The \( p \)-Group Generation Algorithm

E. A. O'BRIEN

Department of Mathematics, Statistics and Computer Science,
Marquette University, Milwaukee, WI 53233, USA

Dedicated to Professor G. E. Wall on the occasion of his 65th birthday.

(Received 18 December 1988)

The theory and implementation of an algorithm used in generating descriptions of \( p \)-groups are described. Some applications and details of the performance of the algorithm are provided.

1. Introduction

In a 1977 paper, Newman gave a theoretical description of an algorithm that can be used to generate descriptions of finite \( p \)-groups. The theory and implementation of this algorithm, now known as the \( p \)-group generation algorithm, are described in detail in this paper. The description of the implementation is intended to be sufficient to allow a reader to write a similar implementation if it is required. In addition, space and time limitations on the performance of the algorithm implementation are discussed. An extension of the algorithm is developed which partially solves these problems.

A partial implementation of the algorithm was carried out by Alford, Ascione, Havas, Leedham-Green & Newman in 1976. For an early application, see Ascione, Havas & Leedham-Green (1977). A complete implementation of the extended algorithm was carried out by Newman & O'Brien in 1986. Some of the material presented in Section 2 of the paper has appeared in a condensed form in Newman (1977); such material is included here for completeness.

The implementation of an extended \( p \)-group generation algorithm has permitted a computer-based determination of the groups of order 128 and 256 to be carried out for the first time. A detailed description of this work is provided in James, Newman & O'Brien (1990) and in O'Brien (to appear). The problem of describing all groups of a fixed order by giving one presentation for each isomorphism type of group of that order was initiated by Cayley (1878). For a detailed bibliography on the history of group determinations, see O'Brien & Short (1988).

In a 1940 paper, P. Hall described a classification theory for groups of prime-power order. He defined an equivalence relation, isoclinism, on groups which splits them into a number of mutually exclusive families. Hall showed that there are 10 isoclinism families of the groups of order \( p^5 \), where \( p \) is an odd prime. In the 1930s, Hall & Senior used the theory of isoclinism to determine the 340 groups of order dividing 64 which are classified into 27 families. This work formed the basis of the tables for the groups of order dividing 64 later published by Hall & Senior (1964). Easterfield (1940) used isoclinism as the basis of his classification of the groups of order \( p^5 \), where \( p \) is an odd prime, as did James (1980) in his work on this classification. James et al. (1990) used the method described by Hall to calculate the isoclinism families of the groups of order 128; the reader is referred to that
work for a further discussion of the subject. No detailed comparison of the algorithm described here with that given by Hall has been carried out.

2. The Theory of the $p$-Group Generation Algorithm

Finite groups of prime-power order may be described uniformly using power-commutator presentations. The generating set is a finite set $\{a_1, \ldots, a_n\}$. The defining relations are

$$a_i^p = \prod_{k=1}^{n} a_k^{\beta(i,k)}, \quad 0 \leq \beta(i,k) < p, \quad 1 \leq i \leq n,$$

$$[a_i, a_j] = \prod_{k=1}^{n} a_k^{\delta(i,j,k)}, \quad 0 \leq \delta(i,j,k) < p, \quad 1 \leq i < j \leq n.$$

Presentations of this kind were first defined by Sylow (1872) who proved that every group of order $p^n$ has a power-commutator presentation on $n$ generators. If such a presentation on $n$ generators defines a group of order $p^n$, then the presentation is consistent.

A number of algorithms for the computation of power-commutator presentations of finite $p$-groups have been developed. These are known as nilpotent quotient algorithms. An implementation of a nilpotent quotient algorithm is described in Havas & Newman (1980). Some important aspects of the implementation of this algorithm are recalled below.

Let $G$ be a $d$-generator $p$-group of order $p^n$. Additional structure is imposed on a consistent power-commutator presentation of $G$ so that, for each $a_k$ in $\{a_{d+1}, \ldots, a_n\}$, there is at least one relation whose right-hand side is $a_k$. Exactly one of these relations is taken as the definition of $a_k$. The definitions are of two types: either $a_i^p = a_k$ where $i < k$ and $a_i$ is a $p$th power of some element or $i \leq d$; or $[a_j, a_l] = a_k$ where $i < j < k$ and $i \leq d$.

A weight function, $\omega$, is also defined on the $n$ generators according to the following rules:

(i) $\omega(a_i) = 1$ for $i = 1, \ldots, d$;
(ii) if the definition of $a_k$ is $a_i^p$, then $\omega(a_k) = \omega(a_i) + 1$;
(iii) if the definition of $a_k$ is $[a_j, a_l]$, then $\omega(a_k) = \omega(a_j) + \omega(a_l)$.

Wamsley (1974), Havas & Newman (1980), and Vaughan-Lee (1984) show that weighted power-commutator presentations can be used to reduce the amount of consistency checking required on the presentation.

The Havas and Newman implementation uses a variation of the lower central series known as the lower exponent-$p$ central series. This is the descending sequence of subgroups

$$G = P_1(G) \supseteq \ldots \supseteq P_{c-1}(G) \supseteq P_c(G) \supseteq \ldots$$

where $P_i(G) = [P_{i-1}(G), G]P_{i-1}(G)^p$ for $i \geq 1$.

If $P_c(G) = 1$ and $c$ is the smallest such integer then $G$ has exponent-$p$ class $c$. A group with exponent-$p$ class $c$ is nilpotent and has nilpotency class at most $c$. In this paper, the class of $G$ refers to the exponent-$p$ class of $G$. Nilpotency class is explicitly indicated as such.

Three important properties of the lower exponent-$p$ central series are collected together for later reference.

1. If $\theta$ is a homomorphism of $G$ then $P_c(G)\theta = P_c(G\theta)$.
2. If $N \triangleleft G$ and the quotient $G/N$ has class $c$ then $P_c(G) \leq N$.
3. If $G$ is a finite $p$-group then $P_1(G)$ is the Frattini subgroup of $G$. 
Given a description of a group $G$, a prime $p$, and a positive integer $c$, the nilpotent quotient algorithm constructs a consistent power–commutator presentation for the largest $p$-quotient of $G$ having class $c$. In theory, one can deal with group descriptions given in various ways—in practice, the Havas & Newman implementation accepts only finite presentations, possibly combined with exponent laws.

The ability to construct such presentations for finite $p$-groups provided the framework for the development of the $p$-group generation algorithm. This algorithm calculates (presentations for) particular extensions, known as immediate descendants, of a finite $p$-group. Let $G$ be a finite $p$-group with (minimal) generator number $d$ and class $c$.

**Definition 2.1.** A group $H$ is a descendant of $G$ if $H$ has generator number $d$ and the quotient $H/P_c(H)$ is isomorphic to $G$. A group is an immediate descendant of $G$ if it is a descendant of $G$ and has class $c + 1$.

Clearly, $G$ is a descendant of the elementary abelian group, $G/P_c(G)$, of order $p^e$; also, $G/P_{c+1}(G)$ is an immediate descendant of $G/P_c(G)$ for $i < c$. Thus, it is possible to calculate $G$ using an iterative method of calculating immediate descendants, starting with the elementary abelian group of rank $d$. Since every group can be calculated in this way, it is theoretically possible to obtain a complete list of all $d$-generator $p$-groups. In practice, it is desirable that the list is both complete and irredundant—that is, a representative of each isomorphism type is present and no two elements in the list have the same isomorphism type. The following theorem is fundamental in an attempt to construct such a list.

**Theorem 2.2.** Let $G$ be a $d$-generator $p$-group. Then there exists a group, $G^*$, where every $d$-generator group $H$ having a central elementary abelian $p$-subgroup, $Z$, such that $H/Z$ is isomorphic to $G$, is a homomorphic image of $G^*$.

**Proof.** Let $F$ be the free group of rank $d$ freely generated by $a_1, \ldots, a_d$, and let $R$ be the kernel of a homomorphism $\theta$ from $F$ onto $G$. Define $R^*$ to be $[R, F]R^p$ and $G^*$ to be $F/R^*$. Then $G^*$ has $d$ generators and, since $R < F$, $R^* \leq R$.

Since $H$ is a $d$-generator group and has a quotient which is isomorphic to $G$, the homomorphism, $\theta$, may be factored through $H$ and the resulting homomorphism, $\psi$, of $F$ onto $H$ maps $R$ into $Z$. Since $Z$ is elementary abelian and central, $\psi$ maps both $R^p$ and $[R, F]$ to the identity in $H$. Thus, the image of $R^*$ in $H$ is the identity and $H$ is a homomorphic image of $F/R^*$.

