We describe in some detail an algorithm which is an important ingredient in the performance of programs for working with descriptions of groups of prime-power order. It is an algorithm for computing a useful power-commutator presentation for the $p$-covering group of a group given by a power-commutator presentation. We prove that the algorithm is correct and give some examples.

1. Introduction

It is well known, and was first proved by Sylow (1872), that every group of prime-power order $p^n$ has a presentation of the form

\[ \{a_1, \ldots, a_n \mid a_i^p = v_{ii}, 1 \leq i \leq n, [a_k, a_j] = v_{jk}, 1 \leq j < k \leq n \}, \]

where $v_{jk}$ is a word in the elements $a_{k+1}, \ldots, a_n$ for $1 \leq j < k \leq n$. A presentation of this form is now called a power-commutator presentation for the group and written briefly $\{A \mid R\}$. As was first observed by Neubüser (1961), these presentations provide a good basis for working with groups of prime-power order using computers. The recent text of Sims (1994) provides much general background; we will refer to it with page numbers.

In terms of a generating set of this kind every element of the group can be written uniquely as a normal word, which is a word of the form $a_1^{\xi(1)} \cdots a_n^{\xi(n)}$ with $0 \leq \xi(i) < p$. Sims (1994, p. 395) calls the above words collected. We can assume without loss of generality that the words $v_{jk}$ occurring on the right-hand sides of the relations in a power-commutator presentation are normal.

The fundamental importance of a power-commutator presentation $\{A \mid R\}$ is that given a word in $A$ the relations in $R$ can be used to compute mechanically a normal word representing the same element of the group and hence to compute products and inverses. Power-commutator presentations play a similar core role in the theory of groups of prime-power order to that played by bases and strong generating sets for permutation groups (e.g. Butler, 1991). They provide a comfortable and uniform way of describing the groups in question. This often makes it possible to divide the task of studying groups of prime-power order into three steps—computing a power-commutator presentation from...
another description, computing a power-commutator presentation especially suited to a specific problem from a given power-commutator presentation and extracting information from a power-commutator presentation.

The context here is the development of effective algorithms for computing power-commutator presentations for $p$-quotients of finitely presented groups. Macdonald (1974) was the first to design and implement such an algorithm. Motivations for further developments have included the study of Burnside groups of prime-power exponent (e.g. Havas and Newman, 1980; Newman and O'Brien, 1996) and making lists of groups of prime-power order (e.g. O'Brien, 1990). The algorithmic ideas that have evolved from these and other studies are implemented in the ANU $p$-Quotient Program (Havas et al., 1995). This in turn is incorporated into the systems GAP (Schönherr et al., 1994) and MAGMA (Bosma and Cannon, 1994) and the package Quotpic (Holt and Rees, 1993). A simpler version has also been written in the GAP language (see Celler et al., 1993 or Schönherr et al., 1994, 'The Prime Quotient Algorithm'); the GAP implementation provides, in a reasonably accessible way, further details beyond those in this paper.

There are a number of descriptions in the works already cited of the ideas involved in the relevant algorithms and their implementations. This paper provides a detailed description of an important feature (called the tails routine) which is not described elsewhere and a correctness proof for the main version (Algorithm 3). We set the scene in Section 2 and then go into details in the following two sections. The basic ideas behind tails routines also play a significant role in computations with graded Lie rings (Havas et al., 1990) and graded associative algebras (Vaughan-Lee, 1993).

2. Setting the Scene

The generator number of a $p$-group $G$ is the least integer $d$ such that $G$ can be generated by $d$ and no fewer elements. A $p$-group $G$ with generator number $d$ can be written as a quotient $F/R$ of a free group $F$ of rank $d$. It is well-known that the isomorphism type of the group $F/[F,R]R^p$ depends only on $G$ (e.g. O'Brien, 1990, Lemma 2.3). We denote this group $G^*$ and call it the $p$-covering group of $G$ or, where $p$ is fixed, the covering group of $G$. It has some similarities to the Schur covering groups (e.g. Huppert, 1967, p. 630). A basic algorithmic step in computing power-commutator presentations is to go from a power-commutator presentation for a group $G$ to one for its covering group (it is usually referred to as the $p$-covering algorithm). In practice it is important for the covering algorithm to be efficient. While some of the ideas go back to quite early implementations (see Havas, 1974, pp. 90-91) there has been no detailed description (apart from the programs) and no formal justification of them before the recent theses of Nickel (1993) and Niemeyer (1993) and the report of Celler et al. (1993). Nickel (1993, 1996) generalized the tails routine to presentations for nilpotent groups. The generalisation involves further technicalities, therefore it seems appropriate to consider the $p$-group case here. The details are given in Sections 3 and 4.

Sylow’s Theorem shows how to write down a power-commutator presentation for a $p$-group. In what follows power-commutator presentations are considered as objects in their own right. They always, of course, define a group; that defined by $\langle A \mid R \rangle$ is here denoted $\langle A \mid R \rangle$. However, when $A$ has $n$ elements, the group $\langle A \mid R \rangle$ need not have order $p^n$ (an example is given later).

In what follows some understanding of collection relative to a power-commutator presentation is needed. Here we give a brief outline; more details can be found in Sims
A word in $A$ which is not normal has (at least) one minimal non-normal subword $a_i^q$ or $a_i^{-1}$ or $a_ka_j$ with $j < k$. The relations in $R$ can be used to replace $a_i^q$ by $v_i$ and $a_i^{-1}$ by $a_i^{p-1}v_i^{-1}$ and $a_ka_j$ by $a_i^{-1}a_kv_{jk}$. Of course, the word obtained by a replacement represents the same element of the group as the word it comes from. Collection of a word consists of a sequence of such replacement steps. If a word is normal, the sequence is empty; otherwise the first step replaces one of the minimal non-normal subwords of the given word. This process is then applied to the resulting word.

A fundamental theorem is that every collection (whatever steps it comprises) of a non-normal word results in a normal word after a finite number of replacement steps (e.g. Sims, 1994, p. 396). In other words collection is an algorithm. A normal word resulting from collecting the word $w$ will be denoted $(w)$. It is equivalent to (represents the same element of the group as) the given word. There are various uniform methods for collecting a word. For instance, collection from the left replaces the left-most minimal non-normal subword at each step. This is the basic method of collection which is now used in implementations (see Leedham-Green and Soicher (1990) and Vaughan-Lee (1990)).

If every word in $A$ collects to a unique normal word relative to $R$, then the group $(A \mid R)$ has order $p^n$ and the presentation $(A \mid R)$ is said to be consistent.

There are usually many power-commutator presentations for a finite $p$-group. Every unrefinable normal series of a $p$-group gives rise to a power-commutator presentation. For this work we choose special unrefinable normal series and hence limit the power-commutator presentations under consideration.

Let $G$ be a group and $p$ a prime. The series

$$G = \mathcal{P}_0(G) \gneq \mathcal{P}_1(G) \gneq \ldots \gneq \mathcal{P}_b(G) \gneq \ldots,$$

where $\mathcal{P}_b(G) = [\mathcal{P}_{b-1}(G),G](\mathcal{P}_{b-1}(G))^p$, for $b \geq 1$, is the lower exponent-$p$ central series of $G$. If $G$ is a $p$-group, there exists an integer $c \geq 0$ such that $\mathcal{P}_c(G) = \langle 1 \rangle$; the smallest such integer is called the exponent-$p$ class, or simply the class, of $G$. (Sims, 1994, p. 445, uses the notation $\varphi$ instead of $\mathcal{P}$ and $G = \varphi_1(G)$.)

