Can. J. Math., Vol. XXVI, No. 4, 1974, pp. 769–782

DETERMINING SUBGROUPS OF A GIVEN FINITE INDEX IN A FINITELY PRESENTED GROUP

ANKE DIETZE AND MARY SCHAPS

1. Introduction. The use of computers to investigate groups has mainly been restricted to finite groups. In this work, a method is given for finding all subgroups of finite index in a given group, which works equally well for finite and for infinite groups. The basic object of study is the finite set of cosets. §2 reviews briefly the representation of a subgroup by permutations of its cosets, introduces the concept of normal coset numbering, due independently to M. Schaps and C. Sims, and describes a version of the Todd-Coxeter algorithm. §3 contains a version due to A. Dietze of a process which was communicated to J. Neubüser by C. Sims, as well as a proof that the process solves the problem stated in the title. A second such process, developed independently by M. Schaps, is described in §4. §5 gives a method for classifying the subgroups by conjugacy, and §6, a suggestion for generalization of the methods to permutation and matrix groups.

This paper concentrates on the theoretical underpinnings of the process rather than the execution, in order to indicate where basic adaptations to a more limited problem could be made with a significant saving in computation time. For example, the authors conjecture that in general the first method, which requires considerably more storage space because it retains more information about the "word problem" for the group at each step, is more efficient; however, for groups in which the "word problem" has a simple solution, an appropriate version of the second method should be faster. Although a working program solving the general problem is available in ALGOL from A. Dietze, it is slow. Improvements suggested by C. Sims which would greatly increase efficiency have been mentioned in this paper even though they have not been tested in practice. In the present form of the program, calculation time for index 10 in various R^2 crystal groups ranged from 1 to 20 minutes, with a slow exponential increase in higher indices.

2. Theoretical introduction. Let G be a group and let U be a subgroup of finite index. Set $n = [G:U]$. Let $P_U = \{Uh_1 = U, Uh_2, \ldots, Uh_n\}$ be the set of right cosets of U. For any $g \in G$, let $g\varphi_U$ be the bijection from P_U to P_U given by

$$g\varphi_U : Uh_i \to Uh_ig.$$

Received March 7, 1972.

φ_U is a canonical homomorphism from G onto a transitive subgroup of S_{P_U}, the symmetric group on P_U. $U\varphi_U$ is the subgroup $\mathrm{St}_{G\varphi}(U)$, the stabilizer of U in $G\varphi$.

Let ν be a bijective mapping of P_U onto the set $\{1, 2, \ldots, n\}$ such that $U\nu = 1$. Such a ν will be called a coset numbering. ν induces a non-canonical isomorphism of S_{P_U} with S_n, the symmetric group of degree n. The $(n-1)!$ distinct coset numberings will define $(n-1)!$ distinct homomorphisms of G onto a transitive permutation group H of S_n. Any such homomorphism $\varphi_{U,\nu}: G \to S_n$ maps U onto $\mathrm{St}_{S_n}(1) \cap H$, which is just $\mathrm{St}_H(1)$. In fact, the homomorphisms $\varphi_{U,\nu}$ exhaust the set of $\varphi: G \to S_n$ such that $G\varphi$ is transitive and $U\varphi = \mathrm{St}_{G\varphi}(1)$. Given such a φ, we define the appropriate ν by

$$\nu: Uh_i \to (1)h_i\varphi.$$

ν is well-defined because, for any $u \in U$,

$$(1)(uh_i)\varphi = ((1)u\varphi)h_i\varphi = (1)h_i\varphi.$$

Assume G is finitely presented. Thus $G = \langle g_1, \ldots, g_r | r_j(g_1, \ldots, g_r) = 1, \text{ for } 1 \leqq j \leqq s \rangle$, where the r_j are words in the generators g_i and their inverses $g_{i+r} = g_i^{-1}$. Henceforth, the definitions will depend on this ordering of the generators.

Let ν be a coset numbering of a subgroup U of finite index, and let K_i be the coset such that $K_i\nu = i$, for $1 \leqq i \leqq n = [G:U]$.

Definition 2.1. The coset table of U with respect to ν is the matrix $T = ((K_ig_k)\nu)$, $1 \leqq i \leqq n$, and $1 \leqq k \leqq 2r$.

Definition 2.2. The pairs (i, k) describing positions in the coset table are ordered by setting $(i', k') < (i, k)$ if $i' < i$, or $i' = i$ and $k' < k$.

Definition 2.3. A coset numbering ν is called a normal numbering if for every pair (i, k) and every coset number j,

$$(K_{i'}g_{k'})\nu \leqq j$$

for all $(i', k') < (i, k)$ implies that $(K_ig_k)\nu \leqq j + 1$. The significance of this definition lies in the following basic result.