**Lemma 2.3.** The isomorphism type of $G^*$ depends only on $G$ and not on $R$.

**Proof.** Let $R_1$ and $R_2$ be normal subgroups of $F$; let $F/R_1 = G_1$ and $F/R_2 = G_2$ where $G_1 \cong G_2$. Following the notation of Theorem 2.2, $R^*_1, G^*_1, R^*_2, G^*_2$ are defined. This theorem shows that $G^*_1$ is isomorphic to $G^*_2$, since each is a homomorphic image of the other.

It follows from Theorem 2.2 that every immediate descendant of $G$ is isomorphic to a quotient of $G^*$ and, since $F/R$ has class $c$, $G^*$ has class at most $c + 1$.

Some of the notation established in Theorem 2.2 is used, without reference, in the remainder of the paper. The group $G^*_p = F/R^*$ is the $p$-covering group of $G$. The factor
group $R/R^*$ is the $p$-multiplicator of $G$ and the nucleus of $G$ is $P_c(G^*)$. An allowable subgroup is a subgroup of the $p$-multiplicator which is the kernel of a homomorphism from $G^*$ onto an immediate descendant of $G$.

Given $G$, the first step in the generation algorithm is to calculate $G^*$. An important requirement of an efficient algorithm for the construction of immediate descendants of $G$ is to characterize the required quotients of $G^*$ easily.

**Theorem 2.4.** A subgroup is allowable if and only if it is a proper subgroup of the $p$-multiplicator of $G$ which supplements the nucleus.

**Proof.** Let $M/R^*$ be an allowable subgroup—that is, the kernel of a map from $F/R^*$ onto an immediate descendant $H$ of $G$. Since $G$ has class $c$ and $H$ has class $c+1$, it is clear that $M$ is a proper subgroup of $R$. Property 2 shows that $P_c(F)$ is a subgroup of $R$ and $M$ is also a subgroup of $R$ and, hence, $MP_c(F)$ is a subgroup of $R$. Following Theorem 2.2, $R\psi$ is a subgroup of $P_c(H)$. Since $F/R$ has class $c$, $P_c(F\psi)$ is also a subgroup of $R\psi$ showing that $R\psi$ equals $P_c(H)$. But $R\psi$ also equals $R/M$ and $P_c(H) = (P_c(F)M)/M$. Therefore, $R/M = (P_c(F)M)/M$ giving $R = P_c(F)M$. Hence, $R^*$ can be factored out showing that $(M/R^*)P_c(F)R^*/R^* = R/R^*$. Property 1 gives the required statement.

Conversely, let $M/R^*$ be a proper subgroup of the $p$-multiplicator that supplements the nucleus. Then $(P_c(F)M)/R^* = R/R^*$ and, hence, $(P_c(F)M)/M = R/M$. Property 1 gives $P_c(F/M) = R/M$. Since $F/M$ is a quotient of $F/R^*$, it has generator number $d$ and the quotient $(F/M)/P_c(F/M)$ is isomorphic to $G$ showing that $F/M$ is a descendant of $G$. But $P_c(F/M) = R/M$ which is non-trivial and, therefore, $F/M$ has class $c+1$ and is an immediate descendant of $G$.

If $G$ has immediate descendants, it is capable; otherwise, it is terminal. Clearly, $G$ is capable if and only if $G^*$ has class exactly $c+1$. Hall & Senior (1964) described a $p$-group, $G$, as capable if there exists a group whose central quotient is isomorphic to $G$.

If $G$ is capable, on taking factor groups of $G^*$ by allowable subgroups a complete list of immediate descendants is obtained; this list usually contains redundancies. To eliminate these redundancies, an obvious equivalence relation is defined on the allowable subgroups: two allowable subgroups $M_1/R^*$ and $M_2/R^*$ are equivalent if and only if their quotients $F/M_1$ and $F/M_2$ are isomorphic.

A complete and irredundant set of immediate descendants of $G$ can be obtained by factoring $G^*$ by one representative of each equivalence class. In practice, this definition is useful only because the equivalence relation can be given a different characterization by using the automorphism group, Aut $G$, of $G$. An extension of each automorphism, $\alpha$, of $G$ to an automorphism, $\alpha^*$, of $G^*$ is described below. The action of $\alpha^*$ when restricted to the $p$-multiplicator of $G$ is uniquely determined by $\alpha$, and $\alpha^*$ induces a permutation of the allowable subgroups. It is shown that the equivalence classes of allowable subgroups are exactly the orbits of the allowable subgroups under the action of these permutations.

**Theorem 2.5.** Let $M_1/R^*$ and $M_2/R^*$ be subgroups of $F/R^*$ which are contained in $R/R^*$ and let $\phi$ be an isomorphism from $F/M_1$ to $F/M_2$. Then there exists an automorphism, $\alpha^*$, of $G^*$ which maps $M_1/R^*$ to $M_2/R^*$ and the map from $F/M_1$ to $F/M_2$ induced by $\alpha^*$ agrees with $\phi$.

**Proof.** For each $i \in \{1, \ldots, d\}$, let $b_i$ be a word in $F$ such that $a_i M_1 \psi = b_i M_2$. Using
Property 1, it follows that
\[(R/M_i)\phi = P_c(F/M_i)\phi = P_c((F/M_i)\phi) = P_c(F/M_2) = R/M_2.\]
Therefore, \(\phi\) induces an automorphism, \(a\), on \(F/R\).

A mapping, \(a^*\), is now defined for the automorphism, \(a\), of \(F/R\). For each \(i \in \{1, \ldots, d\}\), choose a representative \(u_i\) in \(F\) of the coset \(a_iR\); then \(a_iR\alpha = u_iR\). Let \(v(a_1, \ldots, a_d)\) be a word in \(F\); then
\[v(a_1, \ldots, a_d)R\alpha = v(u_1, \ldots, u_d)R.\]
If \(v(a_1, \ldots, a_d)\) is an element of \(R\), then \(R\alpha = v(u_1, \ldots, u_d)R\). But \(R\alpha = R\) and, therefore, \(v(u_1, \ldots, u_d)\) is in \(R\). Since \(R^* = [R, F]R\), it follows that if \(w(a_1, \ldots, a_d)\) is an element of \(R^*\) then \(w(u_1, \ldots, u_d)\) is also an element of \(R^*\).

Assume that \(w_1(a_1, \ldots, a_d)R^* = w_2(a_1, \ldots, a_d)R^*\) where each \(w_i\) is a word in \(F\). Then \(w_2(a_1, \ldots, a_d)^{-1}w_1(a_1, \ldots, a_d)\in R^*.\) Using the result obtained in the previous paragraph, it follows that
\[w_1(u_1, \ldots, u_d)R^* = w_2(u_1, \ldots, u_d)R^*.\]

The mapping \(a^*\) can now be defined as follows: for each word \(w(a_1, \ldots, a_d)\) in \(F\), put
\[w(a_1, \ldots, a_d)R^*a^* = w(u_1, \ldots, u_d)R^*.\]
Clearly, \(a^*\) is a homomorphism and it remains to show that it is onto. But \(a\) is an automorphism of \(F/R\) and \(a_iR\alpha = u_iR\); therefore, \(F/R^*\) is generated by \(\{u_1R^*, \ldots, u_dR^*, R/R^*\}\). Since \(R/R^* \leq P_c(F/R^*)\), it follows that
\[F/R^* = \langle \alpha_1R^*a^*, \ldots, a_dR^*a^* \rangle.\]
Hence, \(a^*\) is an automorphism of \(F/R^*\).

While \(a^*\) is not uniquely determined by \(a\), its restriction to \(R/R^*\) is. This is established by the following argument. Assume that \(a_iR\alpha = u_iR\); then \(F/R^*\) is generated by \(\{u_1R^*, \ldots, u_dR^*, R/R^*\}\). Since \(R/R^* \leq P_c(F/R^*)\), it follows that
\[R^* = \langle \alpha_1R^*a^*, \ldots, a_dR^*a^* \rangle.\]

It remains to establish that \((M_1/R^*)\alpha^*\) is equal to \(M_2/R^*\). Let \(w(a_1, \ldots, a_d)\) be an element of \(M_1\) and let \(\hat{\alpha}^*\) denote the restriction of \(\alpha^*\) to \(R/R^*\); then
\[w(a_1, \ldots, a_d)R^*\hat{\alpha}^* = w(b_1, \ldots, b_d)R^*.\]
It is now shown that \(w(b_1, \ldots, b_d)\in M_2:\)
\[w(b_1, \ldots, b_d)M_2 = w(b_1M_2, \ldots, b_dM_2)\]
\[= w(a_1, \ldots, a_dM_1)\phi\]
\[= w(a_1, \ldots, a_d)M_1\phi\]
\[= M_1 \phi\]
\[= M_2.\]

It follows that \((M_1/R^*)\hat{\alpha}^*\) is a subgroup of \(M_2/R^*\) and, since both have the same index in \(F/R^*\), they are equal.
LEMMA 2.6. Every automorphism $\alpha$ of $F/R$ extends to an automorphism $\alpha^*$ of $F/R^*$ and the restriction of $\alpha^*$ to $R/R^*$ is uniquely determined by $\alpha$.

