The University of Warwick

THEORY OF COMPUTATION

REPORT NO.42

COSET ENUMERATION AS CLOSURE COMPUTATION

BY

W.M. BEYNON

Department of Computer Science
University of Warwick
COVENTRY CV4 7AL
ENGLAND.

SEPTEMBER 1982

Coset enumeration as closure computation

Introduction

Procedures for coset enumeration have been the subject of considerable attention since the first machine programs were developed in the early 60's. Most of these procedures, like the original Todd-Coxeter procedure from which they derived, are expressed in terms of operations on integer arrays whose entries denote coset representatives. The first formal justification of a coset enumeration procedure [T] is also expressed in terms of such a representation.

As the study of computer algorithms has often shown, there are disadvantages in formulating algorithmic processes in terms of a particular data representation. For instance, it can make a precise description unnecessarily complicated, and a formal verification more obscure. An abstract description, which makes no assumption about how data is represented, usually has conceptual disadvantages, and can be more simply prescribed and verified. The purpose of this paper is to describe and justify abstract formulations of coset enumeration procedures closely related to the original Todd-Coxeter procedure (see [TC] or [J] Ch. 6), and Trotter's modified version [T].

It may be of interest that the arguments presented here are only sufficient to guarantee termination of the original Todd-Coxeter procedure when enumerating the cosets of a normal subgroup of finite index under modest additional hypotheses on the form of relations used. (For more details, see section 4 below.)

§.1 Preliminaries

Let F be the group freely generated by $X = \{x_1,\ldots,x_n\}$, and let Q be a finite subset of F. Two related problems are considered:

(1) Determine whether the subgroup H of F generated by Q has finite index, and if so determine [F : H]

(2) Determine whether the normal closure N of H in F has finite index, and if so determine [F : N].

Standard coset enumeration techniques (see [TC] and [J] Ch.6) provide a terminating algorithm to solve problem (1), and a procedure which partially solves problem (2), insofar as it terminates and determines [F : N] if and only if N has finite index in F. The two problems are related in view of the Nielsen-Schreier Theorem [S] which shows that, if N has finite index, then it is finitely generated. Thus, a terminating algorithm to solve problem (1) can easily be adapted to give a procedure which terminates if and only if [F : N] is finite. It is only necessary to determine the index of $\langle Q_i \rangle$ in F for i = 0,1,2,..., where

$$Q_o = Q \text{ and } Q_{i+1} = Q_i \cup \bigcup_{i=1}^{n} x_i Q_i x_i^{-1},$$

until a finite index is obtained.

If F and H are as defined above, the equivalence relation defined by right cosets of H in F will be denoted by R(H). A prefix-closed class of reduced words (i.e. a Schreier system) P together with a reflexive, symmetric relation ρ on P is said to be an enumeration pair for H in F if

$$\{ab^{-1} \mid (a,b) \in \rho\}$$

is a generating set for H. The relation $\hat{\rho}$ on a Schreier system P is said to be R-closed relative to P if

$$(a,b) \in \hat{\rho} \text{ and } (af,bf) \in P \times P \text{ for some } f \text{ in } F \Rightarrow (af,bf) \in \hat{\rho}$$

The main purpose of this paper is to prove (see §.2 Thm.2.3) that if (P,ρ) is an enumeration pair for H in F, then $R(H)|P$ is the smallest relation on P which is transitively closed, R-closed relative to P, and contains ρ (hereafter denoted $R(\rho,P)$). When P is finite, this provides a finite characterisation of R(H) on P which can be used in conjunction with a criterion for H to have finite index in F (see §.3 Thm.3.2) to solve Problem (1). An abstract algorithm based on this principle, and closely related to the original Todd-Coxeter algorithm for solving Problem (1), is described in §.4 (see Algorithm A). To adapt such an algorithm to obtain a procedure for solving Problem (2), it suffices to specify how to extend an enumeration pair (P,ρ) for H in F to an enumeration pair (P',ρ') for $< H,q >$ in F, where q is an arbitrary element of F, and how to compute $R(\rho',P')$ from $R(\rho,P)$. This is illustrated in §.4, where an abstract procedure closely related to the Todd-Coxeter procedure for solving Problem (2) is described (see Procedure B).