LEMMA 2.1. *For every subgroup U of finite index in G there is exactly one normal numbering.*

Proof. Let ν and ν' be two numberings satisfying the definition above, and let K_i and K_i' be the cosets determined by the equations $K_i\nu = K_i'\nu' = i$, for all i with $1 \leqq i \leqq [G:U]$. We may proceed by induction, since $K_1 = K_1' = U$. Suppose $K_i = K_i'$ for $i \leqq j < [G:U]$. Consider the sequence of cosets

$$\begin{array}{ccc} K_1g_1, & \ldots, & K_1g_{2r}, \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ K_jg_1, & \ldots, & K_jg_{2r}. \end{array}$$

Let K_ig_k be the first such coset not in the set $K_1, \ldots, K_j$. Then $(K_ig_k)\nu \leqq j+1$, and $(K_i'g_k)\nu' \leqq j+1$, so $K_{j+1} = K_{j+1}' = K_ig_k$.

Let ν be the normal numbering of P_U and let $\varphi = \varphi_{U,\nu}$ be the corresponding homomorphism $\varphi: G \to S_n$. Let $H = G\varphi$, and set $t_k = g_k\varphi \in H$, $1 \leqq k \leqq 2r$. The permutations $t_1, \ldots, t_{2r}$ have the following properties:

(P1) $r_j(t_1, \ldots, t_r) = 1$, $1 \leqq j \leqq s$;

(P2) for every $j \geqq 2$, there is an $i < j$ such that $it_k = j$ for some k.

Definition 2.4. Let σ be the mapping with domain $\{2, 3, \ldots, n\}$ defined by setting $j\sigma = (i, k)$, where this is the first pair such that $it_k = j$.

(P3) $j < j'$ implies that $j\sigma < j'\sigma$ for $2 \leqq j \leqq n$.

Conversely, suppose an ordered set of permutations $t_1, \ldots, t_{2r}$ is given in S_n, satisfying (P1) – (P3), where $t_{k+r} = t_k^{-1}$. Then by (P1) the mapping $g_k \mapsto t_k$ induces a homomorphism $\varphi: G \to S_n$. By (P2), the image $H = G\varphi$ is transitive, and thus the subgroup $\mathrm{St}_H(1)$ is of index n in G. Its inverse image U must be of index n in G. Then $\varphi = \varphi_{U,\nu}$, where ν is the numbering $(Uh)\nu = (1)h\varphi$. By (P2) and (P3), ν is a normal numbering.

Definition 2.5. A partial coset table R_z is a matrix (it_k^z), where

$$t_k^z: \{1, \ldots, N\} \to \{0, 1, \ldots, N\}$$

is one-to-one except into 0. $N > n$ is a fixed number, which may also depend on G. (In the existing computer program, N is fixed at the minimal value of $n+1$, thus reducing storage. However, with such small N little information is gained from the longer relations at first, thus reducing efficiency.)

Definition 2.6. R_z is in normal order if

(i) $it_k^z = j > 0$ implies that $jt_{k\pm r}^z = i$.

(ii) There is an m_z such that all rows below the mth are zero, and the t_k^z are transitive on $1, \ldots, m_z$.

(iii) If σ is defined as before on $\{2, \ldots, m_z\}$, then $j\sigma < j'\sigma$ when $j < j'$, and $it_k^z \neq 0$ for all $(i, k) \leqq m_z\sigma$.

Let R be a partial coset table, with columns corresponding to m cosets $K_1, \ldots, K_m$ and $2r$ rows corresponding to the generators $g_1, \ldots, g_r$ and their inverses $g_{r+1}, \ldots, g_{2r}$. An entry of j in position (i, k) would mean that

$$K_ig_k = K_j.$$

If this position is blank, it means that we do not yet know which coset equals K_ig_k. Since $K_1, \ldots, K_m$ is generally not a complete list of cosets, K_ig_k may not even be among them.

$K_1, \ldots, K_m$ can be defined inductively from the σ mapping of R, by setting $K_1 = U$, and $K_j = K_ig_k$, where $j\sigma = (i, k)$. Similarly, taking $y_1 = e$, the identity element, we can inductively define coset representatives $y_1, \ldots, y_m$, such that $K_i = Uy_i$.

The cosets $K_1, \ldots, K_m$ may not all be distinct. The main task in the Todd-Coxeter algorithm is to discover for which pairs i and j we have $K_i = K_j$. This will alternate with a procedure for defining new cosets in normal order. We assume U to be given by a set of generators in the finitely presented group G. The goal of the algorithm is to find a non-redundant set of cosets $K_1, \ldots, K_n$ which is a complete listing of all cosets of U, thus showing that $[G:U] = n$.

The procedure for defining a new coset, given a partial coset table R as above, is simply to take the first blank position (i, k) and set

$$K_{m+1} = K_i g_k$$
$$K_i = K_{m+1} g_k^{-1},$$

with the appropriate entries in the coset table. The mapping is extended by setting $(m + 1)\sigma = (i, k)$.