PROOF. The results follow from Theorem 2.5, where both $M_1/R^*$ and $M_2/R^*$ are chosen to be the $p$-multiplicator, $R/R^*$.

The automorphism $\alpha^*$ is called an extended automorphism.

LEMMA 2.7. Each extended automorphism $\alpha^*$ induces a permutation of the allowable subgroups.

PROOF. The nucleus, $P_e(F/R^*)$, of $G$ is characteristic and the $p$-multiplicator, $R/R^*$, is fixed by $\alpha^*$. Let $M/R^*$ be an allowable subgroup. Then

$$(M/R^*)\alpha^*P_e(F/R^*) = (M/R^*)P_e(F/R^*)\alpha^* = R/R^*,$$

showing that $(M/R^*)\alpha^*$ is an allowable subgroup. Clearly, the mapping is one-to-one and onto and is, therefore, a permutation.

The permutation of the allowable subgroups induced by $\alpha^*$ is denoted by $\alpha'$ and, as in the case of the restriction of $\alpha^*$ to the $p$-multiplicator, $\alpha'$ depends only on the automorphism $\alpha$ of $G$. Let $P$ be the permutation group generated by the $\alpha'$ corresponding to the automorphisms $\alpha$ of $G$. The mapping $\alpha \mapsto \alpha'$ is a homomorphism from $\text{Aut} \ G$ onto $P$.

The following theorem is fundamental in determining the equivalence classes of the allowable subgroups.

THEOREM 2.8. The orbits of the allowable subgroups under the action of $P$ are exactly the equivalence classes of the allowable subgroups.

PROOF. Let $M_1/R^*$ and $M_2/R^*$ be allowable subgroups in the same equivalence class; then $F/M_1$ and $F/M_2$ are isomorphic. By Theorem 2.5, there exists an automorphism, $\alpha^*$, of $F/R^*$ which maps $M_1/R^*$ to $M_2/R^*$ and $\alpha^*$ induces a permutation $\alpha'$ of the allowable subgroups. Thus, $(M_1/R^*)\alpha' = M_2/R^*$ showing that $M_1/R^*$ and $M_2/R^*$ lie in the same orbit.

In order to establish the converse, it is simpler to use the following general result, which can be established easily. Let $N$ be a normal subgroup of a group $H$ and let $\gamma$ be an automorphism of $H$; then $H/N \cong H/N\gamma$. Now, let $M_1/R^*$ and $M_2/R^*$ be allowable subgroups that are elements of the same orbit under $P$. Then there exists a permutation, $\alpha'$, of the allowable subgroups such that $(M_1/R^*)\alpha' = M_2/R^*$. This permutation is induced by an automorphism, $\alpha^*$, of $F/R^*$ where $(M_1/R^*)\alpha^* = M_2/R^*$. Using the general result, it follows that there is an isomorphism from $F/M_1$ to $F/M_2$.

An irredundant list of immediate descendants of $G$ is now obtained by choosing a representative of each orbit of $P$ and constructing the corresponding factor group. Each factor group is a representative of a different isomorphism type.

Thus, given a $p$-group $G$, the $p$-group generation algorithm produces a complete and irredundant list of its immediate descendants. In order to iterate the algorithm by applying it to the capable, immediate descendants of $G$, a generating set for the automorphism
group of each immediate descendant is required. The automorphism information is calculated as a part of the generation algorithm.

**Definition 2.9.** The stabilizer, $S_{M/R^*}$, of an allowable subgroup $M/R^*$ is the group of automorphisms $\langle \zeta \in \text{Aut } G : (M/R^*)\zeta^* = M/R^* \rangle$.

Let $\zeta$ be an element of $S_{M/R^*}$ and let $\zeta^*$ be an arbitrary extension of $\zeta$ to an automorphism of $F/R^*$. Then $\zeta^*$ fixes $M/R^*$ and, hence, its restriction to an immediate descendant, $F/M$, of $G$ can be calculated. The automorphism group of $F/M$ is described in the following theorem.

**Theorem 2.10.** Let $S$ consist of the restriction to $F/M$ of one $\zeta^*$ for each automorphism $\zeta$ in $S_{M/R^*}$ and let $V$ be the group of all automorphisms of $F/M$ whose restriction to $G$ is the identity. Then $\text{Aut } F/M = SV$.

**Proof.** Let $\gamma$ be an automorphism of $F/M$. For each $i \in \{1, \ldots, d\}$, choose a representative $u_i$ in $F$ of the coset $a_iM\gamma$; then $a_iM\gamma = u_iM$. By Property 1, $\gamma$ fixes $P_r(F/M)$ which equals $R/M$; therefore, $\gamma$ can be restricted to $F/R$ to give an automorphism, $\zeta$. If $\zeta$ equals the identity then $\gamma \in V$. Otherwise, consider an extension, $\zeta^*$, of $\zeta$ to $F/R^*$ where 

$$w(a_1, \ldots, a_d)R^*\zeta^* = w(u_1, \ldots, u_d)R^*.$$  

It is now shown that $\zeta^*$ stabilises $M/R^*$. Let $w(a_1, \ldots, a_d)$ be an element of $M$; then 

$$w(u_1, \ldots, u_d)M = w(u_1M, \ldots, u_dM) = w(a_1M\gamma, \ldots, a_dM\gamma) = w(a_1, \ldots, a_d)M\gamma = M\gamma = M.$$  

Therefore, $\zeta^*$ stabilises $M/R^*$ and its restriction, $\tilde{\zeta}^*$, to $F/M$ can be calculated. Clearly, $\tilde{\zeta}^*$ is in $S$. It is now established that $\tilde{\zeta}^*$ can be written as a product of $\gamma$ and an element of $V$. By appropriate choice of representatives, $a_iR^*\zeta^* = u_ir_iR^*$ where $r_i \in R$. Then $\tilde{\zeta}^*$ is defined by 

$$a_iM\tilde{\zeta}^* = u_ir_iM.$$  

The set $\{u_1, \ldots, u_d\}$ is a generating set for $F/M$. Define an automorphism, $\theta$, of $F/M$ by 

$$u_iM\theta = u_ir_iM.$$  

The restriction of $\theta$ to $F/R$ is the identity; therefore, $\theta$ is an element of $V$. Clearly, $\tilde{\zeta}^* = \gamma\theta$; it follows that $\gamma = \tilde{\zeta}^*\theta^{-1}$, where $\tilde{\zeta}^* \in S$ and $\theta \in V$.

3. An Implementation of the $p$-Group Generation Algorithm

In any implementation of the $p$-group generation algorithm, there is a natural division of the calculation of the immediate descendants according to their order. The algorithm implementation is described for the calculation of the immediate descendants having order $p^{s+s}$ of a group of order $p^s$, where $s$ is a positive integer known as the step size. Immediate descendants of order $p^{s+s}$ are sometimes known as $s$-step immediate descendants.
The algorithm takes as input a group description and a generating set for its automorphism group and produces a set of descriptions of the immediate descendants of the group that have a fixed order. The group descriptions are consistent power-commutator presentations.

Let the starting group \( G \) have order \( p^n \), generator number \( d \), and class \( c \). A top-level outline of the algorithm for the construction of its immediate descendants of order \( p^{n+1} \) is the following:

- construct a consistent power-commutator presentation for the \( p \)-covering group of \( G \) and determine its nucleus;
- if the order of the nucleus is less than \( p^n \) then stop;
- for each generator \( \alpha \) of \( \text{Aut } G \)
  - calculate an extended automorphism \( \alpha^* \);
  - calculate the permutation \( \alpha' \) of the allowable subgroups induced by \( \alpha^* \);
  - calculate orbits of the group generated by the permutations \( \alpha' \);
- for each orbit
  - choose a representative;
  - calculate its stabilizer;
  - factor the \( p \)-covering group by the representative allowable subgroup to obtain an immediate descendant;
  - calculate its automorphism group;

In the remainder of this section, refinements of the steps in this algorithm are described.