§.2 Coset Equivalence and closure

Throughout Sections 2 and 3, H will denote a subgroup of the group F freely generated by $X = \{x_1, x_2, \ldots, x_n\}$ and (P,ρ) an enumeration pair for H in F. The relation

$$\{(a,b) \mid (af,bf) \in \rho \text{ for some } f \text{ in } F\} \subseteq F \times F$$

(i.e. the R-closure of the relation ρ viewed as a relation on F) will be denoted by $R(\rho)$.

Proposition 2.1

$R(\rho)$ is a reflexive symmetric relation whose transitive closure $R(\rho)^*$ is R(H). (In particular, $(F,R(\rho))$ is an enumeration pair for H in F)

Proof: It is easy to verify that $R(\rho)$ is a reflexive symmetric relation, and that R(H) contains $R(\rho)$ and $R(\rho)^*$.

To show that R(H) is contained in $R(\rho)^*$ it suffices to show that

$$H \times \{1\} \subseteq R(\rho)^*,$$

since both $R(\rho)$ and R(H) are R-closed (relative to F).

If $\rho = \{(a_i,b_i) \mid i \in I\}$, then any element h in H can be represented in the form $\prod_{j=1}^{k} q_j$, where each q_j has the form $a_i b_i^{-1}$ for some i.

If $k = 1$, then $h = a_i b_i^{-1}$ and $(h.b_i = a_i,\ 1.b_i = b_i) \in \rho$, so that $(h,1) \in R(\rho)$. If $k \geq 2$, then

$$h = q_i \prod_{j=2}^{k} q_j = a_i b_i^{-1} h_1$$

where $(h_1,1) \in R(\rho)$ by induction. But then

$$(h.h_1^{-1}b_i = a_i,\ h_1 h_1^{-1} b_i = b_i) \in \rho$$

whence (h,h_1) is also in $R(\rho)$. □

In reasoning about elements of F it is useful to adopt the convention of identifying an element of F with the unique reduced word which represents it. The notation

$$w = e_1 e_2 \cdots e_k$$

will be used to mean that $e_i \in X \cup X^{-1}$ for each i, and w is reduced as written. Note that if S is a Schreier system, v is in S and $w = e_1 e_2 \cdots e_k$,

then vw is in S only if $ve_1e_2 \ldots e_j$ is in S for $1 \leq j \leq k$. This simple observation is often used in subsequent arguments.

Lemma 2.2:

If (v,w) is in R(ρ), then

(i) either v and w have a (non-trivial) common suffix

or v or w is in P.

and (ii) if v is in P, and w is not, then there is a suffix s of w such that vs^{-1} and ws^{-1} are both in P.

Proof: (i) If v and w have no common suffix, neither is in P, and z is in F, then either vz or wz is outside P.

(ii) Let z be such that $(vz,wz) \in \rho$. If $z = e_1e_2 \ldots e_k$, then there is a minimal $r \geq 1$ such that $we_1e_2 \ldots e_r$ is in P. Thus $s = (e_1e_2 \ldots e_r)^{-1}$ is a suffix of w; moreover, since vz is in P, so also is vs^{-1}. □

Theorem 2.3:

The restriction of R(H) to P is the smallest relation R(ρ,P) which is transitively closed, R-closed relative to P, and contains ρ.

Proof: It is easy to verify that $R(\rho,P) \subseteq R(H)|P$.

For the converse, suppose that v and w are in P, and that $(v,w) \in R(H)$. By Proposition 2.1, there is a sequence of elements of F:

$$v = u_o, u_1, \ldots, u_k, u_{k+1} = w$$

such that $(u_i, u_{i+1}) \in R(\rho)$ for $0 \leq i \leq k$. Since R(ρ,P) is transitive and contains the restriction of R(ρ) to P, it will suffice to consider the case when $u_1, \ldots, u_k$ lie outside P.