The difference between this procedure for defining cosets and the inductive procedure given earlier for numbering cosets underscores the difference between the practical and the theoretical approach to a problem in group theory. In the theoretical discussion we assumed that we could always tell when two cosets were identical, and could thus choose the first coset in the sequence

$$\begin{array}{ccc} K_1 g_1, \ldots, & K_1 g_{2r}, \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ K_j g_1, \ldots, & K_j g_{2r}, \end{array}$$

which did not belong to the set $K_1, \ldots, K_j$. In practice, however, far from being able to determine when two cosets are identical, we cannot even determine when two words in the generators $g_1, \ldots, g_{2r}$ represent the same element of the group G. This is the famous "word problem," which has no solution for arbitrary infinite groups.

Since the cosets $K_1, \ldots, K_m$ are not necessarily distinct, the algorithm contains a procedure which searches for identities between cosets. A new book-keeping device is needed, a list of entries in the partial coset table in the order in which they were made. Since we begin with no more information about the cosets than that $K_1 = U$, the coset table is initially blank, as is the list.

We want to insure that each entry in a coset table R is compatible with the following requirements:

(i) for each of the given generators u of U, $u \in U$;

(ii) for each relation r_j, $r_j(g_1, \ldots, g_r) = e$. Given an entry in R, determined by an equation $it_k^{\varepsilon} = j$, we apply the following test procedure: Take the first subgroup generator U which contains g_k, and look at its image $u\varphi$, represented as a product of the generators $t_1, \ldots, t_{2r}$ of $G\varphi$. Thus $u\varphi = t_{k_1} t_{k_2} \ldots t_{k_v}$. In order to avoid having $u \notin U$, we must avoid having $(1)u\varphi \neq 1$, since $U\varphi = \mathrm{St}_{G\varphi}(1)$. Starting with $i_1 = 1$, let $i_2 = i_1 t_{k_1}$, and so forth, until either i_{v+1} is

defined or some $i_q t_{k_q} = 0$ is reached. Similarly, work backwards from $i_{v+1} = 1$.

$$i_1 \xrightarrow{t_{k_1}} i_2 \xrightarrow{t_{k_2}} \cdots \xrightarrow{t_{k_{v-1}}} i_v \xrightarrow{t_{k_v}} i_{v+1}.$$

If the two chains do not meet, or if they overlap and coincide, then no new information is provided by the test. If they just meet, with i_q defined from the front and i_{q+1} defined from the back, we will represent this by

$$\xrightarrow{t_{k_{q-1}}} i_q \xrightarrow{t_{k_q}} [i_{q+1}] \xrightarrow{t_{k_{q+1}}}$$

and we will make a new entry $i_q t_{k_q} = i_{q+1}$. Finally, if the two chains overlap and do not coincide, we will have a situation

$$\xrightarrow{t_{k_{q-1}}} i_q \neq i_q' \xrightarrow{t_{k_q}} i_{q+1}.$$

This shows that $K_{i_q} = K_{i_q'}$, and requires that we make an identification $i_q := :i_q'$. Let us describe in general the procedure for making an identification $i := :j$, where i and j are coset numbers with $i < j$. Since in this case $i\sigma < j\sigma$, we must replace j by i in order to preserve normal order in the partial coset table. We transfer all the information in the jth row to the ith row, using the $2r$ equations, $K_i g_1 = K_j g_1, \ldots, K_i g_{2r} = K_j g_{2r}$. In the process we will make new entries in the ith row, whenever a previously blank position is filled with an entry from the jth row and whenever a j is replaced by an i. We may also discover new identifications, when $K_i g_l$ and $K_j g_l$ are both known and have different numbers. The identifications cannot all be made simultaneously, so they are listed and taken up in turn. When all necessary identifications have been made, the identification $i := :j$ is completed, and R_z is transformed into a table R_{z+1}, with $m_{z+1} < m_z$.

Returning to the specific situation, assume that we have just finished testing the entry $it_k^z = j$ in the first subgroup generator containing g_k. We then go on to the next, and continue through the generators of U. Next come the relations r_l, $1 \leqq l \leqq s$. We write $r_l(g_1, \ldots, g_r)$ in the form $(a_1 \ldots a_p)^\alpha$, where this representation is minimal in the sense that $(a_1 \ldots a_p)$ is not the power of any smaller segment. For each occurrence of g_k as some a_q, we can replace r_l by its conjugate $(a_q \ldots a_p a_1 \ldots a_{q-1})^\alpha$. Taking each relation in turn, we start with the first occurrence in the relation, write the appropriate conjugate in the form $t_{k_1} t_{k_2} \ldots t_{k_v}$, and set $i_1 = i_{v+1} = i$ if $t_{k_1} = t_k$, or $i_1 = i_{v+1} = j$ if $t_{k_1} = t_k^{-1}$. We then proceed exactly as for one of the subgroup generators. Whenever the list of entries to be tested is exhausted, we define a new coset and test that entry. Eventually, if U really is of finite index, the blanks in the table will all be filled in, and we say that the table is closed. The index of U in G is given by the number of rows in the coset table.