We consider power-commutator presentations $(A \mid R)$ for a finite $p$-group $G$ which come from a normal series refining the lower exponent-$p$ central series of $G$. If $G$ has order $p^n$ and generator number $d$ then $\{a_1,\ldots,a_d\}$ generates $G$ and $a_{d+1},\ldots,a_n$ can be expressed in terms of $a_1,\ldots,a_d$. Also the power-commutator presentation takes a special form. For this a weight function $\omega$ is defined from $A$ into the set $\{1,\ldots,c\}$ mapping $a$ to $b$ if $a \in \mathcal{P}_{b-1}(G) \setminus \mathcal{P}_b(G)$. The relations in $R$ have the form:

$$a_i^p = \prod_{a_k \in A, \omega(a_k) > \omega(a_i)} a_k^{\alpha(i,i,k)}, \text{ for } 1 \leq i \leq n,$$

$$[a_i,a_j] = \prod_{a_k \in A, \omega(a_k) \geq \omega(a_i) + \omega(a_j)} a_k^{\alpha(i,j,k)}, \text{ for } 1 \leq i < j \leq n,$$

with $0 \leq \alpha(i,j,k) < p$. A power-commutator presentation together with this weight function is a weighted power-commutator presentation. A word $w$ in $A$ has weight $b$ if the element it represents lies in $\mathcal{P}_{b-1}(G)$ but not in $\mathcal{P}_b(G)$.

Let $p^{d(b)}$ be the order of $G/\mathcal{P}_b(G)$; so $d(1) = d$. A power-commutator presentation for $G$ can then be written down as follows.

1. Initialise $A$ to be $\{a_1,\ldots,a_n\}$ and $R$ to be empty; choose $\{a_1,\ldots,a_d\}$ to be a basis for $G$ modulo $\mathcal{P}_1(G)$;
2. for $b$ in $1,\ldots,c-1$ select a basis $B_b$ for $\mathcal{P}_b(G)$ modulo $\mathcal{P}_{b+1}(G)$ from the set

\{[a_j, a_i]; a_j^\omega = b, \omega(a_i) = 1, i < j\} and add to \mathcal{R} relations whose left-hand sides run through \mathcal{B}_b and have the form \([a_j, a_i] =: a_k \) or \(a_j^\omega =: a_k \) for \(k\) in \([d(b) + 1, \ldots, d(b + 1)]\);

3. find the normal forms of the commutators and powers which do not occur as the left-hand sides of relations in step 2 and add the corresponding relations to \(\mathcal{R}\).

For a proof that the set \(\{[a_j, a_i]; a_j^\omega = b, \omega(a_i) = 1, i < j\}\) generates \(\mathcal{P}_b(G)\) modulo \(\mathcal{P}_{b+1}(G)\) see Sims (1994, p. 446, Propositions 9.11.2 and 9.11.3). The relations added in step 2 are called definitions. For each generator \(a_k\) there is a unique definition with \(a_k\) as its right-hand side, the definition of \(a_k\). For later purposes we also allow a definition of \(a_k\) to have a right-hand side \(ua_k\) where \(u\) is a (possibly empty) word in earlier generators. A weighted power-commutator presentation with definitions is called labelled; the definitions will be indicated by ‘:’ as shown.

A \(p\)-covering algorithm takes as input a consistent labelled weighted power-commutator presentation \(\langle A \mid \mathcal{R}\rangle\) and returns a consistent labelled weighted power-commutator presentation, \(\langle A^+ \mid \mathcal{R}^+\rangle\), for the \(p\)-covering group \(\langle A \mid \mathcal{R}\rangle^+\). In addition \(A^+\) contains \(A\); the elements in \(A^+ \setminus A\) are called new generators. Each relation of \(\mathcal{R}^+\) ends in a (possibly empty) normal word in the new generators; we call such words tails. The result of making the new generators trivial, is, after tidying, the original presentation. We say \(\langle A^+ \mid \mathcal{R}^+\rangle\) extends \(\langle A \mid \mathcal{R}\rangle\). A straightforward way of writing down a (not necessarily consistent) power-commutator presentation \(\langle A \mid \mathcal{R}\rangle\) for the covering group is to add generators, one for each relation which is not a definition, and modify the right-hand side of each such relation by multiplying it on the right by one of the new generators—a different generator for each relation. Relations are added to make the new generators central and of order \(p\). This whole procedure can be called ‘adding tails’; it is described more formally in the following algorithm description. The number of relations in \(\mathcal{R}\) which are not definitions is \(n(n - 1)/2 + d =: s\).

**ALGORITHM 1**

1. Initialise \(\tilde{A}\) to be \(A\);
2. initialise \(\tilde{\mathcal{R}}\) to consist of all relations of \(\mathcal{R}\) which are definitions;
3. modify each non-definition \(a_i^p = v_{ji}\) or \([a_k, a_j] = v_{jk}\) of \(\mathcal{R}\) to read \(a_i^p =: v_{ji}a_r\) or \([a_k, a_j] =: v_{jk}a_r\) for some \(r \in \{n + 1, \ldots, n + s\}\) where different non-definitions are modified by different \(a_r\) and add each \(a_r\) to \(\tilde{A}\);
4. add the modified relations to \(\tilde{\mathcal{R}}\);
5. add to \(\tilde{\mathcal{R}}\) the relations \([a_r, a_i] = 1\) for \(n + 1 \leq r \leq n + s\) and \(1 \leq i < r\);
6. add to \(\tilde{\mathcal{R}}\) the relations \(a_r^p = 1\) for \(n + 1 \leq r \leq n + s\).

In this case the tails are words of length 1 and each new generator occurs in exactly one tail. The presentation \(\langle \tilde{A} \mid \tilde{\mathcal{R}}\rangle\) resulting from **ALGORITHM 1** has two features which are important to note.

A. The definition of each generator \(a_{d+1}, \ldots, a_{n+s}\) can be rewritten as \(a_r = v_{jk}^{-1}[a_k, a_j]\) or \(a_r = v_{ji}^{-1}a_i^p\) which expresses each such generator in terms of earlier generators. Hence, by repeated substitutions, each \(a_r\) for \(d+1 \leq r \leq n+s\) can be expressed as a word \(u_r\) in the generators \(a_1, \ldots, a_d\). Thus the definition of \(a_r\) can be transformed into the relation \(u_r = u_r\).
B. By the previous remark, the relations \([a_r, a_i] = 1\) for \(n + 1 \leq r \leq n + s\) and \(d + 1 \leq i < r\) are consequences of the relations \([a_r, a_i] = 1\) for \(n + 1 \leq r \leq n + s\) and \(1 \leq i \leq d\).

Remark A shows that a presentation on the generators \(a_1, \ldots, a_d\) can be obtained from \(\{\hat{A} \mid \hat{R}\}\) by a sequence of Tietze transformations. In the resulting presentation the relations are in one-to-one correspondence with the relations introduced in steps 5 and 6 of Algorithm 1. By Remark B this set of relations can be further reduced to the set of relations \([u_r, a_i] = 1\) and \(w^r_i = 1\) for \(n + 1 \leq r \leq n + s\) and \(1 \leq i \leq d\).

