A Polycyclic Quotient Algorithm

EDDIE H. LO†

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08903, U.S.A.

This paper describes a generalization of the Gröbner basis method to the integral group ring of a polycyclic group. A polycyclic quotient algorithm is developed using this method. Suppose \( G \) is a group given by a finite presentation and \( G^{(n)} \) is the \( n \)th term in the derived series of \( G \). A polycyclic quotient algorithm computes the quotient \( G/G^{(n)} \) if it is polycyclic. An implementation of this algorithm in C has been developed and its efficiency is encouraging.

\[ \copyright \ 1998 \text{ Academic Press Limited} \]

In this paper, an algorithm to compute polycyclic quotients of a group given by a finite presentation is discussed. The algorithm is based on a generalization of the Gröbner basis method to the integral group ring of a polycyclic group. Most of this material can be found in the author’s thesis (Lo, 1996a). A simplified version of this paper is also available in Lo (1995).

1. Introduction

Definition 1.1. A polycyclic group is a group \( G \) with a subnormal series \( G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \cdots \triangleright G_n \triangleright G_{n+1} = 1 \) in which each of the quotients \( G_i/G_{i+1}, 1 \leq i \leq n \) is cyclic.

For \( 1 \leq i \leq n \), let \( a_i \) be an element of \( G_i \) such that the coset \( a_i G_{i+1} \) generates the quotient \( G_i/G_{i+1} \). Every \( x \) in \( G \) can be written in the form \( a_n^{\lambda_n} a_{n-1}^{\lambda_{n-1}} \cdots a_2^{\lambda_2} a_1^{\lambda_1} \) for some integers \( \lambda_1, \lambda_2, \ldots, \lambda_n \). Let \( I \) be the set \( \{ i \mid 1 \leq i \leq n \text{ and } |G_i/G_{i+1}| < \infty \} \), and for each \( i \) in \( I \), let \( m_i \) be \( |G_i/G_{i+1}| \). If in the form \( a_n^{\lambda_n} a_{n-1}^{\lambda_{n-1}} \cdots a_2^{\lambda_2} a_1^{\lambda_1} \), \( 0 \leq \lambda_i < m_i \) for all \( i \in I \), then the sequence of integers \( \lambda_n, \ldots, \lambda_2, \lambda_1 \) is unique. The form \( a_n^{\lambda_n} \cdots a_2^{\lambda_2} a_1^{\lambda_1} \) is called the collected form of \( x \) and will be denoted by \( \text{cf}(x) \). This definition of the collected form is different from the traditional definition in which elements are written in the form \( a_n^{\mu_n} a_2^{\mu_2} \cdots a_1^{\mu_1} \) for integers \( \mu_1, \mu_2, \ldots, \mu_n \). The reason for this new definition is that this is the definition which works in right ideals and right modules of \( \mathbb{Z}[G] \), the integral group ring of \( G \), for Gröbner basis computations, as we will see later.

Every polycyclic group \( G \) has a consistent polycyclic presentation, which is the analog of an AG-system in a finite solvable group (see Laue et al., 1984). For example, the group

† E-mail: hlo@math.rutgers.edu
described in the previous paragraph has a consistent polycyclic presentation given by

\[(a_1, a_2, \ldots, a_n \mid a_i^{m_i} = cf(a_i^{m_i}) \text{ for } i \in I, a_i^{a_j} = cf(a_i^{a_j}) \text{ for } 1 \leq j < i \leq n)\].

If a consistent polycyclic presentation is known, then there are practical algorithms to solve the word problem in \(G\), to describe the subgroups and quotients of \(G\) and to find normal closures of subsets of \(G\). More details can be found in Chapter 9 of (Sims, 1994).

Getting a consistent polycyclic presentation is in many cases the first step in computing in polycyclic groups.

For \(e \geq 1\), let \(G^{(e)}\) be the \(e\)th term in the derived series of \(G\).

**Definition 1.2.** Given a finite presentation of a group \(G\) and an integer \(e > 1\), a polycyclic quotient algorithm

1. decides whether the quotient \(G/G^{(e)}\) is polycyclic,
2. returns a consistent polycyclic presentation for \(G/G^{(e)}\) and a description for \(G/G^{(e)}\) if \(G/G^{(e)}\) is polycyclic, and
3. solves the word problem in \(G/G^{(e)}\) if \(G/G^{(e-1)}\) is polycyclic.

We will see formally what a description of a quotient is in Section 2.