As an example of the Todd-Coxeter algorithm, let G be the quaternion group, of order 8, generated by I and J with relations

$$I^4 = J^4 = e,\ IJI = J, \text{ and } I^2 = J^2.$$

Let U be the subgroup generated by J. In setting up the coset table we have $t_1 = I, t_2 = J, t_3 = I^{-1}$, and $t_4 = J^{-1}$. The first new coset is defined by $1t_1 = 2$.

$$1 \xrightarrow{t_1} 2 \xrightarrow{t_2} [\ ]^{t_1}\ [\ ] \xrightarrow{t_4} 1$$

$$1 \xrightarrow{t_1} 2 \xrightarrow{t_4} [\ ]^{t_1}\ [\ ] \xrightarrow{t_2} 1$$

$$1 \xrightarrow{t_1} 2 \xrightarrow{t_1} [\ ]^{t_4}\ [\ ] \xrightarrow{t_4} 1$$

$$1 \xrightarrow{t_1} 2 \xrightarrow{t_4} [\ ]^{t_4}\ [\ ] \xrightarrow{t_1} 1, \text{ etc.}$$

Thus the test yields no new information. Next we set $1t_2 = 3$.

$$1 \xrightarrow{t_2} 3 \neq 1.$$

This gives an identification $1 := :3$. After this identification the coset table is

	t_1	t_2	t_3	t_4
1	2	1	0	1
2	0	0	1	0.

The entry we must now test is $1t_2 = 1$.

$$1 \xrightarrow{t_2} 1$$

$$1 \xrightarrow{t_2} 1 \xrightarrow{t_1} 2 \xrightarrow{t_4} [\ ] \xrightarrow{t_1} 1$$

$$1 \xrightarrow{t_4} 1 \xrightarrow{t_1} 2 \xrightarrow{t_2} [\ ] \xrightarrow{t_1} 1$$

$$1 \xrightarrow{t_4} 1 \xrightarrow{t_4} 1 \xrightarrow{t_1} 2 \xrightarrow{t_1} [1].$$

Thus $2t_1 = 1$.

$$1 \xrightarrow{t_4} 1 \xrightarrow{t_1} 2 \xrightarrow{t_1} 1 \xrightarrow{t_4} 1$$

$$1 \xrightarrow{t_1} 2 \xrightarrow{t_1} 1 \xrightarrow{t_1} 2 \xrightarrow{t_1} 1.$$

This completes the test of $1t_2 = 1$. We must now test $2t_1 = 1$.

$$2 \xrightarrow{t_1} 1 \xrightarrow{t_2} 1 \xrightarrow{t_1} 2 \xrightarrow{t_4} [2]$$

Thus $2t_2 = 2$. This closes the table, so $[G:U] = 2$.

	t_1	t_2	t_3	t_4
1	2	1	2	1
2	1	2	1	2.

Since in our problem we are not given generators of a subgroup, but are in fact searching for appropriate sets of subgroup generators, we will be continu-

https://doi.org/10.4153/CJM-1974-072-0 Published online by Cambridge University Press

ally employing a limited version of the Todd-Coxeter algorithm. A list is made of all entries $it_k^z = j$, $1 \leqq k \leqq r$, which have not yet been tested in the relations. Each such entry is tested in each significant place in the relations. All necessary entries which are discovered are added to the coset table and to the list of entries to be tested. All necessary identifications $i := j$, $i < j$, are made immediately, as described earlier. Resulting entries in the coset table are added to the list. At the beginning, or whenever the list of untested entries is exhausted and $m_z < N$, a new coset is defined as follows: it_k^{z+1} is set equal to $m_z + 1$, where (i, k) is the first pair such that $it_k^z = 0$. If none exist, the table is closed. Once the inverse entry in $t_{k\pm r}^z$ has been made, a new table R_{z+1} has been created from R_z, with $m_{z+1} = m_z + 1$. When $m_z = N$, and the list of untested entries is exhausted, the algorithm is halted, and the main process continues.

The procedure just described preserves normal ordering. Consider, for example, an identification $i := j$. The requirement that $it_k^z \neq 0$ for $(i, k) \leqq m_z\sigma$ insures that no such entries it_k^z will be raised. Furthermore, the entry in a position $i'\sigma$ will be lowered only if i' is identified with a lower number. Thus the σ mapping, which determines normal numbering, will be subjected only to deletions, consolidations, and additions onto the end.

3. Method A for determining all subgroups of index n in G. The basic procedure used in searching for subgroups of index n is to start with the identity subgroup, and enlarge it one generator at a time until the resulting subgroup is of index less than or equal to n. At this point, the last one or more generators are removed, so that one again has a subgroup of index greater than n, and they are replaced by different elements.