Remark A describes a method for expressing each generator \(a_r\) for \(d + 1 \leq r \leq n + s\) in terms of \(a_1, \ldots, a_d\). This method can also be used to modify the presentation \(\{A \mid R\}\). Applying it to the relations in \(R\) one obtains the relations \(u_r = a_r\) for \(d + 1 \leq r \leq n\) from definitions and \(u_r = 1\) for \(n + 1 \leq r \leq n + s\) from non-definitions. Eliminating \(a_{d+1}, \ldots, a_n\) from \(A\) yields a presentation on the generators \(a_1, \ldots, a_d\) with the relations \(u_r = 1\) for \(n + 1 \leq r \leq n + s\). This proves the following theorem.

**Theorem 2.1.** The presentation \(\{\hat{A} \mid \hat{R}\}\) is a power-commutator presentation for the covering group \(\langle A \mid R \rangle^\ast\).

Algorithm 1 returns a labelled power-commutator presentation which is not necessarily consistent. A simple example for this occurs in the case of a cyclic group of order 4 given by the power-commutator presentation \(\{a_1, a_2 \mid [a_2, a_1] = 1, a_1^2 = a_2, a_2^2 = 1\}\). Algorithm 1 introduces new generators \(a_3\) and \(a_4\) and returns the following presentation

\[
\begin{align*}
\{ & a_1, a_2, a_3, a_4 \mid \quad a_1^2 = a_2, \\
& [a_2, a_1] = a_3, \quad a_2^2 = a_4, \\
& [a_3, a_1] = 1, \quad [a_3, a_2] = 1, \quad a_3^2 = 1, \\
& [a_4, a_1] = 1, \quad [a_4, a_2] = 1, \quad [a_4, a_3] = 1, \quad a_4^2 = 1 \}
\end{align*}
\]

for the covering group. The word \(a_1^2\) collects to more than one normal word, since \((a_1^2)a_1 = a_2a_1 = a_3a_2a_3\) and \(a_1(a_1^2) = a_4a_2\). Hence \(a_2 = 1\). Using this it is easy to obtain a consistent presentation for the covering group. However, in general, changing to a consistent presentation requires significant work. A method for obtaining a consistent presentation is given in Section 3.

In practice the covering algorithm is split into two parts:

(a) a **tails routine** which takes as input a consistent labelled weighted power-commutator presentation \(\{A \mid R\}\) and returns a labelled power-commutator presentation for \(\langle A \mid R \rangle^\ast\) which extends \(\{A \mid R\}\);

(b) a **consistency routine** which takes the output of the tails routine and returns a consistent labelled weighted power-commutator presentation for \(\langle A \mid R \rangle^\ast\) which extends \(\{A \mid R\}\).

Algorithm 1 is an example of a tails routine. It is quite impractical because, except for small examples, it defines far too many new generators. In Section 4 we discuss more practical tails routines and give a proof for the correctness of the main algorithm.

After a tails routine has been completed, the consistency relations in the next section are used to create a system of homogeneous linear equations in a set of unknowns which
correspond to the new generators. The resulting system of equations is solved by elimination. The fewer unknowns the system has the less space the solution process is likely to take. In practice, this can be, and has been, critical.

3. A Consistency Routine

Let \{A | R\} be a consistent labelled weighted power-commutator presentation and let \{\bar{A} | \bar{R}\} be a power-commutator presentation computed by a tails routine for the group \(\langle A | R \rangle^*\). So the elements of \(\bar{A} \setminus A\) are new generators and each relation in \(\bar{R}\) differs from a relation in \(R\) only by a tail.

The consistency test of Wamsley (1974), as sharpened by Vaughan-Lee (1984) and described in Sims (1994, p. 447), can be used to change the presentation \{\bar{A} | \bar{R}\} to a consistent presentation for the same group. The results of the following collections relative to \{\bar{A} | \bar{R}\} form a set \(S\) of words in the new generators because \{\bar{A} | \bar{R}\} extends the consistent presentation \{A | R\} (recall that \((w)\) means a normal word collected from \(w\)):

\[
\begin{align*}
(a_k (a_j a_i))^{-1} & ((a_k a_j) a_i) & \text{for } 1 \leq i < j < k \leq n, i \leq d, \omega(a_i) + \omega(a_j) + \omega(a_k) \leq c, \\
(a_p^i) a_j^{-1} & (a_j^{p-1} (a_j a_i)) & \text{for } 1 \leq i < j \leq n, i \leq d, \omega(a_i) + \omega(a_j) < c, \\
(a_k (a_j a_i))^{-1} & (a_k a_j a_i^{p-1}) & \text{for } 1 \leq j < k \leq n, \omega(a_j) + \omega(a_k) < c, \\
(a_i a_k^{p-1})^{-1} & (a_k a_i) & \text{for } 1 \leq i \leq n, 2\omega(a_i) < c.
\end{align*}
\]

This gives a set
\[
\{v = 1 \mid v \in S\} \quad (3.1)
\]
of relations which are consequences of \(\bar{R}\). This set of relations can be ‘echelonised’. Thus there is a set \(G\) of new generators such that the set of echelonised relations is equivalent to the set
\[
\{t = v_t(G) \mid t \in \bar{A} \setminus (A \cup G)\}. \quad (3.2)
\]

Add these relations to \(\bar{R}\). These additional relations can be used to eliminate the new generators not in \(G\) using Tietze transformations. This yields a finite presentation \(\{A^* | \mathcal{X}\}\), where \(A^* = A \cup G\), for the covering group. Moreover \(\mathcal{X}\) contains a sub-set \(\mathcal{R}^*\) such that \(\{A^* | \mathcal{R}^*\}\) is a power-commutator presentation. The other relations in \(\mathcal{X}\), namely those involving \(v_t(G)\) on the left-hand side, are consequences of the relations in \(\mathcal{R}^*\), so \(\{A^* | \mathcal{R}^*\}\) is a power-commutator presentation for \(\langle A | R \rangle^*\). The result of Vaughan-Lee (1984) gives that the presentation is consistent. The ordering on the generating set \(A^*\) can be changed to give a weighted power-commutator presentation.

As the class and size of the presentation grow, the size of the set \(S\) becomes much larger than the number of generators that need to be eliminated. The example below illustrates this. It is an open problem to find an \textit{a priori} way of reducing the set \(S\) while still ensuring consistency of the resulting presentation.

4. Tails Routines

In this section we discuss more practical tails routines than \textsc{Algorithm 1}. A first observation, which was made by Macdonald (1974) and by Wamsley (1974), is that for
$a_i, a_j \in A$ with $\omega(a_i) + \omega(a_j) \geq c+2$ the elements $[a_j, a_i]$ are the identity in the covering group. So there is no need to add new generators corresponding to these relations.

Let $q$ be the number of non-definition relations in $R$ whose left-hand side is either a $p$th power or a commutator $[a_j, a_i]$ with $\omega(a_i) + \omega(a_j) \leq c+1$. Consider the presentation computed by the following algorithm. The input to the algorithm is a consistent labelled weighted power-commutator presentation, $\langle A \mid R \rangle$, and its exponent-$p$ class $c$.

**Algorithm 2**

1. Initialise $\tilde{A}$ to be $A$;
2. initialise $\tilde{R}$ to consist of all relations of $R$ which are definitions;
3. add to $\tilde{R}$ all relations of $R$ with left-hand side $[a_j, a_i]$ and $\omega(a_j) + \omega(a_j) > c + 1$;
4. modify each non-definition $a_i^p = v_{ij}$ or $[a_k, a_j] = v_{jk}$ of $R$ with $\omega(a_k) + \omega(a_j) \leq c + 1$ to read $a_i^p = v_{ij} a_r$ or $[a_k, a_j] = v_{jk} a_r$ for some $r \in \{n + 1, \ldots, n + q\}$ where different non-definitions are modified by different $a_r$ and add each $a_r$ to $\tilde{A}$;
5. add the modified relations to $\tilde{R}$;
6. add to $\tilde{R}$ the relations $[a_r, a_i] = 1$ for $n + 1 \leq r \leq n + q$ and $1 \leq i < r$;
7. add to $\tilde{R}$ the relations $a_r^p = 1$ for $n + 1 \leq r \leq n + q$.