It has been proved in Kharlampovič (1981) that there exists a finitely presented solvable group which has derived length 3 and which has an undecidable word problem. This gives an example of a group \(G\) where \(G/G'\) is polycyclic, \(G/G^{(2)}\) is not polycyclic but has a solvable word problem, and \(G/G^{(3)}\) has an unsolvable word problem.

Sims (1987) proved that given a finite presentation for a group \(G\), the polycyclicity of \(G\) can be verified. However, there is no algorithm which can verify that \(G\) is not polycyclic. In other words, there is no algorithm to decide whether \(G\) is polycyclic or not. An application of the polycyclic quotient algorithm is that, in the case when \(G\) is polycyclic, using the polycyclic quotient algorithm, we can find a consistent polycyclic presentation for \(G\).

Given a subgroup \(N\) of \(G\), we denote by \(N'\) the commutator subgroup \([N, N]\) of \(N\).

**Definition 1.3.** Given a finite presentation of a group \(G\), a polycyclic quotient \(G/N\) of \(G\), a consistent polycyclic presentation for \(G/N\) and a description for \(G/N\), an extension algorithm

1. decides whether the quotient \(G/N'\) is polycyclic,
2. returns a consistent polycyclic presentation for \(G/N'\) and a description for \(G/N'\) if \(G/N'\) is polycyclic, and
3. solves the word problem in \(G/N'\).

Given an extension algorithm, a polycyclic quotient algorithm can be obtained by simply applying the extension algorithm inductively with inputs \(N = G, G', G'', \ldots\).

Mathematicians have been trying to compute quotients of a group \(G\) defined by a finite presentation for nearly a century. Computing the abelian quotient \(G/G'\) was discussed by Tietze (1908). Effective algorithms for determining \(p\)-quotients were originated by the work of Macdonald (1973, 1974). His techniques were later extended by Wamsley (1974), Bayes et al. (1974), Newman (1976) and Havas and Newman (1980). Nickel (1994) developed and implemented an algorithm to compute the nilpotent quotients for finitely
presented groups. Computing nilpotent quotients is also covered in depth in Chapter 11 of Sims (1994). Research on determining finite solvable quotients can be found in Plesken (1987), Niemeyer (1994), Wegner (1992) and Havas and Robertson (1994). As for determining polycyclic quotients, in 1981, Baumslag, et al described a polycyclic quotient algorithm in Baumslag et al. (1981a) and Baumslag et al. (1981b). Sims (1990) implemented in Mathematica a practical algorithm to compute the metabelian quotient $G/G^{(2)}$. He generalized the Gröbner basis method to the integral group ring of a finitely generated abelian group. His approach is followed in this paper. The Gröbner basis method is generalized to the integral group ring of a polycyclic group to develop a practical polycyclic quotient algorithm. A computer implementation in the language C is developed for this algorithm to demonstrate its practicality.

2. Describing Quotient Groups

In this paper, the order of composition of functions is taken from right to left.

Given a set $X$, we denote by $F(X)$ the free group on $X$. Let $(X \mid R)$ be a finite presentation for $G$ and $N$ be a normal subgroup of $G$ such that $G/N$ is polycyclic. Assume that $(Y \mid S)$ is a consistent polycyclic presentation for $G/N$. To describe this quotient effectively, the normal subgroup $N$ has to be described in some way also. One approach is to determine a group homomorphism $\sigma$ from $F(X)$ to $F(Y)$. Let $\pi_X$ be the quotient map from $F(X)$ to $G$ and $\pi_Y$ be the quotient map from $F(Y)$ to $G/N$.

Suppose $\sigma : F(X) \rightarrow F(Y)$ is a homomorphism such that $\sigma(R)$ is contained in the normal closure of $S$ in $F(Y)$. Then $\sigma$ induces a homomorphism $\overline{\sigma}$ from $G$ to $G/N$ such that $\overline{\sigma} \circ \pi_X = \pi_Y \circ \sigma$. If $\overline{\sigma}$ is equal to the quotient map from $G$ to $G/N$, then $\sigma$ will be called a defining homomorphism for $G/N$ relative to the presentations $(X \mid R)$ and $(Y \mid S)$. Furthermore, if $\tau : F(Y) \rightarrow F(X)$ is a homomorphism such that $\overline{\sigma} \circ \pi_X \circ \tau = \pi_Y$, then the pair $(\sigma, \tau)$ is called a defining pair for $G/N$ relative to $(X \mid R)$ and $(Y \mid S)$. With a defining pair, the quotient $G/N$ can be described effectively.