The proof is by induction on k; the case k = 0 is trivial. If $k \geq 1$, then by Lemma 2.2(i) the words $u_1, \ldots, u_k$ have a common suffix s which is minimal with respect to the condition that for some j with $1 \leq j \leq k$ the word $u_j s^{-1}$ is in P. Note that s is also the minimal suffix of u_j such that $u_j s^{-1}$ is in P.

By Lemma 2.2(ii), there is a suffix t of u, such that both $u_o t^{-1}$ and $u_1 t^{-1}$ are in P. Whether or not $u_1 s^{-1}$ is in P, it must be that s is a prefix of t. Since u_o and $u_o t^{-1}$ are both in P, so also is $u_o s^{-1}$. Similarly, $u_{k+1} s^{-1}$ is in P. Thus $(u_i s^{-1}, u_{i+1} s^{-1}) \in R(\rho)$ for $0 \leq i \leq k$, where $u_o s^{-1}$, $u_j s^{-1}$ and $u_{k+1} s^{-1}$ are in P. By the inductive hypothesis, $(u_o s^{-1}, u_j s^{-1})$ and $(u_j s^{-1}, u_{k+1} s^{-1})$ are in $R(\rho,P)$, whence $(u_o, u_{k+1}) \in R(\rho,P)$ using transitivity and R-closure relative to P. □

§.3 Testing for finite index

If P is finite, Theorem 2.3 provides a finite characterisation of coset equivalence restricted to P. Thus $R(H)|P$ can be computed as the appropriate closure of a finite relation on a finite set. To solve Problem (1), a criterion for $[F : H]$ to be finite (and a method of evaluating it if it is) must also be described. Such a criterion is given by Lemma 3.1 and Theorem 3.2 below.

A basic theorem of combinatorial group theory asserts that a subgroup of a finitely generated group which has finite index is itself finitely generated (see e.g. [H] Cor.7.2.1). The result connects Problem (1) and the more general problem of determining the index of the subgroup generated by a countable set, of which Problem (2) is a special case. It is convenient in this context to derive the relevant propositions from combinatorial group theory directly, since the arguments required are simple and pertinent.

Lemma 3.1: If H has finite index in F, then every right coset of H in F intersects P.

Proof: Let w be an arbitrary element of F. There is an element e in $X \cup X^{-1}$ such that wew is also reduced. Since $[F : H]$ is finite, there is an integer $k \geqslant 1$ such that $(we)^k$ is in H, and

$$(w, (we)^{1-k} e^{-1}) \in R(H).$$

In view of the choice of e, the word

$$v = (we)^{1-k} e^{-1}$$

is then reduced, and v and w have no non-trivial common suffix.

By Proposition 2.1, there is a sequence

$$w = u_0, u_1, \ldots, u_k = v$$

of elements of F, such that $(u_i, u_{i+1}) \in R(\rho)$ for $i=0, 1, \ldots, k-1$. There must be an index j for which u_j and u_{j+1} have no non-trivial suffix, and one or other is in P by Lemma 2.2(i). Thus, by Proposition 2.1, the right coset of w in F intersects P. □

Theorem 3.2:

Let E denote X or X^{-1}.

(i) If $[F : H]$ is finite, then (*) : given e in E and p in P, there is an r in P such that $(p,r) \in R(H)$ and $re \in P$.

(ii) When P is finite, the converse implication also holds.

Proof: (i) Suppose that $[F : H]$ is finite, that $p \in P$ and $e \in E$. If pe is not in P, then by Prop.2.1 and Lemma 3.1, there is a sequence

$$pe = u_0, u_1, \ldots, u_k = p'$$

of elements of F such that $(u_i, u_{i+1}) \in R(\rho)$ for $0 < i < k-1$, and $\rho' \in P$. Let u_{j+1} be the first element of P in this sequence. By Lemma 2.2(ii), the element u_j has suffix e, and $u_{j+1}\, e^{-1}$ is in P. Thus $r = u_{j+1}\, e^{-1}$ satisfies the required conditions.

