A	$X^{-3}A^2X^{-1}$
4	${}_8A_2$	A	$A^{-2}X^{-1}ZX$
5	${}_9X_5$	X	$X^{-1}A^2X^{-1}A^{-1}X^{-1}$
6	${}_{10}X_8$	X	$X^{-2}A^2X^{-1}A^{-1}$
7	${}_{10}A_{10}$	A	$X^{-3}A^2X^{-1}$
8	${}_{11}X_6$	X	$X^{-2}A^2X^{-1}A^{-1}$
9	${}_{11}A_{11}$	A	$X^{-3}A^2X^{-1}$
10	${}_{12}X_7$	X	$X^{-2}A^2X^{-1}A^{-1}$
11	${}_{12}A_{12}$	A	$X^{-3}A^2X^{-1}$
12	${}_9A_9$	A	$X^{-3}A^2X^{-1}$

TABLE II

THE TWO FORMS OF THE SCHREIER GENERATORS

In Terms of the Generators of G	In Terms of the Generators of H
AX^{-12}	$A^2((A^2Y^{-1})^2Z)^2(A^2Y^{-1})^3A^{-2}Z(A^2Y^{-1})^2Z(A^2Y^{-1})^3A^{-2}$
XAX^{-9}	$(Z(A^2Y^{-1})^2)^3A^2Y^{-1}A^{-2}Z(A^2Y^{-1})^2Z(A^2Y^{-1})^3A^{-2}$
X^2AX^{-2}	YA^{-2}
X^3AX^{-11}	$A^3((A^2Y^{-1})^2Z)^2(A^2Y^{-1})^3A^{-2}(Z(A^2Y^{-1})^2)^2A^2Y^{-1}A^{-2}$
X^4AX^{-4}	$ZA^2Y^{-1}A^2Y^{-1}$
X^5AX^{-1}	$YA^{-2}YA^{-2}$
X^6AX^{-6}	$A^2(YA^{-2})^3Z^{-1}(YA^{-2})^2Z^{-1}$
X^7AX^{-3}	$(Z(A^2Y^{-1})^2)^2$
X^8AX^{-8}	$(YA^{-2})^2Z((A^2Y^{-1})^2Z)^2(A^2Y^{-1})^3A^{-2}Z(A^2Y^{-1})^2Z(A^2Y^{-1})^3A^{-2}$
X^9AX^{-5}	$(A^2Y)(A^{-2}Y)^2A^{-2}Z^{-1}(YA^{-2})^2Z^{-1}A^2Y(A^{-2}Y)^2A^{-2}Z^{-1}(YA^{-2})^2Y^{-1}$
$X^{10}AX^{-10}$	$(Z(A^2Y^{-1})^2)^2A^3(A^2Y^{-1})^2((Z(A^2Y^{-1})^2)^2A^2Y^{-1}A^{-2})^2$
$X^{11}AX^{-7}$	$(A^2(YA^{-2})^3Z^{-1}(YA^{-2})^2Z^{-1})^2(YA^{-2})^2A(YA^{-2})^3Z^{-1}(YA^{-2})^2Z^{-1}$
X^{12}	$(A^2(YA^{-2})^3Z^{-1}(YA^{-2})^2Z^{-1})^2(YA^{-2})^2A^{-1}$

BIBLIOGRAPHY

1. N. S. Mendelsohn, An Algorithmic Solution for a Word Problem in Group Theory, The Canadian Journal of Mathematics 16 (1964), 509-516.

2. Marshall Hall, The Theory of Groups, (New York, 1959), pp 94-106.

3. H. S. M. Coxeter and W. O. J. Moser, Generators and Relations for Discrete Groups, Proc. Glasgow Math. Assoc., 5, Part 4, 166-175(1962).

4. John Leech, Coset Enumeration on Digital Computers, Proc. Camb. Phil. Soc., 59 (1963), 257-267.

5. Some Definitions of Klein's Simple Group of Order 168 and other Groups, Proc. Glasgow Math. Assoc., 5, Part 4, 166-175 (1962).

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.

Introduction

In 1963, Mendelsohn presented an algorithmic solution for a word problem in Group Theory. The method is based on the Schreier process and the Todd-Coxeter process for coset enumeration. For even the simplest examples, the method is laborious and Leech & Trotter have presented machine programs which will enumerate the cosets. This machine program will express a word expressed in the generators of the group as a word in the generators of a subgroup times a coset representative, given the coset enumeration. If this process is applied to the Schreier generators of the subgroup, a table results which enables one to carry out hand computations on word problems in a reasonable time.

GENERAL

The program falls into four distinct parts. The initialization routines read the control and input cards and set up the tables and control data in core storage. The linkage detects situations where a substitution must be made and makes the substitutions. It also controls the branches to and from the last two routines. The cancellation routine detects and deletes inverses. It then squeezes the data together, eliminating the spaces filled by the cancelled digits. The end of job routine types the final result on the consol.

INITIALIZATION

This section of the program stores the tables from the card input and initializes the various implicit indexing locations and constants so that the linkage is free from any consideration other than a test for end of job of levels.

The purpose of this page is to begin setting up the constants needed to control the input and the program linkage.

1. A card is read. Card type must be one or the program halts.

2. The number one less than the lowest assigned to a new generator is stored in CØN for use in the linkage (page L1).

3. The length of a complete set of permutation representations of the cosets is calculated (the number of original generators times the area needed for one generator's expression).

4. BASEI is set to the start address of the inverse permutation representation area. (BASE, the start of the direct permutation representation area, plus the length computed in step 2).

5. ECRT is set to the start address of the relation test (critical situations) area. (BASEI plus the length computed in step 2) This will be used on page I4.

6. The length of a permutation expression is stored in CSTLN for use in calculating the t_o coset (page L1)

7. CURCRT is set to one less than the start address of the substitution relation area. (ECRT less one plus five times the number of levels)

8. Set CURCTL to the start address of the substitution relation area (this will be used on page I5, step 3).

9. Set CCRTCL to the last address of the direct substitution relation table (the start address of the substitution relation table plus the number of relations (levels) times the maximum length of a substitution area).

10. Set ICRCTL to the start address of the inverse substitution area (the number in step 9 plus one to allow for the record mark).

11. Set ABGN to the start of the area which will contain the relation to be translated (ICRCTL plus one plus the number of levels times the maximum length of a substitution area).

12. Proceed to page I2, step 1.

DIAGRAMMING AND CHARTING SHEET

RUN DESCRIPTION	PREPARED BY	PAGE I 1 OF
ROUTINE DESCRIPTION	CHECKED BY	DATE FINALIZED