**Proposition 2.1.** Suppose $(\sigma, \tau)$ is a defining pair for $G/N$ relative to $(X \mid R)$ and $(Y \mid S)$. Let $Q$ be the set $\{(\tau \circ \sigma)(x)x^{-1} \mid x \in X\} \cup \{\tau(s) \mid s \in S\}$. The normal closure of $\pi_X(Q)$ in $G$ is $N$.

**Proof.** Let $L$ be the normal closure of $\pi_X(Q)$ in $G$. First we will show that $L$ is a subgroup of $N$. Since $\pi_X$ is the projection map from $G$ to $G/N$, $L$ is a subgroup of $N$ if $(\overline{\sigma} \circ \pi_X)(q) = 1$ in $G/N$ for each $q \in Q$. Notice that $\overline{\sigma} \circ \pi_X = \pi_Y \circ \sigma$ and $\pi_Y \circ \sigma \circ \tau = \pi_Y$. Let $x$ be in $X$. We have $(\pi_Y \circ \sigma)((\tau \circ \sigma)(x)x^{-1}) = (\pi_Y \circ \sigma \circ \tau \circ \sigma)(x)(\pi_Y \circ \sigma)(x^{-1}) = (\pi_Y \circ \sigma)(x)(\pi_Y \circ \sigma)(x^{-1}) = 1$. If $s$ is in $S$, then $(\pi_Y \circ \sigma \circ \tau)(s) = \pi_Y(s) = 1$. Therefore $L \leq N$.

To show that $N$ is a subgroup of $L$, we take $W$ to be an element in $F(X)$ such that $\pi_X(W)$ is in $N$. We will see that $\pi_X(W)$ is in $L$. Since $L$ contains the element $\pi_X((\tau \circ \sigma)(x)x^{-1})$ for each $x$ in $X$, $\pi_X(W) \in (\pi_X \circ \tau \circ \sigma)(W)L$. Therefore $(\pi_Y \circ \sigma)(W) = (\overline{\sigma} \circ \pi_X \circ \tau \circ \sigma)(W) \in \overline{\sigma}(\pi_X(W)L) = \{1\}$ and $\sigma(W)$ is in the kernel of $\pi_Y$. Now the kernel of $\pi_Y$ is generated as a normal subgroup by $S$. Since $(\pi_X \circ \tau)(s) \in L$ for each $s \in S$ and $L$ is normal in $G$, $(\pi_X \circ \tau)(\sigma(W)) \in L$. So $\pi_X(W) \in (\pi_X \circ \tau \circ \sigma)(W)L = L$ and $N$ is a subgroup of $L$. □
When $S$ is finite, $|Q| \leq |X| + |S|$. By the above proposition, we can describe the normal subgroup $N$ as a normal closure of finitely many elements in $G$.

Since $N/N'$ is a normal abelian subgroup of $G/N'$, $N/N'$ can be viewed as a right $\mathbb{Z}[G/N']$-module under the conjugation action by $G/N'$. Moreover, $N/N'$ acts trivially on itself. Therefore $N/N'$ can be viewed as a $\mathbb{Z}[G/N]$-module. Under the conjugation action, the set $Q$ of normal generators for $N$ gives module generators for $N/N'$. The following corollary summarizes this paragraph.

**Corollary 2.2.** The abelian group $N/N'$ is generated as a right $\mathbb{Z}[G/N]$-module by the set $\{\tau(\sigma(x))x^{-1}N' \mid x \in X\} \cup \{\tau(s)N' \mid s \in S\}$. If $S$ is finite, then $N/N'$ is finitely generated as a right $\mathbb{Z}[G/N]$-module.

Let $M$ be the free right $\mathbb{Z}[G/N]$-module with rank equal to $|Q|$. Then $N/N'$ is isomorphic to $M/M_0$ for some right $\mathbb{Z}[G/N]$-submodule $M_0$ of $M$. To understand the extension $G/N'$ of $N/N'$, we study the module structure of $M/M_0$. A method to find a set of module generators for $M_0$ will be discussed in Sections 3 and 4. Then an algorithm to compute a “Gröbner basis” of $M_0$ from these module generators is described. The Gröbner basis can be used to determine whether $N/N'$ is finitely generated as an abelian group. If this is the case, then the polycyclicity of $G/N'$ follows and the Gröbner basis can be used to construct a consistent polycyclic presentation for $G/N'$, as we will see in Sections 10 and 11.