### 3.1. The \( p \)-Covering Group and Nucleus

Given a consistent power-commutator presentation for \( G \), the \( p \)-covering group algorithm produces a consistent power-commutator presentation for its \( p \)-covering group, \( G^* \). The implementation of the \( p \)-covering algorithm is part of the Havas & Newman implementation of the nilpotent quotient algorithm and a detailed description is provided in Havas & Newman (1980).

The \( p \)-multiplicator is an elementary abelian group and can be viewed as a vector space over the field of \( p \) elements. The number, \( q \), of defining generators of the \( p \)-multiplicator is its rank. These generators, in the consistent presentation of \( G^* \), are named \( a_{n+1}, \ldots, a_{n+q} \).

A feature of the machine implementation is that the generators of the \( p \)-multiplicator are introduced in order by decreasing weight; in adding generators of the same weight, those defined by commutators are added first.

The nucleus of \( G \) can be determined using the following lemma.

**Lemma 3.1.1.** The nucleus of \( G \) is generated by \([a_i, a_j]\) and \( a_j^\alpha \) where \( a_j \) is a generator of weight \( c \), \( i \in \{1, \ldots, d\} \), and \( i < j \).

Thus, when \( G^* \) has been computed, the definitions of the generators can be used to determine the nucleus. Let the nucleus have rank \( r \), where \( 1 \leq r \leq q \). Since the generators of the nucleus have weight \( c+1 \), and the generators are introduced in order by decreasing weight, these generators are \( a_{n+1}, \ldots, a_{n+r} \).
3.2. CALCULATION OF EXTENDED AUTOMORPHISMS

A generating set, \{a_1, \ldots, a_d\}, is supplied for Aut \( G \); each automorphism is described by its action on each of the defining generators, \( a_1, \ldots, a_d \), of \( G \). The exponents of the image of each of these generators under the action of each automorphism are stored.

Let \( \alpha \) be an element of the generating set of Aut \( F/R \). In Section 2, the action of \( \alpha \) on the free group generators \( a_1, \ldots, a_d \) is described by \( a_iR\alpha = u_iR \), for each \( i \in \{1, \ldots, d\} \), where \( u_i \) is a word in the generators \( a_1, \ldots, a_d \). Further, the action of an extended automorphism \( \alpha^* \) on the defining generators of \( G^* \) is given by \( a_iR\alpha^* = u_iR^* \) for \( i \in \{1, \ldots, d\} \). Let \( w(a_1, \ldots, a_{i-1}) \) be the definition of \( a_i \) for \( i \in \{d+1, \ldots, n+q\} \); then put the corresponding \( u_i \) equal to \( w(u_1, \ldots, u_{i-1}) \).

Let \( v(a_1, \ldots, a_d) \) be an arbitrary element of \( R \); then \( v(u_1, \ldots, u_d) \) is an element of \( R \) and, by definition, \( v(a_1, \ldots, a_d)R\alpha^* = v(u_1, \ldots, u_d)R^* \). Since \( \alpha^* \) is an automorphism, \( R/R^* \) is fixed under its action and, modulo \( R^* \), \( u_{s+i} = a_{n+1}^{u_1} \cdots a_{n+q}^{u_n} \), for \( i \in \{1, \ldots, q\} \) where \( 0 \leq \delta \leq p \).

The action of each extended automorphism on the \( p \)-multiplicator of \( G \) is stored as a matrix in order to facilitate the computation of images of the allowable subgroups. The action of \( \alpha^* \) on each generator \( a_{s+i} \) is written as a \( 1 \times q \) vector, where the entries are the image exponents \( \delta_{ij} \); the \( q \) vectors are assembled as a \( q \times q \) automorphism matrix, \( A_{\alpha^*} \).

3.3. A METHOD FOR REPRESENTING ALLOWABLE SUBGROUPS

In order to compute the images of the allowable subgroups under the action of the extended automorphisms, each subgroup is represented by a suitable matrix.

Let \( U \) be a \( u \times v \) matrix. In any non-zero row of \( U \), the first non-zero entry is the leading entry of that row. The matrix \( U \) is left echelonized if it satisfies the following conditions.

(i) The leading entry of every non-zero row is 1.

(ii) Every column containing the leading entry of a row has all other entries zero.

(iii) Each zero row of \( U \) comes below all non-zero rows of \( U \).

(iv) Let there be \( x \) non-zero rows where \( 1 \leq x \leq u \) and let the leading entry of row \( i \) appear in column \( t_i \) for \( i = 1, \ldots, x \). Then \( t_1 < t_2 < \ldots < t_x \).

Every \( u \times v \) matrix is equivalent under elementary row operations to a unique left echelonized matrix [see, for example, Birkhoff & MacLane (1965, p. 165)].

The allowable subgroups for a fixed step size \( s \), where \( 1 \leq s \leq r \), are known as \( s \)-step allowable subgroups and the group generated by the permutations induced on these is denoted by \( P \). By Theorem 2.4, the \( s \)-step allowable subgroups are those subgroups of the \( p \)-multiplicator that have order \( p^{s-q} \) and supplement the nucleus. Thus, the intersection of an \( s \)-step allowable subgroup with the nucleus has order \( p^{r-s} \). Borrowing some notation from linear algebra, the rank of a subgroup of order \( p^{r-s} \) is \( q-s \). The \( s \)-step allowable subgroups are subspaces of rank \( q-s \) that supplement a fixed subspace of rank \( r \) in a space of dimension \( q \).

A one-to-one correspondence is now established between \( s \)-step allowable subgroups and some \( s \times q \) left echelonized matrices.

Let \( M/R^* \) be an arbitrary \( s \)-step allowable subgroup. Recall that the nucleus, \( N/R^* \), has basis \( \{a_{s+1}, \ldots, a_{n+r}\} \). The representation of the allowable subgroups is described relative to this fixed basis. The intersection of \( M/R^* \) with the nucleus has rank \( r-s \). An ordered set, \( K_{M/R^*} \), consisting of \( s \) elements of \( \{a_{s+1}, \ldots, a_{n+r}\} \) that together with this intersection
generates the whole of $N/R^*$ is calculated. This set spans a complement in $N/R^*$ of the intersection of $M/R^*$ with $N/R^*$. Figure 1 illustrates the situation.

A procedure for calculating such an ordered set is now described. Let $U = M/R^*$. Consider the subgroup $\langle U, a_{n+1} \rangle$; if it is larger than $U$ then $a_{n+1}$ is the first element of $K_{M/R^*}$. Now, put $U$ equal to this subgroup and repeat this step with each of $a_{n+2}, \ldots, a_{n+r}$ in turn to obtain the $s$ elements of $K_{M/R^*}$. At the $j$th step, the subgroup is $\langle U, a_{n+j} \rangle$ and $K_{M/R^*} = \{a_{n+k_1}, \ldots, a_{n+k_s}\}$, where each $k_1 < j$. If $\langle U, a_{n+j} \rangle \neq U$, then add $a_{n+j}$ to $K_{M/R^*}$; else $a_{n+j} = w_{a_{n+k_1}} a_{n+k_1} \ldots a_{n+k_s}$, where $w \in M/R^*$ and $0 \leq \xi_i < p$. The ordered set $K_{M/R^*} = \{a_{n+k_1}, \ldots, a_{n+k_s}\}$, where each $k_i \leq r$, is a definition set of $M/R^*$.

A linear transformation, $\theta$, is defined from $R/R^*$, a space of dimension $q$, onto the subspace of dimension $s$ spanned by $K_{M/R^*}$. The map $\theta$ acts as the identity on the elements of $K_{M/R^*}$; for each $a_{n+j}$ not in $K_{M/R^*}$,

$$a_{n+j} \theta = a_{n+k_1} \ldots a_{n+k_s},$$

where the exponents $\xi_i$ for $k_i > j$ are zero. Therefore, $M/R^*$ is the kernel of $\theta$.

The $s \times r$ initial segment submatrix of an $s \times q$ matrix consists of the first $r$ columns of the matrix. The matrix of $\theta$ relative to the basis $\{a_{n+1}, \ldots, a_{n+q}\}$ is a left echelonized $s \times q$ matrix, which has an $s \times r$ initial segment submatrix having rank $s$. Thus, each allowable subgroup can be identified with a left echelonized $s \times q$ matrix, which has an $s \times r$ initial segment submatrix having rank $s$. The matrix is the standard matrix of the allowable subgroup.