More explicitly, given the subgroup $U = \langle e \rangle$, the Todd-Coxeter process described above will generate cosets Uy_1, Uy_2, . . . , Uy_N, which at this stage each contain only a single element. We intend to enlarge U so that there are only n distinct cosets. In the process, at least two of the cosets $Uy_1, \ldots, Uy_{n+1}$ must merge, by the pigeonhole principle. We will consider all the various possibilities, starting with $Uy_1 = Uy_2$, continuing with $Uy_1 = Uy_3$, through $Uy_n = Uy_{n+1}$. Note that $Uy_i = Uy_j$ if and only if $y_iy_j^{-1} \in U$. The subgroups are built up by generators of this form. The technical problem of insuring that each subgroup is generated exactly once is solved by using the normal numbering of cosets, based on a denumeration of the elements of the group, to give an ordering of the subgroup generators $y_iy_j^{-1}$ according to the ordering of the elements y_j, each of which will occur only once in the set of generators for each subgroup. The example worked at the end of the section may be of assistance in understanding the procedure.

Assume $n > 1$. The Todd-Coxeter process is started. If the table closes after $m \leqq N$ cosets have been defined, the order of G is m, and the process need not continue unless n divides m properly. Assume that we have a partial coset table R_z, at which the Todd-Coxeter algorithm stopped.

https://doi.org/10.4153/CJM-1974-072-0 Published online by Cambridge University Press

Definition 3.1. At this stage the branch point of the 0th level is reached. It is assigned the marker 2.

More generally, suppose a branch point of the kth level with marker d is reached. Then the identifications $i:=:j$, with $d \leqq j \leqq n+1$, $1 \leqq i < j$, designate the different possible branches, which are to be taken up in the following order: $i:=:j$ comes before $i':=:j'$ when either $j < j'$ or $j = j'$ and $i < i'$. Any identification resulting directly from $i:=:j$, that is, before testing in the relations, is of the form $iw:=:jw$, where w is a product of the mappings t_k^z. If $jw \geqq j$, it may be eliminated from the list of remaining alternate branches by virtue of (C1) below. After taking one branch, by making one of the identifications $i:=:j$, the Todd-Coxeter algorithm is entered as if this had been a necessary identification. The following cases can occur:

(C1) A necessary identification $f:=:g$ appears, with $g < j$. Then this branch is abandoned, and the next highest branch is taken at the last branch point where an alternative remains to be taken. If no alternatives remain, the process is finished.

(C2) The coset table either does not close, after N cosets have been defined and tested, or it closes with m rows, where m is a proper multiple of n, and the situation in (C1) does not occur.

Definition 3.2. In this case a branch point of the $(k+1)$st level is reached. Corresponding to the identification $i:=:j$ which led to it, this branch point will be assigned

(i) the marker j

(ii) the group element $u = y_i y_j^{-1}$, where $y_1, \ldots, y_{m_z}$ are the set of coset representatives constructed inductively from R_z by setting $y_1 = 1$, and $y_{j'} = y_{i'} g_{k'}$, where $j'\sigma = (i', k')$.

The process continues as above from this new branch point.

(C3) The coset table closes after m rows, and (C1), (C2) do not occur.

Definition 3.3. In this case, an endpoint of the $(k+1)$st level is reached. It corresponds to an element $u = y_i y_j^{-1}$ as in (C2). If $m = n$, this endpoint determines one of the desired subgroups, $U = \langle u_1, \ldots, u_{k+1} \rangle$, where the u_p for $p = 1, \ldots, k+1$, correspond to the branch points $V_1, \ldots, V_{k+1}$ in the chain leading to this endpoint. These u_p clearly generate the subgroup, because its coset table can be reconstructed from them by a regular Todd-Coxeter algorithm. After evaluating the endpoint, one continues as in (C1). The process is finished when all possible branches at the 0th level are exhausted.

Remark. If all subgroups of index $m \leqq n$ are desired, the only change which must be made is to evaluate each endpoint with $m \leqq n$, and then treat it as a branch point.

This completes the inductive description of the process. We now show that it does in fact give each subgroup exactly once.

LEMMA 3.1. *The process is finite and single valued.*

Proof. Since there are only finitely many possible coset tables, and the process follows a directed, connected graph with a unique minimal point, it is sufficient to show that different branches $i:=:j$ and $i':=:j'$ at one branch point never lead to the same table at points of higher level. Assume $j < j'$, or $j = j'$ and $i < i'$. Clearly at branch points stemming from these branches the marker will always be at least j; thus the number in positions $(i'', k'') < j\sigma$ are fixed, since no identifications can occur affecting $j'' < j$, by (C1).

(a) If $j = j'$, the $j\sigma$ position is thereafter fixed at i in one branch and i' in the other.

(b) If $j < j'$, then the $j\sigma$ position is thereafter fixed at i or j, respectively.