### 3. Extension Rewriting Systems

We begin this section by reviewing some facts about rewriting systems.

Suppose $X$ is a set. Let $X^\pm$ be a set of formal inverses of the elements in $X$ and $X^\pm = X \cup X^\pm$. We denote by $X^*$ the monoid $\{x_1x_2\ldots x_n \mid n \geq 0, x_i \in X\}$ for $1 \leq i \leq n$. Elements of $X^*$ are called words on $X$. Denote by $\epsilon$ the empty word. The multiplication in $X^*$ is given by concatenation.

**Definition 3.1.** A rewriting ordering on $X^*$ is a well-ordering $\leq_X$ on $X^*$ such that if $A, B, U, V$ are words on $X$, then $U \leq V$ implies $AUB \leq AVB$.

Rewriting orderings are called reduction orderings in Sims (1994).

**Examples 3.1.**

1. Let $\leq_X$ be a well-ordering on $X$. We define $u_1u_2\ldots u_m < v_1v_2\ldots v_n$ if either $m < n$, or $m = n$ and there exists an integer $i$ with $1 \leq i \leq n$ such that $u_j = v_j$ whenever $1 \leq j < i$ and $u_i <_X v_i$. Then $\leq$ is a rewriting ordering on $X^*$ and is called the length-plus-lexicographic ordering on $X^*$ relative to $\leq_X$.

2. Let $X$ and $Y$ be disjoint sets and $\leq_X$ and $\leq_Y$ be rewriting orderings on $X^*$ and $Y^*$, respectively. We will construct a rewriting ordering on $(X \cup Y)^*$. Suppose $U$ is a word on $X \cup Y$. Then $U$ can be written uniquely as $A_0b_1A_1b_2\ldots A_{r-1}b_rA_r$, where $r$ is a nonnegative integer, $b_1, b_2, \ldots, b_r$ are in $Y$ and $A_0, A_1, \ldots, A_r$ are words on $X$. Let $V$ be a word on $X \cup Y$ and let the corresponding decomposition of $V$ be $C_0d_1C_1d_2\ldots C_{s-1}d_sc_s$. Define $U < V$ if $b_1b_2\ldots b_r <_Y d_1d_2\ldots d_s$, or if $b_1b_2\ldots b_r = d_1d_2\ldots d_s$ and there exists an integer $i$ with $1 \leq i \leq r$ such that $A_j = C_j$ for $i < j \leq r$ and $A_i <_X C_i$. Then $\leq$ is a rewriting ordering on $(X \cup Y)^*$.
(see Sims, 1994, Section 2.1) and is called the reverse-wreath-product ordering on 
\((X \cup Y)^*\) relative to \(\leq_X\) and \(\leq_Y\).

Suppose \(\leq\) is a rewriting ordering on \(X^*\). Let \(\mathcal{R}\) be a relation on \(X^*\) and \(P > Q\) 
for each \((P, Q)\) in \(\mathcal{R}\). Such \(\mathcal{R}\) is called a rewriting system with respect to the rewriting 
ordering \(\leq\). A pair \((P, Q)\) in \(\mathcal{R}\) is called a rewriting rule in \(\mathcal{R}\). We call \(P\) the left side 
of the pair and \(Q\) the right side of the pair. We write \(U \rightarrow V\) and say that \(U\) reduces to \(V\) 
if for some words \(A, B, P, Q\) on \(X\), \(U = APB\), \(V = AQB\) and \((P, Q)\) is a rewriting rule 
in \(\mathcal{R}\). Since \(P > Q\), \(APB > AQB\). So \(U > V\) whenever \(U \rightarrow V\). A word \(U\) is irreducible 
with respect to \(\mathcal{R}\) if there does not exist any word \(V\) with \(U \rightarrow V\). We write \(U \rightarrow^* V\) if 
there exists a nonnegative integer \(l\) and \(U_1, U_2, \ldots, U_{l-1}\) such that \(U = U_0\), \(V = U_l\) and 
\(U_i \rightarrow U_{i+1}\) for \(0 \leq i \leq l - 1\). In other words, \(\rightarrow^*\) is the reflexive and transitive closure of 
\(\rightarrow\). Clearly \(U \geq V\) whenever \(U \rightarrow^* V\). The rewriting system \(\mathcal{R}\) is confluent if given any 
words \(U, V, W\) on \(X\) such that \(W \rightarrow^* U\) and \(W \rightarrow^* V\), there exists a word \(Q\) on \(X\) such 
that \(U \rightarrow^* Q\) and \(V \rightarrow^* Q\).