Given an arbitrary left echelonized $s \times q$ matrix which has an $s \times r$ initial segment submatrix having rank $s$, it defines a linear transformation from the $p$-multiplicator with its usual basis to a space of dimension $s$ and the kernel of this transformation is an $s$-step allowable subgroup. If the procedure described above is applied to this subgroup, it is easy to see that the definition set obtained is the same as that which can be read off from the matrix. Thus, each allowable subgroup has only one definition set.

The allowable subgroups are constructed as kernels rather than as images because, in this way, each is represented as an $s \times q$ matrix rather than as a $(q-s)\times q$ matrix. This
choice is advantageous since, in practice, the value of $s$ is small and is, generally, less than
that of $q-s$.

In practice, the standard matrices of the allowable subgroups are written down in a
linear order. A definition set is an ordered $s$ element subset of $\{a_{n+1}, \ldots, a_{n+s}\}$ and the
number of such sets is $\binom{r}{s}$. The definition sets are chosen in a lexicographic order and all of
the allowable subgroups determined by a particular definition set are also ordered. In
order to write down explicitly the permutations of the allowable subgroups, a label is
associated with each standard matrix and, hence, each allowable subgroup. The
enumeration of the subgroups is now reviewed briefly and the label for each subgroup
defined.

A basis consisting of $q-s$ generators for each allowable subgroup is obtained by
choosing a definition set, $K = \{a_{n+k_1}, \ldots, a_{n+k_s}\}$, from the $r$ generators of the nucleus. Let
$a_{n+j}$ be an element of $\{a_{n+1}, \ldots, a_{n+q}\}$ that is not contained in $K$. For each of these $q-s$
genrators, an element $h_j$ that has the following definition can be written down:

\[ h_j = a_{n+k_1} \cdots a_{n+k_s} a_{n+j} \]  

where, for each $a_{n+k_i}$ in $K$,

\[ \xi_{ij} \in \begin{cases} \{0, \ldots, p-1\} & \text{if } k_i < j \\ \{0\} & \text{otherwise.} \end{cases} \]

These $q-s$ elements, $h_j$ for $a_{n+j} \notin K$, generate an $s$-step allowable subgroup.

Consider the standard matrices of the allowable subgroups determined by the chosen
definition set $K$. The elements of $K$ determine the positions of those entries of the standard
matrices whose values are fixed—either 0 or 1—and those whose values range over
$\{0, \ldots, p-1\}$. The latter positions are available. The number of available positions in row $l$
of one of these matrices is $q-k_l-(s-l)$. One method of counting the number of allowable
subgroups having definition set $K$ is to count the number of available positions. Thus, the
number of allowable subgroups determined by $K$ is $p^{x(K)}$ where

\[ x(K) = qS - \sum_{i=1}^{q} k_i - s(s-1)/2. \]

The total number of allowable subgroups or, equivalently, the degree, $D$, of the
permutation group $P$ is:

\[ D = \sum_K p^{x(K)}. \]

The label of an allowable subgroup is a positive integer in $\{1, \ldots, D\}$. Let $M/R^*$ be an
allowable subgroup and let $S = (\xi_{ij})$ be its standard matrix. For the purpose of defining the
label of $S$, the definition set, $K$, of $S$ is written as $\{k_1, \ldots, k_s\}$. The lexicographic ordering of
the definition sets is now formally defined.

**Definition 3.3.1.** Let $K$ and $K^*$ be two definition sets; then $K > K^*$ if there exists an $l$ in
$\{1, \ldots, s\}$ where $k_l > k^*_l$ and $k_i = k^*_i$ for $1 \leq i < l$.

The unique label for $S$ has two components. The first is the offset $O_k$, which is the
number of standard matrices determined by definition sets that occur earlier in the linear
ordering:

\[ O_K = \sum_{K^* < K} p^{x(K^*)}. \]
The second component is the position of $S$ relative to $K$. A position function is defined on the available positions and its range is \{0, \ldots, x(K) - 1\}. The value 0 is assigned to the leftmost available position in the first row of the matrix. The assigned value increases across the available positions in each row in turn; the value $x(K) - 1$ is assigned to the right-most available position in the last row having an available position. Let $y(i,j)$ be the value of the position function for the available position $(i,j)$ in $S$. More formally, $y(i,j)$ is given by the following equation

$$y(i,j) = \sum_{l=1}^{t} (q - k_l - (s - 1)) - |\{t : j \leq t \leq q, t \notin K\}|.$$

**Definition 3.3.2.** The label $L$ of $S$ is defined by the following equation

$$L = \sum_{i=1}^{s} \sum_{j} \xi_{ij} p^{R(i,j)} + O_K + 1$$

where $\xi_{ij}$ is the $(i,j)$ entry in $S$ and, for each $i$, the second summation is over $j$ such that $k_i < j \leq q$ and $j \notin K$.

### 3.4. Calculation of Permutations

The labelling scheme is chosen, in part, to facilitate the calculation of the permutations of the allowable subgroups which is now discussed. Let $\alpha$ be an automorphism of $G$; recall that the action of the extended automorphism $\alpha^*$ on the $p$-multiplicator is represented by a $q \times q$ automorphism matrix $A_{\alpha^*} = (\delta_{ij})$ where, using additive notation,

$$a_{n+i} \alpha^* = \sum_{j=1}^{q} \delta_{ij} a_{n+j}.$$

The equivalence relation defined on the allowable subgroups provides an equivalence relation on the standard matrices of the allowable subgroups.

Recall that an allowable subgroup, $M/R^*$, is the kernel of a map, $\theta$, from the $p$-multiplicator to the space which has as its basis the definition set of $M/R^*$ and the matrix of this map is the standard matrix, $S$. Then $(M/R^*)\alpha^* - 1$ is the kernel of the map $\alpha^* \theta$. It is now shown that the matrix of the product $\alpha^* \theta$ is given by $SA_{\alpha^*}$: let $a_{n+i}$ be an element of the basis of the $p$-multiplicator, then

$$(a_{n+i} \alpha^*) \theta = \sum_{j=1}^{s} \delta_{ij} (a_{n+j} \theta)$$

$$= \sum_{j=1}^{s} \delta_{ij} \sum_{l=1}^{s} \xi_{ij} a_{n+k_l}$$

$$= \sum_{l=1}^{s} \left( \sum_{j=1}^{s} \xi_{ij} \delta_{ij} \right) a_{n+k_l}.$$
The permutations of the allowable subgroups induced by the inverses of the automorphisms \( \sigma_1^*, \ldots, \sigma_m^* \) generate the same group as the automorphisms themselves. Take the smallest set of \( s \times q \) matrices that contains \( S \) and is closed under right multiplication by the transposes of \( A_{\sigma_1^*}, \ldots, A_{\sigma_m^*} \). Left echelonization of the matrices in this set gives exactly the standard matrices corresponding to the allowable subgroups in the orbit of \( M/R^* \).

These facts are used in the calculation of the permutations induced by the extended automorphisms. The standard matrices of the allowable subgroups are written down and their products by the transposes of the automorphism matrices calculated. In practice, the product \( A_{\sigma^*}S^T \) is computed and its transpose is left echelonized. The label of the standard matrix obtained is then computed and stored. Thus, the machine implementation constructs the inverse of the permutation induced by the extended automorphism \( \sigma^* \).

The labelling of the matrices representing the allowable subgroups is arranged so that the results of the matrix multiplications can be obtained by adding columns of the automorphism matrix. If \( A \) is a \( q \times q \) matrix and \( v \) a \( 1 \times q \) vector, \((v_1, \ldots, v_q)\), the product \( A(v_1, \ldots, v_q)^T \) equals \((A_1)v_1 + \ldots + (A_q)v_q\) where \( A_i \) is the \( i \)th column of \( A \).

The application of this in building up the images is best illustrated by example: for a particular step size, say \( 2 \), and for \( p \neq 2 \). The standard matrices are processed by increasing label; thus, the first definition set chosen is \( \{a_{n+1}, a_{n+2}\} \). The standard matrix, \( S \), corresponding to the allowable subgroup labelled 1 is

\[
\begin{pmatrix}
1 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0
\end{pmatrix}
\]

The product \( AS^T \) equals \((A_1, A_2)\). The standard matrix, \( S \), corresponding to the allowable subgroup labelled 2 is

\[
\begin{pmatrix}
1 & 0 & 1 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0
\end{pmatrix}
\]

In this case, the product \( AS^T \), which equals \((A_1 + A_3, A_3)\), can be obtained by adding the column \( A_3 \) to \( A_1 \) in the previously computed product, \((A_1, A_3)\). To compute the required product for the standard matrix corresponding to the allowable subgroup labelled 3, it is only necessary to add the column \( A_3 \) to the sum \( A_1 + A_3 \).