(ii) Suppose that P is finite, and that (*) holds. It suffices to show that if (*) holds when E is either X or X^{-1}, it also holds with $E = X \cup X^{-1}$, since an easy induction on length then proves that each element f in F is in the same right coset as an element of P.

Let $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ be the set of equivalence classes of P under R(H). If $e \in E$ (= X or X^{-1}), then by (*) each C_j contains an element r such that re is also in P. It is easy to verify that mapping C_j to the class C_k which contains re is a well-defined 1-1 map $\mathcal{C} \to \mathcal{C}$. It is thus bijective, whence every class C_k contains an element s such that se^{-1} is in P. □

It remains to indicate how Theorem 2.3, Lemma 3.1 and Theorem 3.2 can be used in enumeration algorithms.

Suppose that G is a subgroup of F generated by the set Z. The set of all prefixes of Z will be denoted by pre(Z). The pair (pre(Z), ρ(Z)), where $\rho(Z) = (Z \cup \{1\}) \times (Z \cup \{1\})$, is then an enumeration pair for G in F.

If Z is finite, then the restriction of R(G) to pre(Z) can be obtained by evaluating the finite relation R(P(Z), pre(Z)). Condition (*) of Theorem 3.2 is then satisfied if and only if $[F : G]$ is finite. Moreover,

if (*) holds, then $[F : G]$ can be determined by Lemma 3.1. Thus there is a terminating algorithm to compute $[F : G]$ when G is finitely generated.

If $Z = \{z_i \mid i \geq 1\}$ and $Z_k = \{z_i \mid 1 \leq i \leq k\}$, then $(P_i, \rho_i) = (\mathrm{pre}(Z_i), \rho(Z_i))$ is an enumeration pair for $\langle Z_i \rangle$, and $\bigcup_{i \geq 1} (P_i, \rho_i)$ is the enumeration pair $(\mathrm{pre}(Z), \rho(Z))$ for $G = \langle Z \rangle$. Should $[F : G]$ be finite, then condition (*) will be satisfied for $P = \mathrm{pre}(Z)$, and hence for P_m for some sufficiently large m. For this m, the index of $\langle Z_m \rangle$ in F is finite, and may be computed as above. (Clearly there is also an index M for which $[F : \langle Z_M \rangle] = [F : H]$ (i.e. such that H is finitely generated by Z_M), but M is not in general computable). Thus, given the countable set Z, there is a procedure (viz. computing $[F : \langle Z_i \rangle]$ for $i = 0,1,2,\ldots$ until a finite index is encountered) which terminates if and only if $[F : G = \langle Z \rangle]$ is finite (but cannot in general evaluate a finite index).

If $G = N$ is the normal closure of H as specified in Problem (2), then N is generated by the infinite set $Z = \bigcup_{f \in F} fQf^{-1}$. If $[F:N]$ is finite, then the procedure described above will construct a finitely generated subgroup $R = \langle Z_m \rangle$ of N such that $[F : R]$ is finite. In this special case, it is then possible to determine $[F : N]$ precisely. To see this, suppose that (P, ρ) is a finite enumeration pair for R. By Lemma 3.1, if f is an element of F, then f is in the same right coset of R as p in P, and $\langle R,f \rangle = \langle R,p \rangle$. Thus (P, ρ'), where $\rho' = \rho \cup (p,1) \cup (1,p) \cup (p,p)$, is a finite enumeration pair for $\langle R,f \rangle$. Now consider the sequence of subsets

$$Y_1 = Z_m, Y_2, \ldots, Y_i, \ldots$$

where $Y_{i+1} = Y_i \cup \bigcup_{p \in P} pY_ip^{-1}$. By an extension of the above principle, $T_i = \langle Y_i \rangle$ has an enumeration pair of the form (P, ρ_i) for each i. Moreover, since $[F : R]$ is finite, there is an index j for which $T_{j+1} = T_j \supseteq \bigcup_{p \in P} pT_jp^{-1}$. If now $f \in F$, then $f \in T_j p$ for some p in P by Lemma 3.1, whence $fT_jf^{-1} \in T_j p T_j p^{-1} T_j^{-1} \subseteq T_j$. This proves that T_j is normal, so that $T_j = N$.