An example of a confluent rewriting system is a consistent polycyclic presentation. 
In the form described in Section 1, the rewriting ordering of this rewriting system is 
a combination of reverse-wreath-product orderings and the irreducible words are the 
collected forms of elements. Refer to Section 9.4 of Sims (1994) for detail.

The next three propositions are well-known in the literature of rewriting systems. For 
completeness their proofs are given. More detail can be found in Sections 2.2 and 2.3 of 
Sims (1994).

**Proposition 3.1.** Let \(\mathcal{R}\) be a confluent rewriting system on \(X^*\). If \(U, V\) and \(W\) are 
words on \(X\) such that \(U \rightarrow^* V\), \(U \rightarrow^* W\) and both \(V\) and \(W\) are irreducible, then \(V = W\).

**Proof.** Suppose that \(U, V, W\) are as above. Since \(\mathcal{R}\) is confluent, there exists \(Q\) such that 
\(V \rightarrow^* Q\) and \(W \rightarrow^* Q\). Now both \(V\) and \(W\) are irreducible. So \(V = Q = W\). \(\Box\)

We say that local confluence holds at a word \(W\) if for all words \(U, V\) on \(X\) such that 
\(W \rightarrow U\) and \(W \rightarrow V\), there exists a word \(Q\) such that \(U \rightarrow^* Q\) and \(V \rightarrow^* Q\).

**Proposition 3.2.** A rewriting system \(\mathcal{R}\) is confluent iff local confluence holds at all words.

**Proof.** If \(\mathcal{R}\) is confluent, then clearly local confluence holds at all words. Suppose now 
that local confluence holds at all words. Let us assume by contradiction that \(\mathcal{R}\) is not 
confluent. So there exists a word \(W\) such that \(W \rightarrow^* U\) and \(W \rightarrow^* V\) but there does not 
exist a word \(Q\) with \(U \rightarrow^* Q\) and \(V \rightarrow^* Q\). Since \(\leq\) is a well-ordering, we may take \(W\) 
to be the smallest such word. If \(W = U\), then we may take \(Q\) to be \(V\). This gives a 
contradiction. Let us assume \(W \neq U\). Similarly, assume \(W \neq V\). So there exists \(U'\) and 
\(V'\) such that \(W \rightarrow U' \rightarrow^* U\) and \(W \rightarrow V' \rightarrow^* V\). Local confluence holds at \(W\). So there 
exists \(W'\) such that \(U' \rightarrow^* W'\) and \(V' \rightarrow^* W'\). Note that \(W' < W\). Now \(U' < W\), \(U' \rightarrow^* U\) 
and \(U' \rightarrow^* W'\). By minimality of \(W\), there exists a word \(U''\) with \(U \rightarrow^* U''\) and \(W' \rightarrow^* U''\). 
Similarly, there exists a word \(V''\) with \(V \rightarrow^* V''\) and \(W' \rightarrow^* V''\). Moreover, \(W' \rightarrow^* U''\) and 
\(W' \rightarrow^* V''\) imply that there exists \(W''\) such that \(U'' \rightarrow^* W''\) and \(V'' \rightarrow^* W''\). By transitivity, 
\(U \rightarrow^* W''\) and \(V \rightarrow^* W''\). This is a contradiction. \(\Box\)
Proposition 3.3. Let $\mathcal{R}$ be a rewriting system. Suppose local confluence does not hold at a word $W$ but holds at all proper subwords of $W$. Then at least one of the following conditions holds:

1. $W$ is the left side of two different rules in $\mathcal{R}$.
2. $W$ is the left side of a rule in $\mathcal{R}$ and contains the left side of another rule as a proper subword.
3. $W$ can be written as $ABC$, where $A$, $B$, and $C$ are nonempty words and both $AB$ and $BC$ are left sides in $\mathcal{R}$.

In other words, $W$ can be written as $ABC$ where $B$ is a nonempty word and $AB$, $BC$ are left sides of two rewriting rules in $\mathcal{R}$.

Words of type 2 above will be called type 2 words. Words of type 3 above will be called overlaps.

Sims’ proof of Proposition 3.3 in Section 2.3 of Sims (1994) is sketched below.