In this case the tails are words of length 0 or 1 and each new generator occurs in exactly one tail.

**Theorem 4.1.** The presentation $\langle \tilde{A} \mid \tilde{R} \rangle$ is a power-commutator presentation for the covering group $\langle A \mid R \rangle^\ast$.

**Proof.** Sims (1994, Proposition 9.11.1) shows that for $a_i, a_j \in A$ with $\omega(a_i) + \omega(a_j) > c + 1$ the relations $[a_j, a_i] = 1$ hold. The result follows from Theorem 2.1. □

**Example**

Consider as an example an intermediate stage in the computation of a power-commutator presentation for $B(4,4)$, the free group of exponent 4 on four generators. The largest 2-quotient of class 7, $B(4,4;7)$, of $B(4,4)$ has order $2^{352}$. It has a power-commutator presentation $\langle A \mid R \rangle$ with 4, 10, 20, 55, 99, 84 and 80 generators of weights 1, ... 7, respectively. Then Algorithm 2 computes a covering presentation with 8782 new generators. The covering group of $B(4,4;7)$ has order $2^{1208}$ and so can be defined by 856 additional generators. In this situation Algorithm 1 would define 61780 new generators. Thus, while Algorithm 2 is good progress, there is scope for further improvement. Algorithm 3 described below introduces only 1058 new generators to define the covering presentation. There are 11453 words in the list for enforcing consistency which give rise to 202 relations. ◊

As mentioned in Section 2, the set

$S_b = \{[a_j, a_i], a_i^p \mid \omega(a_j) = b, \omega(a_i) = 1, i < j\}$

is a spanning set for $\mathcal{P}_b(G)$ modulo $\mathcal{P}_{b+1}(G)$. This shows that the number of new generators added in Algorithm 2 is too large. We describe a method which adds new generators only to non-definitions which have left-hand sides in the union of the $S_b$.

Let $E$ be the set of new generators in $\tilde{A}$ whose definition is a relation with left-hand side either a $p$th power or a commutator $[a_j, a_i]$ with $\omega(a_i) = 1$. The definitions of the
generators in $A \cup E$ involve precisely all the $p$th powers and all the commutators $[a_j, a_i]$ with $\omega(a_i) = 1$. Of these $n - d$ are definitions of generators in $A$, so $E$ has $nd - d(d-1)/2$ elements.

By the above each new generator not in $E$ can be written as a word in $E$ using the relations in $\hat{R}$. In the following we will show how a series of calculations relative to $\{\hat{A} | \hat{R}\}$ computes for each new generator $a_r$ added in Algorithm 2 an explicit word in $E$ equivalent to it. These relations can be used to eliminate all those $a_r$ not in $E$ from $\{\hat{A} | \hat{R}\}$ by a simple sequence of Tietze transformations. Algorithm 3 below is based on this approach. Thus the tails in the presentation obtained from Algorithm 3 will be normal words in $E$.

The calculations can be so organised that it suffices to introduce new generators from $E$ and then carry out certain collections in an appropriate order from which the tails can be computed as normal words in $E$. This means that these collections are computed in presentations which, in general, are not covering presentations. This is discussed further after Theorem 4.6. The procedure outlined reflects the current implementation of the tails routine in the ANU $p$-Quotient Program (Havas et al., 1995).

The way in which the covering algorithm is usually used means that the input presentation has definitions of the form $[a_j, a_i] = a_k$ or $a_p = a_j$. We call such presentations simply labelled. For ease of exposition we restrict attention to them.

The proof of the next lemma illustrates the use of collection as a first step in evaluating tails.

**Lemma 4.2.** Let $F$ be the free group on $\{x, y, z\}$. Then

$$[z, [y, x]] [z, x, y] = [z, y, x] \mod P_3(F)$$

and

$$[z, y^p] = [z, y]^p \mod P_3(F).$$

**Proof.** We prove the first equation by collecting the word $zyx$ in two different ways assuming $x < y < [y, x] < z$:

$$zyx = yz[z, y]x = xy[y, x]z[z, x][z, y][z, y, x]$$

and

$$zyx = xz[z, x][y, x] = xy[y, x]z[y, x][z, y][z, x][z, y, x] \mod P_3(F)$$

$$= xy[y, x]z[z, x][z, y][z, y, x][z, y, x] \mod P_3(F).$$

Comparing the results gives the first formula. The second formula is obtained similarly by collecting the word $zy^p$ in two different ways. □

A version of this proof can be used to calculate formulas modulo higher weights, e.g.

$$[z, [y, x]] = [z, y, x][[z, x, y][z, x, [y, x]][z, y, [y, x]]][z, y, [z, x]]^{-1} \mod P_4(F).$$

The commutators of weight 4 constitute a correction term to the formula modulo $P_3(F)$.

If the collections in the proof are carried out modulo $P_4(F)$, and the weight-3 formula is substituted for $[z, [y, x]]$, the results of the two collections differ precisely in the correction term above. Similarly, working modulo weight $c + 1$ and given a formula for $[z, [y, x]]$
The function \( \hat{a}(\text{relation}) \), particularly we write \( [a, b] \), where \( a \in \hat{b} \) is a correction term for the weight-$$a$$ by collecting \( z, y \in \hat{b} \), then \( [a, b] \), \( a \in \hat{b} \) and its left-hand side \( [a, b] \) is the corresponding relation in \( \langle A \mid R \rangle \). It can be computed by collecting \( a_k a_j a_i \) in two different ways and substituting the weight-

\[
[a_k, [a_j, a_i]] = [a_k, a_j, a_i][a_k, a_i, a_j]^{-1} \mod \mathcal{P}_b(G)
\]

and

\[
[a_k, a_h^p] = [a_k, a_h]^p \mod \mathcal{P}_b(G).
\]

As before, formulas for \( [a_k, [a_j, a_i]] \) modulo higher weights can be obtained.

Now consider the presentation \( \langle A \mid R \rangle \). The left-hand side of the relation \( [a_k, a_h] = v_{hk} \) is not in the union of the \( S_b \), \( 1 \leq b \leq c \). This means that \( a_h \) has a definition \( [a_j, a_i] = a_h \). Substituting this for \( a_h \), one gets the commutator \( [a_k, [a_j, a_i]] \) and the relation \( [a_k, a_h] = v_{kh} \) can be viewed as a formula for \( [a_k, [a_j, a_i]] \) modulo weight \( c \). In this context a tail is a correction term for the weight-(\( c+1 \)) formula. It can be computed by collecting \( a_k a_j a_i \) in two different ways with respect to \( \langle A \mid \hat{R} \rangle \). Note that these collections arise from specific consistency words.