LEMMA 3.2. *Let V_p be a branch point of level p with marker d, and let T_p be its coset table, with transitivity mapping σ_p. Let T, with mapping σ, be a different table, closed with $m \leqq n$ rows. If $p > 0$, suppose that T agrees with T_p in all positions $(i', k') \leqq d\sigma_{p-1}$. Then there is a d', with $d \leqq d' \leqq m + 1$, such that they agree in all positions $(i', k') < d'\sigma_p$ and disagree there.*

Proof. Let (i, k) be the first position in which they disagree. If $p = 0$, then $2\sigma_0 = (1, 1)$; if $(i, k) = (1, 1)$, we can set $d' = 2$, so we may assume that there is a $d' > 2$ with

$$(d' - 1)\sigma_0 < (i, k) \leqq d'\sigma_0.$$

There is one circumstance in which $2\sigma_0 > (1, 1)$, when $g_1 = e$, but in that case we would simply eliminate this generator. Since T is closed, $d' \leqq m + 1$. Then in fact, $(i, k) = d'\sigma_0$, since the lower entries $i't_{k'} = j$ can be constructed from the equations $1(y_{i''}\varphi) = i''$, for $i'' = 2, 3, \ldots, d' - 1$, and from $(i'')r_l\varphi = i''$, for $i'' \leqq N$. The coset representatives y_i'' agree for T and T_0, in the given range of cosets. This completes the first case in the lemma. For $p > 0$, we choose $d' \geqq d$ such that

$$(d' - 1)\sigma_p < (i, k) \leqq d'\sigma_p$$

The entries lower than $d'\sigma_p$ can be constructed from the following sets of equations:

$$(1)y_{i''}\varphi = i'', \text{ for all } i'' < d',$$
$$(i'')r_l\varphi = i'', \text{ for } 1 \leqq i'' \leqq N,\ 1 \leqq l \leqq s,$$
$$(1)u_t\varphi = 1, \text{ for } 1 \leqq t \leqq p.$$

Here u_t is the element corresponding to V_t in the unique chain of branch points leading to V_p. Clearly these equations hold for T as well, since the $y_{i''}$ are the same, and each $u_t = y_{i''}{}^t(y_{j''}{}^t)^{-1}$, where these coset representatives are computed from the table for V_t. By the definition of the marker, d, we must have $i'' < j'' \leqq d$, so that $(1)y_{i''}{}^t$ and $(1)y_{j''}{}^t$ are determined by numbers in positions less than $d\sigma_{t-1} \leqq d\sigma_{p-1}$. Thus again, $(i, k) = d'\sigma_p$.

THEOREM *The process produces each subgroup of index n exactly once.*

Proof. Each subgroup U corresponds to a unique coset table T in normal form, so by Lemma 3.1, it is sufficient to construct a chain of branch points $V_0, \ldots, V_q$ such that $T = T_q$. Proceeding by induction, given $V_0, \ldots, V_p$, find d' as in Lemma 3.2. If j is the number in $d'\sigma$ in T, take the branch $j{:=}{:}d'$ at V_p to get V_{p+1}.

As an example, let us take the dihedral group D_4. We will use the ordinary presentation, by two generators A and B with relations $A^2 = B^4 = e$ and $ABA = B^{-1}$, the latter of which will be used in the form $(AB)^2 = e$. D_4 is a finite group of order 8, with five elements of order 2, B^2, A, AB, AB^2, and AB^{-1}. There are three subgroups of order 4, generated by $\langle B\rangle$, $\langle A, B^2\rangle$, and $\langle AB, B^2\rangle$. Method A will be used to find these subgroups, with $n = 2$ and $N = 4$.

The Todd-Coxeter algorithm initially leads to the branch point V_0, at which stage $U = \langle e\rangle$. Letting $t_1 = A$, $t_2 = B$, $t_3 = A^{-1}$, and $t_4 = B^{-1}$, the table for V_0 is

	t_1	t_2	t_3	t_4	
1	2	3	2	4	
2	1	0	1	0	$2\sigma = (1, 1)$
3	0	0	0	1	$3\sigma = (1, 2)$
4	0	1	0	0	$4\sigma = (1, 4)$.

The coset representatives are $y_1 = e$, $y_2 = A$, $y_3 = B$, and $y_4 = B^{-1}$.

From V_0 there are three possible branches, $1{:=}{:}\,2$, $1{:=}{:}3$, and $2{:=}{:}3$. The first branch, $1{:} = {:}2$, corresponds to $U = \langle u_1\rangle$, with $u_1 = y_1y_2^{-1} = A^{-1}$. After making the identification, we arrive at the table

	t_1	t_2	t_3	t_4
1	1	2	1	3
2	0	0	0	1
3	0	1	0	0

.