Proof. Suppose that local confluence does not hold at $W$. There are words $A_1, B_1, P_1, Q_1$, $A_2, B_2, P_2, Q_2$ such that $W = A_1P_1B_1 = A_2P_2B_2$ and $(P_1, Q_1)$, $(P_2, Q_2)$ are different rules in $\mathcal{R}$. First assume that the occurrences of $P_1$ and $P_2$ in $W$ do not overlap. We may assume that $W = A_1P_1C_1P_2B_2$ for some word $C$. Reducing $W$ using the two rules we obtain the words $A_1Q_1C_2P_2B_2$ and $A_1P_1C_1Q_2B_2$. Both can be reduced to the same word $A_1Q_1C_1Q_2B_2$. Thus local confluence holds and this is a contradiction.

Now suppose that the occurrences of $P_1$ and $P_2$ in $W$ do overlap. Therefore $W = A_1ABCB_2$ where $B$ is nonempty, $P_1 = AB$ and $P_2 = BC$. We need to show that $A_1B_2$ are empty words. If they are not, then $ABC$ is a proper subword of $W$. So local confluence holds at $ABC$. Reducing $ABC$ using the two rules we obtain $Q_1C$ and $AQ_2$. Both can be reduced to a word $Q$ since local confluence holds. Therefore reducing $W$ using the two rules we obtain $A_1Q_1CB_2$ and $A_1AQ_2B_2$. Both can be reduced to $A_1QB_2$ and local confluence holds at $W$. Therefore $A_1B_2$ are empty words. So $W$ is equal to $ABC$ and is of one of the three types in the proposition. □

Let $(Y \mid S)$ be a consistent polycyclic presentation for a polycyclic group $G$. Suppose $S$ consists of the relations $U_1 = V_1, U_2 = V_2, \ldots, U_s = V_s$. Let $M$ be a right $\mathbb{Z}[G]$-module. We will describe a presentation for a group $K$ generated by the set $Z = Y \cup M$. The group $K$ will turn out to be an extension of a quotient of $M$ as an abelian group by the group $G$. To avoid confusion between the multiplication in $K$ and the module action of $G$ on $M$, elements in $M$ will be enclosed in square brackets when considered as elements in $K$. Within the brackets, multiplication denotes the module action. Outside the brackets, multiplication is the group multiplication in $K$. Thus if $m$ is an element in $M$ and $U$ is a word on $Y$, then $[mU]$ denotes the element in $K$ represented by the module element which is the result of $m$ acted on by the element in the group $G$ represented by $U$, while $[m][U]$ is the product of the two elements in $K$ represented by $[m]$ and $[U]$. The element $[0]$ is identified with the identity in $K$. We call a word of the form $[m][U]$, where $m$ is a nonzero element in $M$ and $U$ is a collected word on $Y$, a collected word on $Z$. If $m = 0$, then $[m][U]$ is identified with $U$. So a collected word on $Y$ will also be called a collected word on $Z$. 
Let \( \mu_1, \mu_2, \ldots, \mu_s \) be predetermined elements in \( M \). The defining relations for \( K \) are as follow:

1. \( [m_1] [m_2] \Rightarrow [m_1 + m_2], \quad m_1, m_2 \in M, \)
2. \( a[m] \Rightarrow [ma^{-1}]a, \quad m \in M, a \in Y^\pm, \)
3. \( U_i \Rightarrow [\mu_i] V_i, \quad 1 \leq i \leq s, \)

Let \( \pi_Z \) be the quotient map from \( F(Z) \) to \( K \) and \( \pi_Y \) be the quotient map from \( F(Y) \) to \( G \). Define \( \phi : K \to G \) by \( \phi(\pi_Z([m])) = 1 \) for every \( m \) in \( M \) and \( \phi(\pi_Z(a)) \) to be the element \( \pi_Y(a) \) in \( G \) for every \( a \) in \( Y \). Clearly \( \phi \) is an epimorphism from \( K \) to \( G \). Every element in \( K \) can be written as a collected word on \( Z \). To see this, we need to construct a rewriting system for \( K \).