Define the function \( \hat{\omega} \) from \( \hat{A} \) into the set \( \{1, \ldots, c+1\} \) by

\[
\hat{\omega}(a_k) = \begin{cases} 
\omega(a_k) & \text{for } 1 \leq k \leq n, \\
\omega(a_j) + 1 & \text{if } k > n \text{ and the definition of } a_h \text{ has left-hand side } a_h^p, \\
\omega(a_i) & \text{if } k > n \text{ and the definition of } a_h \text{ has left-hand side } [a_j, a_i].
\end{cases}
\]

The function \( \hat{\omega} \) need not be a weight function because \( a \) may not lie in the term of the lower exponent-\( p \) central series of \( \langle A \mid \hat{R} \rangle^* \) specified by \( \hat{\omega}(a) \); we refer to \( \hat{\omega}(a) \) as the pseudo-weight of \( a \). A word \( w \) in \( \hat{A} \) has pseudo-weight \( r \) if \( r \) is the smallest pseudo-weight of a generator occurring in any equivalent normal word for \( w \). A word in \( \hat{A} \) is \( r \)-heavy if it has pseudo-weight at least \( r \). Let \( \hat{E}_b \) denote the elements in \( \hat{E} \) of pseudo-weight at least \( b \).

Throughout the rest of this section we will use that the new generators introduced in Algorithm 2 are central and of order \( p \) in \( \langle A \mid \hat{R} \rangle^* \) without explicit mention. For \( y, z \in A \) we will write the relation with left-hand side \( [z, y] \) in the form \( [z, y] = v[z, y]t[z, y] \) where \( [z, y] = v[z, y] \) is the corresponding relation in \( \langle A \mid R \rangle \). Thus \( t[z, y] \) is a tail. Similarly we write \( y^p := v[y^p]t[y^p] \).

A non-definition \( [z, u] = v[z, u] \) in \( \langle A \mid \hat{R} \rangle \) with \( \omega(u) > 1 \) will be called a permanent non-definition, because the corresponding relation remains a non-definition in the presentation for \( \langle A \mid \hat{R} \rangle^* \) obtained by Algorithm 3. However, they are replaced by definitions in Algorithm 2. We say that \( \omega(z) + \omega(u) \) is the weight of the non-definition. If the definition of \( u \) has left-hand side \( [y, x] \), then \( (z(yx))^{-1}(zyx) \) is called the associated consistency word of the non-definition; if the definition of \( u \) has left-hand side \( y^p \), then \( (z(y^p))^{-1}(zy^p) \) is the associated consistency word. For a non-definition \( [z, u] = v[z, u] \) with weight \( b \), a normal word \( e[z, u] \) in \( \hat{E}_b \) is called an associated normal word if \( [z, u] \) and \( v[z, u]e[z, u] \) are equivalent in \( \langle A \mid \hat{R} \rangle^* \). Note that a non-definition can have more than one associated normal word in \( \hat{E} \).
Linearily order all the non-definitions inversely by weight, then those of equal weight directly by the weight of the second generator on the left-hand side and then arbitrarily.

**Lemma 4.4.** Evaluating the associated consistency words in an appropriate order yields an associated normal word in \( \mathcal{E}_b \) for each permanent non-definition of weight \( b \).

**Proof.** The proof is by induction on the above order. Let \( [z,u] = v_{[z,u]} \) be a permanent non-definition. We denote it \( r(z,u) \). Let \( b \geq 4 \) be its weight. Assume that for all earlier non-definitions an associated normal word in \( \mathcal{E}_b \) is known. Let \( m = \hat{\omega}(u) \). Note that \( 2m \leq b \). The rest of the proof depends on the nature of the definition of \( u \).

**Case 1:** \( [y,x] =: u \)

<table>
<thead>
<tr>
<th>( t_{[z,u]} )</th>
<th>( b )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z )</td>
<td>( b - m )</td>
</tr>
<tr>
<td>( y )</td>
<td>( m - 1 )</td>
</tr>
<tr>
<td>( x )</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 1.** \( \hat{\omega} \).

If the first step is interchanging \( z \) and \( y \), one gets a sequence of collection steps of which the important steps are shown in Figure 2. The collection steps shown have to occur at some stage of the collection in the indicated order. The words \( v_{[z,y]} \) and \( t_{[z,y]} \) are \((b-1)\)-heavy.

In the course of the collection there will be a (unique) step which replaces the subword \( zx \) by \( xze_{[z,x]}t_{[z,x]} \). That step will be applied to a word of the form \( yzxw_1 \) where \( w_1 \) is a not necessarily normal word which is \((b-1)\)-heavy since it was obtained by applying collection steps to the word \( v_{[z,y]}t_{[z,y]}x \). The result of the step is \( yzxw_1 \). The collection continues until the next step replaces the subword \( zx \) in a word of the form \( yzxw_2 \) to give \( xyuw_2 \). The collection finally yields \( xyuw_3t_3 \) where \( w_3 \) is a normal word in \( \mathcal{A} \) and \( t_3 \) is a normal word in the new generators not containing \( t_{[z,u]} \).

The important collection steps that occur if the first step is interchanging \( y \) and \( x \) are shown in Figure 3. In the course of the collection there will be a next step which replaces the subword \( zy \) by \( yzw_{[z,y]}t_{[z,y]} \). That step will be applied to a word of the form \( xzyw_4 \), where \( w_4 \) contains \( u \). The result of the step is \( xzyw_{[z,y]}t_{[z,y]}w_4 \). Collection continues until it replaces the subword \( zu \). That step will be applied to a word of the form \( xzywu_5 \) and gives \( xyuw_{[z,u]}t_{[z,u]}w_5 \). The collection finally yields \( xyuw_6t_6 \) where \( w_6 \) is a normal word in \( \mathcal{A} \) and \( t_6 \) is a normal word in the new generators containing \( t_{[z,u]} \) once.

Since \( ((zy)x) \) and \( (z(yx)) \) represent the same element of the covering group and the consistency of \( \{ \mathcal{A} \mid \mathcal{R} \} \) implies that \( w_3 = w_6 \), the relation \( t_3 = t_6 \) is a consequence of the relations in \( \mathcal{R} \).
Consider the steps of the first collection as far as words in the new generators are concerned. The word \( w_1 \) contains \( t_{[z,y]} \), which is not in \( \mathcal{E}_b \), once. All the other new generators in \( w_1 \) arise in one of two ways. One possibility is that they come from steps which collect \( x \) past a generator in \( v_{[z,y]} \). Since \( v_{[z,y]} \) is \((b-1)\)-heavy and \( x \) has pseudo-weight 1, such a collection introduces a new generator from \( \mathcal{E}_b \). The other possibility is that they “come from earlier permanent non-definitions” in the sense that they are introduced by collection steps which use relations in \( \mathcal{R} \) corresponding to permanent non-definitions in \( \mathcal{R} \) which are earlier in the given order than \( r(z,u) \). These collection steps have associated normal words in \( \mathcal{E}_b \) by the induction hypothesis. The words \( w_2 \) and \( t_3 \) contain \( t_{[z,y]} \) and \( t_{[z,x]} \) once each. Since \( w_1 \) is \((b-1)\)-heavy and \( t_{[z,y]} \) is \((b-m+1)\)-heavy (and \( b-m+1 \geq 2 \)) the word \( t_3 \) is equivalent to \( t_{[z,y]}t_{[z,x]}t_3^* \), where \( t_3^* \) is a normal word in \( \mathcal{E}_b \).