There is one entry, $1t_1 = 1$, to be tested.

$$1 \xrightarrow{t_1} 1 \xrightarrow{t_1} 1$$

$$1 \xrightarrow{t_1} 1 \xrightarrow{t_2} 2 \xrightarrow{t_1} [3] \xrightarrow{t_2} 1.$$

Thus $2t_1 = 3$. Testing this entry yields the information that $3t_1 = 2$, and testing this uncovers nothing new. Since the entries to be tested are now exhausted, we define a new coset, by setting $2t_2 = 4$, and $4\sigma = (2, 2)$. Testing

https://doi.org/10.4153/CJM-1974-072-0 Published online by Cambridge University Press

uncovers two new entries, $4t_2 = 3$, and $4t_1 = 4$, yielding a closed table

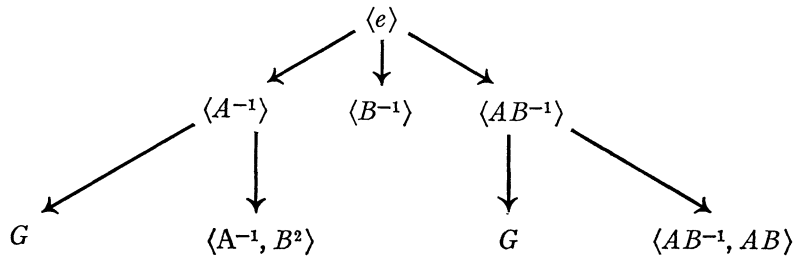
	t_1	t_2	t_3	t_4
1	1	2	1	3
2	3	4	3	1
3	2	1	2	4
4	4	3	4	2

.

Testing the entries uncovers no new identifications. We have thus reached a branch-point V_1, with marker 2. It corresponds to the subgroup $U = \langle A^{-1} \rangle$. The new coset representatives are $y_1 = e$, $y_2 = B$, $y_3 = B^{-1}$, and $y_4 = B^2$. Once again three branches are available, $1: = :2$, $1: = :3$, and $2: = :3$. Performing the identification $1: = :2$ implies immediately that $1: = :3$ and $2: = :4$, so the resulting subgroup, $U_2 = \langle A^{-1}, B^{-1} \rangle$, is identical to the group. The branch $1: = :3$ is equivalent to this branch, so the only one remaining is $2: = :3$. This identification leads immediately to an end-point V_2, corresponding to the subgroup $U_2 = \langle A^{-1}, B^2 \rangle$ of index 2.

Having exhausted the branch $1: = :2$ from V_0, we take the next branch $1: = :3$, corresponding to the subgroup $U_1 = B^{-1}$. The identification $1: = :3$ also entails $1: = :4$, The only new entry is $1t_2 = 1$; in testing this we find that $2t_2 = 2$, which closes the table. This gives an end-point, corresponding to the subgroup $\langle B^{-1} \rangle$, of index 2.

The third branch at V_0, $2: = :3$, is almost identical to the first, except that there are only two branches from the first branch point, because it has marker 3. This branch yields the subgroup $\langle AB^{-1}, AB \rangle$, of index 2. The complete graph of this example is as follows:



4. Method B for determining all subgroups of given index n in G. Let G be as before. In this alternative method, a coset table is set up with only n rows, which may not be identified. Instead, it is filled by making entries in individual positions. We thus require at the outset that $K_1, \ldots, K_n$ represent a complete, non-redundant collection of cosets.

At the unique branch point of the 0th level, V_0, the coset table is blank. The branches are the possible entries in the first blank position, (1, 1), the choices being 2 and, if $r > 1$, 1. As at every subsequent branch point, the branches, that is, the possible entries in the first blank position, are taken in

https://doi.org/10.4153/CJM-1974-072-0 Published online by Cambridge University Press

ascending numerical order. On each branch, the following possibilities can occur:

(C1) When the entry and all its consequences are tested in the relations, there are no necessary identifications of cosets. When the list of entries to be tested is exhausted, the Todd-Coxeter algorithm is halted without defining a new coset.

(C1.a) If the table closes with $m \leqq n$ rows, construct the coset representatives $y_1, \ldots, y_m$. For $g \in G$, let $\bar{g}$ be the representative of Ug, that is, $\bar{g} = y_i$, where $i = (1)g\varphi$. At this stage an end-point is reached corresponding to the subgroup

$$U = \langle y_i g_k \overline{y_i g_k}^{-1} | (i, k) \text{ is a branch point and there is no } j' \text{ with } j'\sigma = (i, k) \rangle.$$

U is of index m. One continues as in (C2).

(C1.b) If (i, k) is the first free place in the table, and the last free place in the m rows in which entries have been made, with $m \leqq n$, then one sets $it_k = m + 1$ and returns to the beginning. (If all subgroups of index $m \leqq n$ are desired, one need only omit this provision.)

(C1.c) If (i, k) is the first free place and (C1.b) does not occur, then we have a new branch point. If $k \leqq r$, the branches are those i' with $i \leqq i' \leqq \min(n, m + 1)$ such that the number in $(i', k + r)$ is 0. If $r < k \leqq 2r$, then the branches are those i' with $i < i' \leqq \min(n, m + 1)$ such that the number in $(i', k - r)$ is 0.

(C2) If in testing an entry in the relations a necessary identification is found, all entries made since the last branch point at which alternative branches remain are erased, and there the next highest branch is taken.

The process ends when the branches at V_0 are exhausted. The description is now complete, and as in the previous section we proceed to show that this method solves the given problem.

LEMMA 4.1. *The process is finite and single-valued.*

Proof. If the rows of the coset table are written one after another, giving a number in base $n + 1$, then these tables are generated in ascending numerical order, bounded above by the number equal to $(n + 1)^{2rm}$.