§.4 Examples of abstract coset enumeration

Suppose that F, Q and H are as in Problem (1). The following simple algorithm accepts Q as input, determines whether $|F : H|$ is finite, and evaluates the index when finite.

Algorithm A:

1. $(P, \rho) \leftarrow (pre(Q), \rho(Q))$

   repeat

2. $\rho' \leftarrow \rho$

3. $\rho \leftarrow <$ transitive closure of $\rho >$

4. $\rho \leftarrow <$ R-closure of ρ relative to $P >$

   until $\rho = \rho'$

5. $R(\rho, P) \leftarrow \rho$

6. if $<$ for all pairs (p,X) in $P \times X$, there is a pair (r, rx) in $P \times P$ with (p,r) in $R(\rho, P) >$

   then $[F : H] \leftarrow |P/R(\rho, P)|$

   else $<[F : H]$ is infinite $>$

From a standard implementation of a coset enumeration algorithm, in which integers are used to represent equivalence classes of elements of F, the abstract algorithm used to evaluate $R(\rho,P)$ (lines 2-4 in Algorithm A) may be difficult to discern. Indeed, in hand computations with the classical Todd-Coxeter algorithm for Problem (1), the evaluation of equivalences is often carried out in an ad hoc manner. Even so, it is not difficult to recognise the classical algorithm as essentially an implementation of Algorithm A. The data structures used consist of a $1 \times (|q| \times 1)$ array A_q for each generator q in Q, and a table T whose rows are indexed by integers representing equivalence classes of the (dynamically changing) relation ρ on pre(Q), and whose columns are indexed by X. The principle of the enumeration is to assign integers to the elements of pre(Q) (which correspond to the entries in the tables A_q) in such a way that elements found to be

equivalent are assigned the same integer. This method of representation automatically guarantees that ρ is transitive (indeed an equivalence relation) at all times (c.f. line 2 of Algorithm A). The initialisaton of each table A_q so that '1' appears as first and last entry corresponds to assigning ρ to $(Q \times \{1\}) \cup (\{1\} \times Q)$ initially, and $\rho(Q)$ is the transitive closure of this relation (c.f. lines 1-3 of Algorithm A). Table T is used to monitor instances of pairs (vx_i, w) in as they are identified, so as to ensure that w_1 and w_2 are equivalenced if either (vx_i, w_1) and (vx_i, w_2) or (w_1x_i, v) and (w_2x_i, v) are in ρ. This corresponds to computing the R-closure of ρ (c.f. line 3 of Algorithm A). When all the tables A_q are completed, the relation $R(\rho, P)$ has been evaluated, and the condition for $[F : H]$ to have finite index (c.f. line 6 of Algorithm A) is tested by determining whether T is complete.

Problem (2) differs significantly from Problem (1) in that even if N is finitely generated, a set of generators must be dynamically constructed. From the abstract perspective adopted in this paper, it may be seen that different techniques for solving Problem (2) are obtained, depending on

(a) the choice of algorithm used to evaluate the equivalence relation $R(P, \rho)$

and (b) the strategy used to construct an enumeration pair (P', ρ') for $< H, q >$ in F from the enumeration pair (P, ρ) for H in F, when a new generator q is dynamically introduced.

Procedure B below is essentially an abstract version of the coset enumeration technique described in [J]. Without additional assumptions about Q, there appears to be no guarantee that the procedure terminates even when $[F : N]$ is finite. (This point will be discussed more fully later).

Procedure B

Input: A finite subset Q of the group F freely generated by $X = \{x_1, x_2, \ldots, x_n\}$, such that the elements of Q collectively involve all elements of X.

Output if termination: The index [F : N], where N is the normal closure of Q in F. Finiteness of [F : N] is a necessary, but possibly not sufficient, condition for termination.