Consider the steps of the second collection. The word \( w_4 \) contains \( t_{[z,y]} \) and \( u \) once each and otherwise is \((b-m+1)\)-heavy. All the new generators in \( w_4 \) either are in \( \mathcal{E}_b \) or come from earlier permanent non-definitions. The word \( w_5 \) contains \( t_{[z,y]} \) and \( t_{[z,x]} \) once each and otherwise is \((b-m+1)\)-heavy. All the new generators introduced in the steps from \( xyzv_{[z,y]}t_{[z,y]}w_4 \) to \( xyzuw_5 \) come from earlier permanent non-definitions. Finally the word \( t_6 \) contains \( t_{[z,u]} \), \( t_{[z,y]} \) and \( t_{[z,x]} \) once each and hence, since \( v_{[z,u]} \) is \((b-1)\)-heavy, \( t_6 \) is equivalent to \( t_{[z,u]}t_{[z,y]}t_{[z,x]}t_6^* \), where \( t_6^* \) is a normal word in \( \mathcal{E}_b \). Hence the normal word associated with \( r(z,u) \) is \( (t_3^*t_6^*)^{-1} \).

**CASE 2:** \( y^p =: u \)

Then \( \omega(y) = m - 1 \). We compute an associated normal word in \( \mathcal{E}_b \) for \( r(z,u) \) by collecting the word \( zy^p \) in two ways which begin with different steps and comparing the resulting normal words \( (z(y^p)) \) and \( ((zy)y^{p-1}) \). As in Case 1 we introduce words \( w_1, \ldots, w_p \) and \( t_p \) in the course of describing the two collections and then retrace the steps to show they give the desired conclusion. If the first step is the application of the power relation, then one obtains the collection shown in Figure 4.

Now we consider the collection when the first step is interchanging \( z \) and \( y \) as shown in Figure 5. In the course of the collection there will be a next step which replaces the subword \( zy \) by \( yzv_{[z,y]}t_{[z,y]} \). That step will be applied to a word of the form \( yzwy_1 \) where \( w_1 \) contains exactly \( p-2 \) occurrences of the generator \( y \) and one occurrence of \( t_{[z,y]} \). The result of the step is \( y^2zv_{[z,y]}t_{[z,y]}w_1 \). The collection continues until all generators \( y \) have been interchanged with \( z \). In each case the subword \( zy \) is replaced in a word of the form \( y^izwy_1 \), where \( w_1 \) contains exactly \( p-i-1 \) occurrences of \( y \) and \( i \) occurrences of \( t_{[z,y]} \). After the last replacement of the subword \( zy \) the result is \( y^nzw_{[z,y]}t_{[z,y]}w_{p-1} \). The collection finally yields \( uzwtp \), where \( w_p \) is a normal word in \( \mathcal{A} \) and \( t_p \) is a normal word in the new generators with no occurrence of \( t_{[z,y]} \).

Since \( (z(y^p)) \) and \( ((zy)y^{p-1}) \) represent the same element of the covering group \( \langle \mathcal{A} | \mathcal{R} \rangle^* \) and \( \{ \mathcal{A} | \mathcal{R} \} \) is consistent, \( t_{[z,u]} = t_p \) is a consequence of the relations in \( \mathcal{R} \).
Consider the steps of the second collection as far as words in the new generators are concerned. The word \( w_1 \) contains \( t_{[z,y]} \), which is not in \( \mathcal{E}_b \), once. All the other new generators in \( w_1 \) arise in one of two ways. One possibility is that they come from steps which collect \( y \) past a generator in \( v_{[z,y]} \). Since \( v_{[z,y]} \) is \((b - 1)\)-heavy and \( y \) has pseudo-weight \( m - 1 \), such a collection introduces a new generator from \( \mathcal{E}_b \). The other possibility is that they come from earlier permanent non-definitions. These collection steps have associated normal words in \( \mathcal{E}_b \) by the induction hypothesis. The word \( w_i \) for \( i < p \) contains the new generator \( t_{[z,y]} \), which is not in \( \mathcal{E}_b \), exactly \( i \) times and all other new generators come from earlier non-definitions. Finally the word \( w_p \) is a normal word in \( \mathcal{A} \). The word \( t_p \) has no occurrences of the new generator \( t_{[z,y]} \) as new generators have order \( p \), and all other new generators come from earlier non-definitions. \( \square \)

We reduce \( \mathcal{E} \) a little further. In the following we show how the new generators in \( \mathcal{E} \) can be expressed in terms of a subset of \( \mathcal{E} \). Let \( \mathcal{F} \) be the set of those elements \( t \) in \( \mathcal{E} \) for which \( t \) is defined as a commutator or as the \( p \)th power of a generator \( y \), where \( y \) itself is defined as a \( p \)th power. Let \( \mathcal{F}_b \) denote the elements in \( \mathcal{F} \) of pseudo-weight at least \( b \). The size of \( \mathcal{F} \) is \( n(d - 1) + m - d(d - 1)/2 \) where \( p^m \) is the order of the commutator quotient of \( \langle A \mid R \rangle \).

We extend the definition of permanent non-definitions to include the further relations that remain non-definitions in ALGORITHM 3. Again, these relations are replaced by definitions in ALGORITHM 2. A non-definition \( w = v_{u^p} \) in \( \{ A \mid R \} \) where \( u \) is defined by a commutator relation is also called a permanent non-definition and \( \omega(u) + 1 \) is its weight. If the definition of \( u \) has left-hand side \([z,y]\) then \((z^p)y^{-1}(z^{p-1}zy)\) is called the associated consistency word of the non-definition. For a permanent non-definition of the form \( w = v_{u^p} \) with weight \( b \), a normal word \( f_{u^p} \) in \( \mathcal{F}_b \) is called an associated normal word if \( u^p \) and \( v_{u^p} f_{u^p} \) are equivalent in \( \langle A \mid R \rangle^* \).

Linearly order this larger set of permanent non-definitions to extend the previous order. Again, order the non-definitions inversely by weight, then for those of equal weight according to their left-hand sides, where those with left-hand side a commutator come after those with left-hand side a \( p \)th power. The non-definitions of equal weight with left-hand side a commutator are ordered as before; those with left-hand side a \( p \)th power are ordered arbitrarily.

**LEMMA 4.5.** Evaluating the associated consistency words in an appropriate order yields an associated normal word in \( \mathcal{F}_b \) for each permanent non-definition of weight \( b \).

**PROOF.** The proof is by induction on the linear order of the non-definitions.

For a permanent non-definition \([z,u] = v_{[z,u]} \) the proof is essentially the same as in Lemma 4.4. It remains to consider a permanent non-definition \( u^p = v_{u^p} \) of weight \( b \). We
denote it by \( r(u^p) \). Let \( u \) be defined by \([z, y] := u\). Assume that for all earlier permanent non-definitions an associated normal word in \( F_b \) is known.

Since \( u \in A \) we know that \( \hat{\omega}(y) = 1 \), thus \( \hat{\omega}(z) = b - 2 \), since \( \hat{\omega}(u) = b - 1 \). Note this means \( b \geq 3 \). The pseudo-weights are displayed in Figure 6. We compute an associated word in \( F_b \) for \( r(u^p) \) by collecting the word \( z^p y \) in two different ways which begin with different steps and comparing the resulting normal words ((\( z^p y \)) and (\( z^{p-1} (z y) \)) which represent the same element of \( \langle A | R \rangle^* \). In a similar way to the above we introduce words \( w_1, \ldots, w_{p+2}, t_1 \) and \( t_2 \) in the course of describing the two collections and then retrace the steps to show they give the desired result.

If the first step is applying the power relation then \( z^p y \rightarrow v_z t_z y \). The collection continues and finally yields \( y w_1 t_1 \). Here \( w_1 \) is a normal word in \( A \) and \( t_1 \) is a normal word in the new generators, see Figure 7.