### 3.5. ORBIT AND STABILISER CALCULATIONS

If the automorphism group of \( G \) is soluble, it can be described using a PAG-presentation, in which a sequence of generators is supplied relative to a composition series for the group. Presentations of this type were introduced by Jürgensen (1970) who used the term AG-system to describe them. The calculation of the orbits in this case is based on an algorithm of Leedham-Green which is described in Laue, Neubüser & Schoenwaelder (1984). Each element of the automorphism group can be written uniquely as a product of the supplied defining generators and this fact is used in computing a stabiliser for each orbit representative.

If the automorphism group is insoluble, the system CAYLEY (see Cannon, 1984) is used to perform these calculations. The orbit algorithms implemented in CAYLEY are described in Butler (1984, Part II, Chapter 1). The insoluble automorphism group case is handled using an interface between the implementation of the \( p \)-group generation
algorithm and this system. The permutations of the allowable subgroups are calculated and transferred to CAYLEY, where orbits and their representatives are determined.

The general purpose stabiliser algorithms present in CAYLEY can be used to calculate the stabiliser of an orbit representative in the permutation group. However, the determination of the generators of the stabiliser as words in the defining generators of the permutation group, and hence their recognition as words in the defining generators of Aut G, remains a problem. One reasonable method of calculating the stabiliser is to build up systematically the orbit of the representative under P and to keep track of stabiliser words. The order of the stabiliser is known in advance since the size of the orbit is known. When sufficient generators for the stabiliser have been obtained, the process terminates.

If the stabiliser of an orbit representative is soluble, the corresponding immediate descendant has a soluble automorphism group. In iterating the p-group generation algorithm, it is preferable to use the more efficient machinery for computations with soluble automorphism groups. Thus, the solubility of the stabiliser of each orbit representative is checked and, where appropriate, a CAYLEY procedure is used to calculate a PAG-generating sequence for the stabiliser. Each orbit representative and a generating set for its stabiliser are written to a file and this information is transferred back to the implementation of the p-group generation algorithm.

3.6. SETTING UP THE IMMEDIATE DESCENDANTS

Recall that the leading term of each orbit is chosen as its representative. The representatives are organised as a list in which they are ordered by increasing label. This list determines the sequence in which descriptions of the immediate descendants are produced. A generating set for the allowable subgroup determined by a representative can be written down easily and a presentation for the appropriate factor group calculated. The new generators introduced have weight c + 1 and have definitions as commutators or pth powers; therefore, the presentation obtained is a weighted consistent power-commutator presentation.

The description of the automorphism group required for iteration of the algorithm is now discussed. If an immediate descendant is terminal, no description of its automorphism group is required. In order to save space, usually only descriptions of the capable groups are saved to a data file for later access. Thus, the p-covering group of each immediate descendant, H, is constructed and the nuclear rank of H determined. If H is capable, the automorphism group of H is calculated, using Theorem 2.10.

The performance of the implementation depends heavily on the number of automorphism group generators, as this determines the number of permutations that must be constructed and also affects the time taken in the calculation of orbits. The performance can be improved by noting that if the action of an extended automorphism on the p-multiplicator of a group is trivial, it induces the trivial permutation on the allowable subgroups and plays no role in determining the orbits of the allowable subgroups.

Lemma 3.6.1. The extensions of the inner automorphisms of G act trivially on the p-multiplicator.

Thus, the automorphism group description supplied to the implementation is a set of automorphisms that together with the inner automorphisms generates the automorphism group. Such a description is supplied by a user for the automorphism group of the starting group. When the automorphism group of an immediate descendant H is being calculated,
the implementation discards the inner automorphisms induced by the generators of weight $c$ in $H$.

The group $V$, of Theorem 2.10, has a generating set $\{\theta_{ij}\}$ where each $\theta_{ij}$ is defined as follows:

$$\theta_{ij}: a_i \mapsto a_ia_{n+j} \quad \text{for } i \in \{1, \ldots, d\}, j \in \{1, \ldots, s\},$$
$$a_k \mapsto a_k \quad \text{for } k \in \{1, \ldots, d\}\backslash\{i\}.$$  

The $ds$ generators of $V$ obtained in this way may not all be required in the iteration of the algorithm, as some combinations of these generators may be inner automorphisms of $H$. First, the commutators of the generators of weight $c$ in $H$ with the defining generators of $H$ are calculated. Let $a_i$ be a generator of weight $c$ and let $a_i$ be one of the defining generators; then $[a_i, a_j] = a_{n+1}^{\lambda_1} \ldots a_{n+s}^{\lambda_s}$, where $0 \leq \lambda_j < p$ for $j = 1, \ldots, s$. If the commutator is non-trivial, one of the non-zero exponents in the result, say $\lambda_j$, can be chosen and the corresponding generator $\theta_{ij}$ may be discarded.

In order to remove as many generators from $\{\theta_{ij}\}$ as possible, a standard echelonization is carried out. Let $x$ be the number of generators of $H$ having weight $c$. The exponents of the words obtained from the commutator calculations are assembled as an $x \times ds$ matrix which is echelonized from left to right. Automorphisms that can be removed are determined using the entries in the echelonized matrix. The first non-zero entry (if any) in each row of the matrix determines an automorphism that can be removed, since this automorphism can be obtained as a combination of inner automorphisms and those generators of $V$ that are retained.

The supplied description of Aut $G$ is a set of automorphisms that together with the inner automorphisms of $G$ generates Aut $G$. The stabilizer that is computed for an allowable subgroup, $M/R^*$, is the stabilizer of $M/R^*$ in the group generated by the supplied set of automorphisms. The computed stabilizer together with the inner automorphisms of $G$ generates the stabilizer of the allowable subgroup in Aut $G$. For each generator $\zeta$ of the computed stabilizer, the action of an extended automorphism $\zeta^*$ on each of the defining generators of $G^*$ is first computed and the restriction of $\zeta^*$ to $H$ is then determined. The remaining $\theta_{ij}$s and the restrictions of the $\zeta^*$s to $H$ form a generating set for Aut $H$ modulo the inner automorphisms of $H$.

The previously computed power-commutator presentation of $H^*$, as well as the automorphism information, is now written to a data file where it may be accessed as required. The next iteration of the algorithm begins by calculating the actions of the extended automorphisms on the $p$-multiplicator of $H$.

Throughout this discussion, the value of the step size was fixed. When a new step size is chosen, the relevant steps of the algorithm are repeated. Options are provided that permit a user to select a range of step sizes. The construction of all immediate descendants of the starting group is the default for the implementation.

### 4. Characteristic Subgroups in the $p$-Multiplicator

Some practical limitations of the implementation described in the previous section are now discussed. The internal structure of the $p$-multiplicator is used to develop a modification of the basic algorithm that removes some of these limitations.

The performance of the implementation depends heavily on the number of allowable subgroups of a particular step size; that is, the degree, $D$, of the permutation group $P$ constructed. The generating permutation induced by each extended automorphism is
temporarily stored in image form in an integer array, having dimension $D$, before being used in the calculation of the orbits. The storage requirement for this array is a limiting factor on the implementation. As an example, in calculating the 2-step immediate descendants of the elementary abelian group of order 16, which has a 2-multiplicator and nucleus of rank 10, the degree of $P$ is 174251. The time taken in calculating the generating permutations for $P$ and in determining its orbits is an additional limiting factor.

The algorithm performance can be improved by using structural features of the $p$-multiplicator, such as the presence of characteristic subgroups of the $p$-covering group. In general, the structure of the $p$-multiplicator allows the set of allowable subgroups to be divided up into a number of smaller sets that are unions of orbits. These divisions allow the construction of permutation groups of smaller degree.