The procedure:

1. $(P, \rho) \leftarrow (\{1\}, \{(1, 1)\})$
2. $G \leftarrow \phi$
3. while <there is an unmarked class in P/ρ do
   (i.e. a class containing no marked element)>
   begin
4. assert "$(F, \rho = R(\rho, P))$ is an enumeration pair for $< G >$"
5. $p \leftarrow <$ the shortest element in the union of all unmarked classes $>$
6. for < each element q in Q > do
   begin
7. $(P, \rho) \leftarrow (P \cup pre(\{pq\}), \rho \cup \{(pq, p), (p, pq)\})$
8. $G \leftarrow G \cup pqp^{-1}$
9. < compute $R(\rho, P)$ (e.g. as in Algorithm A lines 2-4)>
10. $(P, \rho) \leftarrow (P, R(\rho, P))$
   end
11. mark(p)
   end
12. assert " for all pairs (p,x) in $P \times X$ there is a pair
   (r, rx) in $P \times P$ with (p,r) in $R(\rho, P)$ "
13. $[F : N] \leftarrow | P/(\rho = R(\rho, P)) |$

(The assignments to G (lines 2 and 8) are included for conceptual reasons, and permit the formulation of the invariant of the while-loop (line 4).)

Procedure B is justified by the following theorem.

Theorem 4.1:

If Procedure B terminates, the index $\lceil F : N \rceil$ is finite, and is correctly determined.

Proof: If (P, ρ) is an enumeration pair for $< G >$, then

$(P \cup pre(\{pq\}), \rho \cup \{(pq, p), (p, pq), (pq, pq), (p, p)\})$

is an enumeration pair for $< G, pqp^{-1} >$. From this, it follows that the assertion at line 4 is an invariant of the while-loop.

Suppose that the while-loop terminates with $\langle G \rangle = K$. It is obvious that $K \subseteq N$; moreover, by Theorem 3.2(i), the truth of the assertion at line 12 will suffice to prove that $\lceil F : K \rceil$ is finite. Accordingly, let $(p, x) \in P \times X$. By hypothesis, there is a prefix s in pre(Q) such that both s and sx are in pre(Q). Since all classes in P/ρ are marked on termination of the while-loop, for each p_1 in P there is a marked element p_1' with (p_1, p_1') in ρ. There is then a well-defined map $P/\rho \to P/\rho$ taking the class of p_1 to the class of p_1's which is 1-1, whence bijective. Thus p_1 can be chosen such that $(p, p_1 s) \in \rho$, and $r = p_1 s$ satisfies the finiteness condition at line 12.

Finally, it suffices to show that K = N; that is, that K contains all conjugates of elements of Q. Suppose then that f is any element of F. By Lemma 3.1, there is an element k in K, and a (marked) element p in P such that f = kp, and

$$fqf^{-1} = kpqp^{-1}k^{-1} \in K. \qquad \square$$

It remains to consider sufficient conditions for termination of Procedure B. Let G_i, ρ_i and p_i be the values of the variables G, ρ and p at the end of the ith iteration of the while-loop, so that $G_o = \phi$ and

$G_{i+1} = G_i \cup p_i Q p_i^{-1}$. It may be seen that G_{i+1} depends upon G_i and the class of p_i under p_i, not upon the element p_i.

If [F:N] is finite, then arguments similar to those at the end of §.3 will establish that non-termination of the while-loop is only consistent with $\bigcup_{i=1}^{\infty} \langle G_i \rangle \subsetneq N$. Accordingly, to prove termination it suffices (for instance) to establish that given any integer k, all conjugates of elements of Q by elements of F of length at most k are guaranteed to be in $\langle G \rangle$ if the while-loop is repeated sufficiently often.

Examination of Procedure B shows that the method by which p is chosen at line 5 and G is augmented at line 7 will ensure that $\bigcup_{i=1}^{\infty} \langle G_i \rangle$ contains all conjugates of elements of Q by elements of $\text{pre}(Q)^*$. The following lemma will be used to guarantee termination of Procedure B when [F:N] is finite under reasonably general assumptions about the form of Q.