If the first step is interchanging \( z \) and \( y \), the next step again interchanges \( z \) and \( y \) (see Figure 8). In the course of the collection there will be a next step which replaces the subword \( z y \) by \( y z u \). That step will be applied to a word of the form \( z^{p-2} y w_2 \) where \( w_2 \) is a not necessarily normal word that contains exactly two occurrences of \( z \) and two occurrences of \( u \). The result of the step is \( z^{p-3} y zu w_2 \).

The collection continues until all generators \( z \) have been interchanged with \( y \). In each case the subword \( z y \) is replaced in a word of the form \( z^{i-p} y w_1 \), where \( w_1 \) contains exactly \( i \) occurrences of \( z \) and \( i \) occurrences of \( u \) and \( i < p \). The collection continues until the next step replaces the subword \( w^p \) by \( v_{w^{p+1}} t_{w^p} \). It is applied to a word of the form \( y w_p t_{w^p} w_{p+1} \). The result is \( y w_{p+1} t_{w^p} w_{p+1} \). Finally the collection yields \( y w_{p+2} t_{w^p} w_{p+1} \), where \( w_{p+2} \) is a normal word in \( A \) and \( t_2 \) is a normal word in the new generators. Note that \( t_{w^p} \) occurs exactly once in \( t_2 \).

Since \( ((z^p) y) \) and \( (z^{p-1} (z y)) \) represent the same element of the covering group and the consistency of \( \{ A | R \} \) implies \( w_1 = w_{p+2}, \) the relation \( t_1 = t_2 \) is a consequence of the relations in \( R \). So it suffices to prove that \( t_1 \) and \( (t_{w^p} t_{z^p}) \) are equivalent to normal words in \( F_b \).

Consider the steps of the first collection as far as words in the new generators are concerned. The new generators in \( w_1 \) arise either by interchanging two generators of pseudo-weight at least \( b - 1 \) and come from earlier permanent non-definitions or by interchanging \( y \) with a generator of pseudo-weight at least \( b - 1 \) and thus also come from earlier permanent non-definitions.

Consider the steps of the second collection. All new generators occurring in the word \( w_2 \) come from applying steps to the \((b - 2)\)-heavy word \( zuzu \). Thus all the new generators that arise come from earlier permanent non-definitions. Similarly all new generators in \( w_i \) for \( i \leq p + 2 \) arise from earlier permanent non-definitions, since they occur by applying...
collection steps to a \((b-2)\)-heavy word in which the only element of pseudo-weight \(b-2\) is \(z\). Thus \(t_1\) and \((t_2t_3^{-1})\) are equivalent to normal words in \(F_b\).

We obtain a power-commutator presentation \(\{\bar{A} \mid \bar{R}\}\), where \(\bar{A} = A \cup F\), for the covering group \(\langle A \mid R \rangle^*\) in the following way. Let \(v_t(F)\) denote a normal word in \(F\) associated with the permanent non-definition in \(R\) which was replaced in \(\hat{R}\) by the definition of \(t\). The lemmas give that \(\{\bar{A} \mid \bar{R}, t = v_t(F)\}\) is also a presentation for the covering group. Using Tietze transformations we can use the additional relations to eliminate the new generators not in \(F\) and obtain a finite presentation \(\{\bar{A} \mid X\}\). Then \(X\) contains a subset \(\bar{R}\) such that \(\{\bar{A} \mid \bar{R}\}\) is a power-commutator presentation. The other relations in \(X\), namely those with \(v_t(F)\) on the left-hand side, are consequences of the relations in \(\bar{R}\), so \(\{\bar{A} \mid \bar{R}\}\) is a power-commutator presentation for \(\langle A \mid R \rangle^*\).

**Theorem 4.6.** The presentation \(\{\bar{A} \mid \bar{R}\}\) is a power-commutator presentation for the covering group \(\langle A \mid R \rangle^*\).

The above proofs show how each of the new generators in \(\bar{A} \setminus F\) can be expressed as a word in \(F\). Moreover, the calculations have been so ordered that the expressions can be obtained one at a time. This means that to get \(\{\bar{A} \mid \bar{R}\}\) it suffices to add the new generators in \(F\) to \(A\) and their definitions to \(R\) and then compute the other tails in the appropriate order. Note these computations are done in presentations which are not necessarily covering presentations and may define different groups. The proofs above, suitably modified, justify this claim.

Observe that in computing the tail for \([a_j, a_i] = v_{[a_j,a_i]}\) with \(\omega(a_i) + \omega(a_j) = b\) or \(a_j^b\) with \(\omega(a_j) = b-1\) the new generators of weight less than \(b\) play no tangible role. Thus another practical consequence of the lemmas is that one can compute the tails for these relations before inserting the new generators in \(F\) of pseudo-weight less than \(b\). Again the proofs, suitably modified, justify this procedure. Both procedures are incorporated into Algorithm 3 which is described in Figure 9.

**Corollary 4.7.** Let \(G\) be a \(d\)-generator \(p\)-group. The order of the covering group of \(G\) is at most \(|G|^{d+1}\).

The computed presentation \(\{\bar{A} \mid \bar{R}\}\) is not necessarily consistent. The previous section discusses how to obtain a consistent power-commutator presentation for \(\langle A \mid R \rangle^*\) from it.

Note that the consistent covering presentation obtained this way is not necessarily simply labelled. In the context of computing a consistent power-commutator presentation for a finitely presented group the relations of the given finite presentation guarantee this.

**Example**

We illustrate Algorithm 3 on the following example. (For brevity we occasionally list only subscripts of generators and omit trivial right-hand sides.) Consider the group \(B(2,4)\), the free group on 2 generators of exponent 4. Its largest 2-quotient of class 4 has order \(2^{10}\) and is given by the following consistent labelled weighted presentation:
Input \( \{ A | R \} \), a consistent labelled weighted power-commutator presentation, and its exponent-\( p \) class \( c \).

initialise \( \bar{A} \) to be the set \( A \);
initialise \( \bar{R} \) to be the subset of definitions of \( R \);
add to \( R \) all relations of \( R \) with left-hand side \([a_j, a_i]\) with \( \omega(a_i) + \omega(a_j) > c + 1 \);
for \( b \) in \( \{ c + 1, \ldots, 2 \} \) do
append the new generators of weight \( b \) in \( F \) to \( \bar{A} \);
add to \( \bar{R} \) the definitions of those new generators in \( F \) which have pseudo-weight \( b \);
** Compute tails for \( p^h \) powers \( \omega \) with \( \omega(u) = b - 1 \) and \( u \)
** is defined as a commutator with definition \([z, y] := u \)
for each \( u \) in \( A \) of weight \( b - 1 \) do
if \([z, y] := u \) then
\( t := ((z * (y * x))^{-1} * ((z * p) * x)) \);
add to \( \bar{R} \) the relation \([z, u] = v_{z, u}^t \);
fi;
od;
m := 1;
** Compute tails for commutators \([z, u]\) where \( \omega(z) = b - m \)
** and \( \omega(u) = m + 1 \)
while \( b - m >= m + 1 \) do
for each \( u \) in \( A \) of weight \( m + 1 \) do
if \([y, x] := u \) then
** Case 1: the second generator \( u \) is defined as commutator
for each \( z \) of weight \( b - m \) with \( z > u \) do
\( t := ((z * (y * x))^{-1} * ((z * p) * x)) \);
add to \( \bar{R} \) the relation \([z, u] = v_{z, u}^t \);
fi;
od;
elseif \( y^p := u \) then
** Case 2: the second generator \( u \) is defined as \( p^h \) power
for each \( z \) of weight \( b - m \) with \( z > u \) do
\( t := ((z * (y^p))^m * ((z * p) * x)) \);
add to \( \bar{R} \) the relation \([z, u] = v_{z, u}^t \);
fi;
od;
m := m + 1;
od;
add to \( \bar{R} \) the relations \([a_i, a_j] = 1 \) for \( a_i \in F \) and \( 1 \leq i < r \);
add to \( \bar{R} \) the relations \( a_i^r = 1 \) for \( a_r \in F \).