We begin by constructing a rewriting ordering \( \leq \) on \( Z^{\pm*} \). Consider a well-ordering on the module \( M \). Extend this well-ordering to a well-ordering \( \leq \) on \( M^{\pm} \) such that for each \( m \) in \( M \), \( [m] < [m]^{-1} \). Let \( \leq_M \) be the length-plus-lexicographic ordering on \( M^{\pm*} \) relative to the well-ordering of \( M^{\pm} \). Since \( \langle Y \mid S \rangle \) is a consistent polycyclic presentation for \( G \), the set of relations \( S \) gives a confluent rewriting system. Define \( \leq \) on \( Z^{\pm*} \) to be the reverse-wreath-product ordering relative to \( \leq_Y \) and \( \leq_M \). Consider the rewriting system \( \mathcal{R} \) on \( Z^{\pm*} \) with the following rules:

1. \( [m_1] [m_2] \Rightarrow [m_1 + m_2], \quad m_1, m_2 \in M, \)
2. \( m^{-1} \Rightarrow [-m], \quad m \in M, \)
3. \( a^{-1} \Rightarrow \epsilon, \quad a \in Y^\pm, \)
4. \( a[m] \Rightarrow [ma^{-1}]a, \quad m \in M, a \in Y^\pm, \)
5. \( U_i \Rightarrow [\mu_i] V_i, \quad 1 \leq i \leq s, \)
6. \( [0] \Rightarrow \epsilon. \)

This rewriting system, which we denote by \( \mathcal{R} \), will be called the extension rewriting system relative to the right \( \mathbb{Z}[H] \)-module \( M \) and the elements \( \mu_1, \mu_2, \ldots, \mu_s \) in \( M \). Any word \( W \) on \( Z \) can be rewritten to a collected word on \( Z \) by applying the rules above finitely many times. By using the rules of type (2), we can rewrite \( W \) to the form \( W_1[W_2][W_3] \ldots W_r[m_r] W_{r+1}, \) where \( r \) is a nonnegative integer, \( m_1, m_2, \ldots, m_r \) are in \( M \) and \( W_1, W_2, \ldots, W_{r+1} \) are words on \( Y \). By repeated use of the rules of type (4), we can move elements in \( M \) to the left side of the word where they can be merged to form a single module element using the rules of type (1). Therefore \( W \) can be rewritten to the form \( [m] W_1 W_2 \ldots W_{r+1} \) with \( m \) in \( M \), or to simply \( W_1 W_2 \ldots W_{r+1} \) if \( m = 0 \). We are done if \( W_1 W_2 \ldots W_{r+1} \) is a collected word on \( Y \). Otherwise we can use the rules of type (3) or (5). If we use a rule \( U_i \Rightarrow [\mu_i] V_i, \) where \( 1 \leq i \leq s, \) then we introduce an element \([\mu_i]\) which can again be moved to the left side of the word. Since \( \langle Y \mid S \rangle \) is a consistent polycyclic presentation, repeating this process, eventually we will obtain a word on \( Z \) of the form \( [m'] W' \) where \( m' \) is in \( M \) and \( W' \) is a collected word on \( Y \), or to simply \( W' \) if \( m' = 0 \). In either case, \( W \) is rewritten to a collected word on \( Z \). For each rewriting rule above, the image under \( \phi \) of the left side is equal to the image under \( \phi \) of the right side. Therefore \( \phi(W) = \phi(W') \) is exactly the element in \( G \) given by \( W' \).

We would like this rewriting system to be confluent. By Proposition 3.3, since there are no two rules with the same left side, this rewriting system is confluent if local confluence holds at all type 2 words and all overlaps. We will work out a few examples here. The rest is very similar.
Examples 3.2.
1. Consider an overlap formed by two rules of type (1) in \( R \). Let \( m_1, m_2, m_3 \) be in \( M \). The overlap \([m_1][m_2][m_3]\) can be reduced in two ways. We can reduce it using the rule \([m_1][m_2] \rightarrow [m_1 + m_2]\) first or using the rule \([m_2][m_3] \rightarrow [m_2 + m_3]\) first. To indicate the order of reduction, we underline the subword which we reduce first. Since \([m_1][m_2][m_3] \rightarrow [m_1 + m_2][m_3] \rightarrow [m_1 + m_2 + m_3] \) and \([m_1][m_2][m_3] \rightarrow [m_1][m_2 + m_3] \rightarrow [m_1 + m_2 + m_3]\), local confluence holds at the overlap \([m_1][m_2][m_3]\).

2. Now consider an overlap formed by a rule of type (1) and a rule of type (4). Let \( m_1, m_2 \) be elements in \( M \) and \( a \) be in \( Y^\pm \). The overlap is \([a][m_1][m_2]\). Since \([a][m_1][m_2] \rightarrow [m_1 a^{-1}][m_2] \rightarrow [m_1 a^{-1}][m_2 a] \rightarrow [m_1 a^{-1} + m_2 a]a\) and \([a][m_1][m_2] \rightarrow a[m_1 + m_2] \rightarrow [(m_1 + m_2)a^{-1}][a]\), local confluence holds at this overlap.