Lemma 4.2: Let Q and F be as above. Suppose that $a_1,\ldots,a_k$ in F are in the same right coset of $\langle Q \rangle$ in F as $b_1,b_2,\ldots,b_k$ respectively. Then

$$A \equiv \langle Q,\ a_1 Q a_1^{-1},\ a_1 a_2\, Q (a_1 a_2)^{-1},\ldots,a_1\ldots a_k Q (a_1\ldots a_k)^{-1} \rangle$$

$$= B \equiv \langle Q,\ b_1\, Q b_1^{-1},\ b_1 b_2\, Q (b_1 b_2)^{-1},\ldots,b_1\ldots b_k Q (b_1\ldots b_k)^{-1} \rangle$$

Proof: (Induction on k). If k = 0, there is nothing to prove. Suppose then that $k = r > 0$. By the inductive hypothes

$$\langle Q_1\ a_2 Q a_2^{-1},\ldots,a_2\ldots a_r\, Q (a_2\ldots a_r)^{-1} \rangle$$

$$= \langle Q_1\ b_2 Q b_2^{-1},\ldots,b_2\ldots b_r\, Q (b_2\ldots b_r)^{-1} \rangle$$

Let $S = Q \cup a_2 Q a_2^{-1} \cup \ldots \cup a_2\ldots a_r\, Q (a_2\ldots a_r)^{-1}$.

Then

$$A = \langle Q,\ a_1 S a_1^{-1} \rangle \text{ and } B = \langle Q,\ b_1 S b_1^{-1} \rangle .$$

But if $s \in S$, then

$$a_1 s a_1^{-1} = a_1 b_1^{-1}\ b_1 s b_1^{-1}\ b_1 a_1^{-1} \in \langle Q, b_1 S b_1^{-1} \rangle$$

whence $A \subseteq B$. Similarly $B \subseteq A$. □

In view of Lemma 4.2, termination of Procedure B is guaranteed if $[F:N]$ is finite and $pre(Q)^*$ contains representatives for the right cosets of Q in F which contain the generators $x_1,\ldots,x_n$ and their inverses. This is necessarily the case (for example) if Q contains an element with prefix x_i or suffix x_i^{-1} and an element with suffix x_i or prefix x_i^{-1} for $1 \leq i \leq n$. In particular, if Q contains an element of x_i^* for $1 \leq i \leq n$, then termination is guaranteed. For an arbitrary Q, it is possible to ensure that the above condition for termination is met without affecting the normal closure of Q by introducing appropriate cyclic permutations of the relations in Q.

The above arguments are not sufficient to guarantee termination of Procedure B under the sole assumption that $[F:N]$ is finite. A satisfactory proof that finiteness of $[F:N]$ is sufficient for termination (or a counterexample!) would be of interest. There are many facile and erroneous arguments which depend upon making the implicit assumption that $\cup \langle C_i \rangle = N$.

(Note particularly that Lemma 4.2 deals with cosets of $\langle Q \rangle$ and not N, so that, for instance, $a_1 \ldots a_k$ and $b_1 \ldots b_k$ may lie in distinct right cosets.) It may be significant that the justification of the Todd-Coxeter procedure presented in [LS] p.164-166 omits details over this point.

The identification of the enumeration procedure in [J] as an implementation of Procedure B is very similar to that of Algorithm A, and only the principal details are described here. Tables A_q for q in Q, and a monitoring table T are used as before, but here each A_q (like T) has rows indexed by integers representing equivalence classes. If n is the encoding of the class of p as in line 4 of Procedure B, the nth row of A_q (which has first and last entry n) has entries corresponding to elements of the form pr where r is a prefix of q. Since the first row of A_q faithfully reflects the cyclic nature of the relation $q = 1$, it is not necessary to re-write the relations Q to ensure termination, provided that when a new equivalence class is introduced its encoding is guaranteed

to appear in the left context of an x_i or the right context of an x_i^{-1} in some table A_q. (This is achieved in some enumeration algorithms by introducing new rows in which the newly defined integer appears in each possible column).

Procedure B has a particularly simple control structure, and the subtleties of its justification reflect this. Procedure C below affords an alternative solution to Problem (2), closer in spirit to Trotter's procedure [T]. Its justification is more straightforward, and only the principal details are given here.