Figure 9. Algorithm 3.

\( A = \{ a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10} \} \) and \( R \) is the set of relations containing
\[
1^2 := 4,
2^3 := 5,
3^2 := 7,
4^3 := 8910,
5^2 := 8910,
[2, 1] := 3,
[3, 1] := 6,
[4, 2] := 68910,
[5, 2] := 68910,
[5, 3] := 8910,
[6, 2] := 6,
[6, 3] := 64,
[7, 2] := 70,
[7, 3] := 70,
[8, 2] := 9,
[9, 2] := 9,
[10, 2] := 70,
[10, 3] := 70,
\]
and the relations
\[
\begin{align*}
& a_8^2 = 1, \quad a_9^2 = 1, \quad a_{10}^2 = 1 \\
& [a_8, x] = 1, \quad [a_9, x] = 1, \quad [a_{10}, x] = 1 \quad \text{for the relevant } x \in A.
\end{align*}
\]
The generators of weight 1 are $a_1$ and $a_2$, those of weight 2 are $a_3$, $a_4$ and $a_5$, those of weight 3 are $a_6$ and $a_7$ and those of weight 4 are $a_8$, $a_9$ and $a_{10}$.

The presentation $\langle A \mid R \rangle$ is the input to Algorithm 3. It initialises $\bar{A} = A$ and $\bar{R}$ to be the set consisting of the relations

\[
[a_2, a_1] = : a_3, \quad a_1^2 = : a_4, \quad a_2^2 = : a_5, \quad [a_3, a_1] = : a_6,
\]

\[
[a_3, a_2] = : a_7, \quad [a_6, a_1] = : a_8, \quad [a_6, a_2] = : a_9, \quad [a_6, a_2] = : a_{10}.
\]

Next, the relations $[y, x] = 1$ where $\omega(x) + \omega(y) > 5$ are added to $\bar{R}$, namely the following relations: $[a_7, a_6] = 1$, $[a_6, x] = 1$, $[a_9, x] = 1$, $[a_{10}, x] = 1$ for $x \geq a_4$. Since $\langle A \mid R \rangle$ has class 4 it follows that $c = 4$. Thus $b$ runs from 5 to 2.

In the first iteration $b = 5$ and the following definitions are added to $\bar{R}$:

\[
[a_8, a_1] = : a_{11}, \quad [a_8, a_2] = : a_{12}, \quad [a_9, a_1] = : a_{13},
\]

\[
[a_9, a_2] = : a_{14}, \quad [a_{10}, a_1] = : a_{15}, \quad [a_{10}, a_2] = : a_{16}.
\]

The next step is to compute relations with left-hand side $u^2$ for $\omega(u) = 4$, such that $u$ is defined by a commutator. In this step the following relations are added to $\bar{R}$:

\[
a_5^2 = 1, \quad a_6^2 = 1, \quad a_{10}^2 = 1.
\]

Now the tails of relations with left-hand side a commutator $[z, u]$ with $\omega(u) = 2$ and $\omega(z) = 3$ are computed and the following relations are added to $\bar{R}$:

\[
[a_6, a_3] = a_{12}a_{13}, \quad [a_6, a_4] = a_{11}, \quad [a_6, a_5] = a_{14},
\]

\[
[a_7, a_3] = a_{14}a_{15}, \quad [a_7, a_4] = a_{13}, \quad [a_7, a_5] = a_{16}.
\]

In the next iteration $b = 4$ and the first step is to add to $\bar{R}$ the definitions, namely the relation $[a_7, a_1] = : a_{17}$. The next step is to compute relations with left-hand side $u^2$ for $\omega(u) = 3$, such that $u$ is defined by a commutator. In this step the following relations are added to $\bar{R}$:

\[
a_5^2 = a_{11}a_{12}a_{15}, \quad a_7^2 = a_{12}a_{15}a_{16}.
\]

Now the tails of relations with left-hand side a commutator $[z, u]$ with $\omega(u) = 2$ and $\omega(z) = 2$ are computed and the following relations are added to $\bar{R}$:

\[
[a_4, a_3] = a_8a_{11}a_{12}a_{15}, \quad [a_5, a_3] = a_{10}a_{12}a_{15}a_{16}, \quad [a_5, a_4] = a_9a_{11}a_{12}a_{13}a_{16}a_{17}.
\]

In the next iteration $b = 3$ and the first step is to add the definitions, namely the relations

\[
[a_4, a_1] = : a_{18}, \quad [a_4, a_2] = : a_{19}, \quad [a_5, a_1] = : a_{20}, \quad [a_5, a_2] = : a_{21},
\]

\[
a_4^2 = : a_{22}, \quad a_5^2 = : a_{23}.
\]

The next step is to compute relations with left-hand side $u^2$ for $\omega(u) = 2$, such that $u$ is defined by a commutator. In this step the following relation is added to $\bar{R}$:

\[
a_4^2 = a_8a_9a_{10}a_{14}a_{15}a_{20}.
\]

In the final iteration $b = 2$ and there is no change.

Now $\langle \bar{A} \mid \bar{R} \rangle$ is an inconsistent power-commutator presentation for the covering group on 23 generators. This presentation is the output of Algorithm 3.

In order to obtain a consistent power-commutator presentation for the same group we
follow the description of Section 3 to change the presentation to a consistent one. After renumbering the remaining new generators the resulting consistent power-commutator presentation has a set of generators

\[ \mathcal{A} = \{ a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}, a_{17}, a_{18} \} \]

and a set of relations \( \mathcal{R} \) containing

\[
\begin{align*}
1^2 &= 4, \\
2^2 &= 5, \\
3^2 &= 7, \\
4^2 &= 8, \\
5^2 &= 11, \\
6^2 &= 13, \\
7^2 &= 15, \\
8^2 &= 17, \\
9^2 &= 19, \\
10^2 &= 21, \\
11^2 &= 23, \\
12^2 &= 25, \\
&
\end{align*}
\]

and for \( x, y \in \{ a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}, a_{17}, a_{18} \} \) and \( y > x \) all relations of the form \( [y, x] = 1 \) and \( y^2 = 1 \) as well as for \( x \in \mathcal{A} \setminus \{ a_1, a_2 \} \) and \( y \in \{ a_8, a_9, a_{10} \} \) with \( y > x \) the relations of the form \( [y, x] = 1 \) and \( y^2 = 1 \).

\[ \diamond \]

Acknowledgements

We thank Joachim Neubüser and Cheryl Praeger for providing stimulating conditions for some of the writing of this paper. We thank Frank Celler for implementing the routines for manipulating tails in GAP 3 and George Havas and Eamonn O’Brien for valuable discussions. The second and third author acknowledge support of OPRSA and ANU PhD scholarships. The second author acknowledges the support of a postdoctoral scholarship from the ‘Graduiertenkolleg “Analyse und Konstruktion in der Mathematik”’ at the RWTH Aachen. The first and third authors also acknowledge the support of ARC Grant A69230241. We are indebted to a referee for saving us from some serious obscurities.

References


*Originally received 29 November 1995
Accepted 14 April 1997*