The use of characteristic subgroups in the $p$-multiplicator for this purpose is now described in the cases of practical interest. Let the $p$-multiplicator, $R/R^*$, have rank $q$. Let $C/R^*$ be a proper, non-trivial, characteristic subgroup of $G^*$ in the $p$-multiplicator and let $C/R^*$ either contain the nucleus, $N/R^*$, of $G$ or be contained in $N/R^*$. The nucleus of $G$ relative to $C/R^*$ may be defined as the intersection of $N/R^*$ with $C/R^*$. In the first of these cases, the relative nucleus is $N/R^*$ and, in the second case, the relative nucleus is $C/R^*$. Similarly, allowable subgroups relative to $C/R^*$ may be defined as the intersection of the allowable subgroups of the $p$-multiplicator with $C/R^*$. The relative allowable subgroups supplement the relative nucleus in $C/R^*$. Figures 2 and 3 illustrate the two cases that arise.

The generation algorithm may first be applied using the subgroup $C/R^*$. The allowable subgroups relative to $C/R^*$ can be described by choosing relative definition sets. The orbits of the relative allowable subgroups under the actions of the extended automorphisms are calculated, the stabiliser for each orbit representative constructed, and the appropriate factor groups constructed. Each of the factor groups constructed is a reduced $p$-covering group. Since $C/R^*$ is a proper subgroup of $R/R^*$, the degrees of the permutation groups constructed are smaller than that obtained by applying the algorithm to the whole of the $p$-multiplicator.

Let $M/R^*$ be an allowable subgroup of $R/R^*$ such that $M/R^* \cap C/R^*$, an allowable subgroup relative to $C/R^*$, is an orbit representative. Denote $M/R^* \cap C/R^*$ by $M_1/R^*$. Then $F/R^*$ is factored by $M_1/R^*$ to obtain the reduced $p$-covering group $F/M_1$. The subgroup $R/M_1$ is the reduced $p$-multiplicator of $F/M_1$ and $NM_1/M_1$ is the nucleus of the reduced $p$-covering group. The stabiliser of $M_1/R^*$ is called the stabiliser of $F/M_1$. The $p$-multiplicator rank of each reduced $p$-covering group is less than $q$.

Each reduced $p$-covering group, $F/M_1$, is now processed. The allowable subgroups of the reduced $p$-multiplicator that supplement the nucleus of $F/M_1$ can be described and their orbits under the action of the stabiliser of $F/M_1$ calculated. On factoring $F/M_1$ by the orbit representatives, immediate descendants of $G$ are obtained. The calculation of immediate descendants of a group by processing each reduced $p$-covering group, in turn, is known as intermediate stage calculations.

**Lemma 4.1.** Every immediate descendant of $G$ is a factor group of one of the reduced $p$-covering groups. The isomorphism types of the immediate descendants are determined by the orbits of the allowable subgroups of each reduced $p$-covering group under the action of its stabiliser.

In the previous discussion, the characteristic subgroup chosen is a subgroup that either
contains the nucleus of $G$ or is contained in the nucleus. These are the cases of practical interest, but the discussion also applies to any characteristic subgroup in the $p$-multiplicator where the allowable subgroups relative to this characteristic subgroup supplement the relative nucleus.

In the above description, a characteristic subgroup is chosen in the $p$-multiplicator, orbits of the allowable subgroups relative to this subgroup computed, and reduced $p$-covering groups constructed. Each reduced $p$-covering group is then processed in turn. The orbits of allowable subgroups of its reduced $p$-multiplicator are computed and factor groups constructed to give immediate descendants. However, the selection of a characteristic subgroup and the computation of orbits of allowable subgroups relative to this subgroup may be iterated.
Let $F/M_1$ be a reduced $p$-covering group where $M_1/R^*$ is a representative allowable subgroup relative to the characteristic subgroup chosen at the first stage. A subgroup in the reduced $p$-multiplicator of $F/M_1$ is chosen that is characteristic under the action of the stabiliser and contains the nucleus of $F/M_1$ or is contained in it. This subgroup must contain the definition set of $M_1/R^*$. The allowable subgroups relative to this subgroup are now described, their orbits computed under the action of the stabiliser, and factor groups (that is, reduced $p$-covering groups) constructed. In turn, a suitable characteristic subgroup in the reduced $p$-multiplicator of each of these reduced $p$-covering groups can be determined and the computations iterated. When the chosen characteristic subgroup equals the reduced $p$-multiplicator of a reduced $p$-covering group, immediate descendants of $G$ are obtained and the iteration terminates.

The implementation of the algorithm for the construction of immediate descendants by intermediate stage calculations is now described. It is possible to design an implementation so that any characteristic subgroup that contains the nucleus or is contained in it can be chosen. However, such an implementation would require a change of basis to be performed on the representative allowable subgroups in order to factor them from the $p$-covering group. A simpler alternative is to choose a characteristic subgroup that is also an initial segment in the $p$-multiplicator.

**Definition 4.2.** Let $k$ be an element of $\{0, \ldots, q-1\}$. The $k$-initial segment subgroup in the $p$-multiplicator is $\langle a_{n+1}, \ldots, a_{n+k+1} \rangle$.

In the implementation, the subgroups chosen are characteristic, initial segment subgroups that have the smallest possible rank. This choice permits the reduced $p$-covering groups to be constructed easily and ensures that space requirements are kept to a minimum.

Two natural divisions in the implementation arise, depending on whether the characteristic subgroup chosen contains the nucleus or not.

Let $s$ be a fixed step size. Initially, the characteristic, 0-initial segment subgroup, $C$, in the $p$-multiplicator is calculated using the following method: initialise $C$ to equal $\langle a_{n+1} \rangle$; check the action of each extended automorphism on $a_{n+1}$ and, if necessary, add new generators to $C$; now, apply the automorphisms to the new generators until $C$ is characteristically closed. This calculation is easy, since the action of each extended automorphism on each of the generators of the $p$-multiplicator has been calculated. Let $t$ be the rank of $C$.

The case where the nucleus is contained in $C$ (that is, $t$ is at least $r$) is first discussed. The notation $\hat{G}^*$ and Aut $\hat{G}$ is used to denote a reduced $p$-covering group and the automorphism group acting, respectively. Note that these variables and $C$ are updated in the course of the following description.

1. Set $\hat{G}^*$ equal to $G^*$ and Aut $\hat{G}$ equal to Aut $G$.
2. The orbits of the allowable subgroups relative to $C$, under the actions of the extensions of the generators of Aut $\hat{G}$, are determined and the stabilisers of the orbit representatives are calculated, as outlined in Section 3 above.
3. For each orbit representative in turn, $\hat{G}^*$ is factored by the appropriate allowable subgroup, giving a reduced $p$-covering group whose reduced $p$-multiplicator rank has been decreased by $t-s$.
4. If $C$ is the reduced $p$-multiplicator of $\hat{G}^*$, the reduced $p$-covering groups obtained in step (3) are the immediate descendants of the starting group. The generators of the
group $V$, of Theorem 2.10, are calculated for each capable descendant and the algorithm terminates.

(5) Otherwise, $C$ is a proper subgroup of the reduced $p$-multiplicator of $G^*$. For each orbit representative, the variable $\hat{G}^*$ is set equal to the corresponding reduced $p$-covering group obtained in step (3) and $\text{Aut } G$ is set equal to the corresponding stabiliser. The characteristic, $s$-initial segment subgroup, $C$, is now determined. Let it have rank $t$. Steps (2) to (5) are repeated for $\hat{G}^*$.

The second case, where the characteristic, $0$-initial segment subgroup is properly contained in the nucleus, is more complex.

(1) Set $y_1$ equal to $s + t - r$ and $y_2$ equal to the minimum of $s$ and $t$.

(2) For each step size, $s'$, running from $y_1$ to $y_2$, reduced $p$-covering groups are constructed. Each reduced $p$-covering group has a nucleus whose rank $r = r + s' - t$. If the step size is 0, the (reduced) $p$-covering group is factored by the whole of the characteristic subgroup.

(3) Each of the reduced $p$-covering groups is processed in turn. First, determine the characteristic, $s'$-initial segment subgroup $C$, which has rank $t$. If $t$ is at least $r$ then proceed as outlined in the first case. Otherwise, the intersection of an allowable subgroup relative to $C$ with the subgroup generated by the first $s'$ generators of the nucleus must be trivial. Therefore, in describing the allowable subgroups relative to $C$, the definition sets used in eqn (1) of Section 3.3 must be selected to satisfy this condition. The parameter $y_1$ now takes the maximum of $s'$ and $s + t - r$ and $y_2$, as before, takes the minimum of $s$ and $t$. Steps (2) and (3) are now repeated.