The following subroutine is used at line 6:

```
subroutine reduce (g = e_1 e_2...e_k, P, ρ)

    begin
1.      s ← 1
2.      for i = 1 to k do
3.        if < s ∈ P and ∃ r ∈ P such that (r,s) ∈ ρ and re_i ∈ P >
          then
              begin
4.                < let r_o be the lexicographically first r
                    satisfying the above condition >
5.                  s ← r_o e_i
              end
6.        else  s ← se_i
7.      return(s)
    end
```

Procedure C

Input: A finite subset Q of the group F freely generated by $X = \{x_1, x_2, \ldots, x_n\}$.

Output if termination: The index [F:N], where N is the normal closure of Q in F. Finiteness of [F:N] is a necessary and sufficient condition for termination.

The procedure

1. $(P, \rho) \leftarrow (\{1\}, \{(1, 1)\})$
2. $G \leftarrow \phi$
3. < let $f_o = 1, f_1, f_2, \ldots$ be an enumeration of F >
4. $i \leftarrow 0$
5. while < there is an unmarked class in P/ρ >
   or < (*) (see below) does not hold for (P, ρ) > do
   begin
6. $f \leftarrow$ reduce (f_i, P, ρ)
7. if < f is not marked > then
8. for < each element q in Q > do
   begin
9. $(P,\rho) \leftarrow (P \cup \text{pre}(\{fq\}), \rho \cup \{(fq,f),(f,fq)\})$
10. $G \leftarrow G \cup fqf^{-1}$
11. < compute $R(\rho, P)$ >
12. $(P, \rho) \leftarrow (P, R(\rho, P))$
   end
13. mark(f)
14. $i \leftarrow i+1$
   end
15. $[F:N] \leftarrow |P/(\rho = R(\rho, P))|$

(*) is here the finiteness condition used in Procedure B line 12, viz: "for all pairs (p,x) in P×X there is a pair (r, rx) in P×P with (p,r) in $R(\rho, P)$"

The principle of the algorithm is to augment the generating set Q by adjoining conjugates of elements of Q by the elements of F in a systematic fashion until (if [F:N] is finite) a set of generators for N has been found. If (as above) G_i and P_i are used to denote the values of the variables G and ρ at the end of the ith iteration of the while-loop, the pair of boolean conditions at line 5 together ensure that on termination with i = k, the subgroup $< G_k >$ is normal in F, and has finite index. The subroutine "reduce" is used to determine the lexicographically first element f of F which is the same right coset of G_i in F as f_i. Since $< G_i, f_i\, q f_i^{-1} > = < G_i, fqf^{-1} >$, it is easy to show in this case that termination occurs when N has finite index.

For conceptual reasons, Procedure C has been described in terms of an explicit enumeration of F. In practice, the element f_i introduced at line 6 would be produced by a procedure call, and the enumeration of F generated might depend on the form of Q. Reference to Lemma 4.2 shows that for certain sets of relations it is enough to enumerate a subset of F, viz. a set of representatives for the right cosets of $< Q >$ in F. Thus an enumeration of $F \cap X^*$ is sufficient if (for example) Q contains an element of x_i^+ for $1 \leqslant i \leqslant n$.

REFERENCES

[H] HALL, M., Jr. The Theory of Groups. NY Macmillan 1959.

[J] JOHNSON, D.L. Presentation of Groups.
LMS Lecture Note Series 22, Cambridge University Press, 1976.

[LS] LYNDON, R.C. & SCHUPP, P.E. Combinatorial Group Theory.
Ergebnisse der Mathematik und ihrer Grenzgebiete 89,
Springer Verlag, 1977.

[S] SCHREIER, O. Die Untergruppen der freien Gruppen.
Abh. Math. Sem. Univ. Hamburg 5, 161-183 (1927).

[TC] TODD, J.A. & COXETER, H.S.M. A practical method for enumerating
cosets of a finite abstract group.
Proc. Edinburgh Math. Soc. 5, 25-34 (1936).

[T] TROTTER, H. An algorithm for the Todd-Coxeter method of
coset enumeration. Canad. Math. Bull. 7, 357-368 (1964).