LEMMA 4.2. *If V_p is a branch point with corresponding position (i, k) and coset table T_p, and if T is a closed table agreeing with T_p in all positions up to and including (i', k'), corresponding to the previous branch point V_{p-1}, then they agree on all $(i'', k'') < (i, k)$.*

Proof. The proof is almost identical to that of Lemma 3.2, the intervening entries being determined by data on which the two tables agree.

THEOREM 4.1. *The process generates each subgroup of index n exactly once.*

Proof. The restriction on the branches in (C1.b) and (C1.c) insures that the tables are in normal order and the mappings injective. The testing procedure

insures that the relations of the group are satisfied by the completed tables. In view of Lemma 4.1, it is sufficient to construct, for each U of index n, a chain of branch points V_p, $p = 0, \ldots, k$, with the table of V_k identical to the table of U. By Lemma 4.2, we can proceed inductively, starting with V_0. At each branch point we choose the branch which fills the position (i, k), corresponding to that branch point, with the number in position (i, k) in the table of U.

Method B was given in the above form to avoid changes in notation. The original version, in Schaps [**4**], differed in the following respects: The ordering of the pairs was reversed, putting all pairs in one column before those in the next. The order in which optional entries were made was different, in that entries determining $j\sigma$ were made first to insure transitivity. Relations of order, such as $g_1^2 = e$, were tested as the entries were made. Finally, since the procedure in §5 for determining conjugates was used, only those cosets were generated in which 1 was in an orbit of maximal length in t_1.

5. Computation of conjugates. Given a coset table T_U corresponding to a subgroup U, one can generate the coset table of its conjugate $y_i^{-1}Uy_i$ by renumbering the rows, starting with $i \mapsto 1$. If $i_1, \ldots, i_j$ have been renumbered, as $1, \ldots, j$, then let i_{j+1} be the first new number in the sequence of positions

$$\begin{array}{ll} (i_1, 1), \ldots, & (i_1, 2r) \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ (i_j, 1), \ldots, & (i_j, 2r). \end{array}$$

Applying the mapping $i_j \mapsto j$ to T_U and rearranging the rows gives the desired table. The procedure can either be applied to the output of Method A or B, using standard classification procedures, or incorporated into the process of generating subgroups. This latter, a suggestion of C. Sims, can be done by checking at each branch point to insure that no conjugate of the partial coset table, insofar as it can be determined, precedes the table in lexicographical order.

6. Generalizations. Methods A and B above, apparently dissimilar, are in fact almost identical. The identification $i := :j$ at a branch point in the first method is equivalent to the entry of i in the position $j\sigma = (i', k)$ at the appropriate branch point in the second. A review of the case $j = j'$ in Lemma 3.1 may make this equivalence clearer. The element $y_i y_j^{-1}$ is the inverse of $y_{i'} g_k \overline{y_{i'} g_k}^{-1}$, since $\overline{y_{i'} g_k} = y_i$. The branch points and thus the subgroups will be generated in the same order, but there will be more branch points in the second method, since coset definitions are regarded as branches, and less information is available from the Todd-Coxeter algorithm to eliminate fruitless branches.

This underlying method can also be applied to the regular permutation representation of a group generated by a finite number of permutations or matrices. One generates the elements of the group to some sufficiently large $N > n$ by multiplying the given generators; the group elements are numbered in normal order as cosets of the identity, giving the regular representation of the group. One then uses Method A, without the Todd-Coxeter algorithm, defining new cosets by multiplying the coset representatives by the permutations or matrices given as generators.

The authors wish to thank Dr. J. Neubüser and Dr. V. Felsch for their help and encouragement in the research described above, and, for suggesting improvements in the manuscript and reading it, Drs. C. Sims, and M. Hall. The authors take full responsibility.

References

1. A. Dietze, *Drei Verfahren zur Bestimmung sämtlicher Untergruppen endlich präsentierbarer Gruppen zu vorgegebenem Index* (Diplomarbeit, Kiel, 1970).

2. H. Felsch, *Programmierung der Restklassenabzählung eine Gruppe nach Untergruppen*, Numer. Math. *3* (1961), 250–256.

3. N. S. Mendelsohn, *An algorithmic solution for a word problem in group theory*, Can. J. Math. *16* (1964), 509–516. Correction: Can. J. Math. *17* (1965), 505.

4. M. Schaps, *An algorithm to generate subgroups of finite index in a group given by defining relations* (Manuscript, Kiel, 1968).

5. C. C. Sims, *Computational methods in the study of groups*, conference on Computational Problems in Abstract Algebra, Oxford, 1967; private communications, Oxford, 1967, and Kiel, 1969.

6. J. Todd and H. S. M. Coxeter, *A practical method for enumerating cosets of a finite abstract group*, Proc. Edinburgh Math. Soc. (1936).

IBM,
Hamburg, West Germany;
Tel-Aviv University,
Tel-Aviv, Israel