The storage requirement for the construction of immediate descendants of a group using intermediate stage calculations is usually significantly smaller than that required in a single iteration of the algorithm. As an example, in calculating 2-step immediate descendants of the elementary abelian group of order 16, the largest degree of a permutation group constructed is only 651 while a permutation group of degree 174,251 is constructed in a single iteration. If the rank of the $p$-multiplicator is greater than about 10, the cumulative time taken for the intermediate stage calculations is usually less than that taken by a single iteration over the full $p$-multiplicator because of the difference in time taken to access storage locations. Thus, the application range of the algorithm has been significantly extended.

In the implementation, an option is provided that allows a user to select a characteristic, initial segment subgroup. This option has been used to help demonstrate the internal consistency of the implementation. Information on orbit sizes and the number of immediate descendants obtained by selecting subgroups of different rank has been compared and agreement found. The subgroup chosen by the user is checked for characteristic closure by applying the extended automorphisms to its generators.

The calculation of immediate descendants by intermediate stage calculations, proceeding through characteristic, initial segment subgroups of the smallest possible rank, is the default mode for the implementation.

The use of characteristic, initial segment subgroups in this way was sufficient to permit a complete and independent determination of the groups of order 128; this work is described in James et al. (1990). It also permitted the determination of a majority of the groups of order 256. However, in determining 3-step immediate descendants of the elementary abelian group of order 32 and 2-step immediate descendants of the elementary abelian
group of order 64, permutation groups of very large degree (in excess of 6,000,000) are constructed and a new method was required to complete these calculations. The central idea of this method is to use known information on the orbits of \(s\)-step allowable subgroups to obtain information on the orbits of \((s+1)\)-step allowable subgroups; it is described in detail in O'Brien (1988).

5. An Example of Some Calculations

In this section, some of the immediate descendants of a group of order 16 are calculated. Subject to the convention that all relations whose right-hand side are trivial are not shown, the group, \(G\), has a consistent power-commutator presentation: \(<a_1, \ldots, a_4 : a_4^2 = a_4, [a_2, a_4] = a_3>\). It is isomorphic to a split extension of \(C_2 \times C_2\) by \(C_4\) acting invertingly and has 2-covering group

\[
\langle a_1, \ldots, a_8 : a_1^2 = a_4, a_2^2 = a_8, a_3^2 = a_6, a_4^2 = a_7, [a_2, a_1] = a_3, [a_3, a_1] = a_5, [a_3, a_2] = a_6, [a_4, a_2] = a_5 a_6 \rangle.
\]

It is capable, having nuclear rank 3; its nucleus is \(<a_5, a_6, a_7>\).

A PAG-generating sequence, in reverse order, for the automorphism group is

\[
a_2 \rightarrow a_2 a_5 \rightarrow a_1 a_4 \rightarrow a_1 a_4 a_5 \rightarrow a_1 a_2 a_3.
\]

The automorphism matrix representing the action of \(a_1^\ast\) on the 2-multiplicator of \(G\) is the identity; the matrices, \(A_{a_1}\) and \(A_{a_2}\), are, respectively:

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1
\end{pmatrix}
\]

Some immediate descendants of order 32 are calculated by proceeding through intermediate stage calculations. For the purpose of illustration, the nucleus, \(<a_5, a_6, a_7>\), is chosen. Case I of the algorithm outlined in Section 4 is relevant.

The first definition set is \(\{a_2\}\) and the associated allowable subgroups relative to the characteristic subgroup are \(<a_2^c, a_6, a_2^c a_7>\). Their standard matrices have the form (1, 1, \(\xi_2\)) and labels running from 1 to 4. The second definition set is \(\{a_6\}\) and the relative allowable subgroups are \(<a_5, a_6^c, a_7>\). Their standard matrices have the form (0, 1, \(\xi_1\)) and labels 5 and 6. The third and final definition set is \(\{a_7\}\); its relative allowable subgroup is \(<a_5, a_6>\) which has a standard matrix (0, 0, 1) and label 7. Thus, the induced permutations have degree 7.

Since the action of both \(a_1^\ast\) and \(a_2^\ast\) on the characteristic subgroup is the identity, \(P\) is cyclic, generated by \((2, 6)(4, 5)\), and there are 5 orbits, namely: \(\{1\}\), \(\{2, 6\}\), \(\{3\}\), \(\{4, 5\}\), and \(\{7\}\).

On factoring \(G^*\) by \(<a_6, a_7>\) (which is stabilised by all three automorphisms), the reduced 2-covering group, \(G^*_\ast\), is obtained:

\[
\langle a_1, \ldots, a_6 : a_1^2 = a_4, a_2^2 = a_6, [a_2, a_1] = a_3, [a_3, a_1] = a_5, [a_3, a_2] = a_5, [a_4, a_2] = a_5 a_6 \rangle.
\]

The characteristic, 1-initial segment subgroup equals the whole of the reduced 2-multiplicator of \(G^*_\ast\). The extended automorphism \(a_1^\ast\) acts as the identity on the reduced
2-multiplicator while $a_2$ and $a_3$ have the same action:

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

The two allowable subgroups, $\langle a_2^{-6}, a_4 \rangle$, are interchanged by the extended automorphism, giving one immediate descendant. The subgroup $\langle a_3 \rangle$ is stabilised by $a_1$ and $a_2^2a_3$ and the presentation for the immediate descendant is

$$\langle a_1, \ldots, a_5; a_1^2 = a_4, [a_2, a_1] = a_3, [a_3, a_2] = a_5, [a_4, a_2] = a_5 \rangle.$$

This group is capable, having 2-multiplicator rank 4 and nuclear rank 1. Neither of the generators of $V$ are required for iteration. The remaining four reduced 2-covering groups may be processed similarly to obtain six other immediate descendants of order 32.

Another example of calculations, including the construction of the starting group $G$ for this example, is given in Newman (1977).

6. Implementation Performance and Use

The implementation of the $p$-group generation algorithm is combined with that of the nilpotent quotient algorithm in a single program, which is known as the Nilpotent Quotient Program. The program is written in FORTRAN 77 and is designed for use on virtual memory machines. It is planned to include a complete implementation of the algorithm in Version 4 of CAYLEY.

Both implementations of the algorithm have been used in a number of contexts. Ascione et al. (1977) used the program to generate a list of 3-groups of order at most $3^{10}$. Leedham-Green and Newman have used it to obtain descriptions of 3-groups having nilpotency co-class 2. Work on $p$-groups having nilpotency co-class 1, Newman has used it to obtain descriptions of 5-groups having orders at most $5^{10}$.

Newman and O'Brien have used the implementation in checking the work of James (1980) and others on groups of order $p^6$, for $p$ odd. It was used by Baldwin (1987) to determine a complete list of the 505 groups of order $3^6$.

The implementation was used by James (1983) in work on 2-groups of nilpotency co-class 2. It was also used by Newman and O'Brien to obtain information on 2-groups of nilpotency co-class 3.

The CPU time taken to obtain a complete list of descriptions of the 2328 groups of order 128 using the current implementation on a VAX 8700 is 8 min. For the groups of order 256, the CPU times taken to obtain complete lists of the 540 2-generator groups, the 6190 3-generator groups, and the 20 241 4-generator groups are, respectively, 2, 18, and 60 min. On average, 70% of the CPU time taken in calculating the immediate descendants of a group is spent on computing the generating permutations and orbits. The CPU time taken to calculate the generators for a 2-generator insoluble permutation group of degree 174 251 is 1.5 min. For a soluble group having composition length 16 and degree 3280, the time taken to calculate its generating permutations and to construct its 113 orbits is 12 s. For a soluble group having composition length 16 and degree 11 011, the time taken to construct its generators and 321 orbits is 39 s. For a soluble group having composition length 12 and degree 33 880, the time taken to construct its generators and 5830 orbits is 110 s. All of the above times are rounded and averaged over a number of calculations.
The time taken in calculating the permutations is directly proportional to the degree and the number of generators of the permutation group constructed. The complexity of the algorithm as a whole is determined by the algorithms used in computing the orbits of this permutation group. An analysis of orbit algorithms is provided in Butler (1984, Part II, Chapter 1).

With access to "reasonable" computational time and space resources, the implementation can construct permutations of about 1,000,000 allowable subgroups and calculate the orbits of these subgroups. The implementation has been used in calculating generating permutations for 2-generator insoluble groups having degrees about 2,000,000 and 6,000,000. These computations took about 22 and 120 min of CPU time, respectively.

This work formed part of my PhD, which was obtained from the Institute of Advanced Studies, Australian National University, Canberra. I thank my supervisor, Dr. M. F. Newman, for many helpful discussions on its contents. I also thank the referees for their helpful suggestions.

References