3. Let us consider an overlap formed by a rule of type (4) and a rule of type (5). The left sides of the rules are \([a][m]\) and \(U_i\) respectively, where \( m \) is in \( M \), \( a \) is in \( Y^\pm \), \( 1 \leq i \leq s \), and as a word, \( U_i = Ua\) for some word \( U \). The overlap is \([U][a][m]\) and there are two ways to process this overlap. We can reduce it as \([U][a][m]\) which gives \([\mu_i + mV_i^{-1}][V_i]\), or we can reduce it as \([U][a][m]\) which yields the same result.

It can be checked that the only type (2) words and overlaps at which local confluence may not hold are formed from rules of type (3) and rules of type (5). Suppose \( W \) is one such word. Reducing \( W \) in two ways we obtain two irreducible collected words on \( Z \). Suppose they are \([m_1]W_1\) and \([m_2]W_2\). Since \( \phi(W_1) = \phi([m_1]W_1) = \phi(W) = \phi([m_2]W_2) = \phi(W_2) \) and \((Y \mid S)\) is a consistent polycyclic presentation, \( W_1 = W_2 \). Therefore local confluence holds at \( W \) iff \( m_1 = m_2 \) in \( M \). Let us denote by \( m_W \) the element \( m_1 - m_2 \) in \( M \). Since there are only finitely many rules of types (3) and (5), there are only finitely many such \( m_W \)'s. We call the right \( Z[G/N]\)-submodule generated by the set of \( m_W \)'s the overlap module for the extension rewriting system \( R \). Then \( R \) is confluent if the overlap module is trivial. This gives the following proposition.

**Proposition 3.4.** If \( R \) is the extension rewriting system relative to a right \( Z[H]\)-module \( M \) and module elements \( \mu_1, \mu_2, \ldots, \mu_s \), and \( M_0 \) is a right \( Z[H]\)-submodule of \( M \) containing the overlap module for \( R \), then the extension rewriting system relative to \( M/M_0 \) and the module elements \( \mu_1 + M_0, \mu_2 + M_0, \ldots, \mu_s + M_0 \) is confluent.

4. Forming Module Generators

We will continue to use the notation established in Sections 2 and 3. Assume that \( G \) has a finite presentation \((X \mid R)\), \( X = \{x_1, x_2, \ldots, x_r\} \) and \( G/N \) is given by a consistent polycyclic presentation \((Y \mid S)\). Let \( S \) consist of the relations \( U_1 = V_1, U_2 = V_2, \ldots, U_s = V_s \). Let \( \pi_X : F(X) \rightarrow G \) and \( \pi_Y : F(Y) \rightarrow G/N \) be the projection maps and \((\sigma, \tau)\) be a defining pair for \( G/N \). By Proposition 1.1, \( N \) is generated as a normal subgroup by the images of the elements in \( Q = \{(\tau \circ \sigma)(x)x^{-1} \mid x \in X\} \cup \{\tau(U_iV_i^{-1}) \mid 1 \leq i \leq s\} \) under \( \pi_X \). Let \( Z' = \{z_1, z_2, \ldots, z_r, z_1', z_2', \ldots, z_s'\} \) be a set of symbols, \( M \) be the free right \( Z[G/N]\)-module on \( Z' \) and \( Z \) be the set \( Y \cup M \). We would like to find a presentation \((Z \mid T)\) for \( G/N' \).

We construct homomorphisms \( \mu : F(X) \rightarrow F(Z) \) and \( \nu : F(Z) \rightarrow F(X) \) which will turn out to be a defining pair for \( G/N' \). It suffices to define \( \mu \) and \( \nu \) on \( X \) and \( Z \), respectively. Define \( \mu(x_i) \) to be \([z_i]^{-1}\sigma(x_i)\) for each \( x_i \) in \( X \). Define \( \nu(a) \) to be \( \tau(a) \) for \( a \) in \( Y \), \( \nu([z_i]) \) to be \( \tau(\sigma([z_i]))x_i^{-1} \) for \( 1 \leq i \leq r \) and \( \nu([z'_i]) \) to be \( \tau(U_iV_i^{-1}) \) for \( 1 \leq i \leq s \).
Define $\nu([0])$ to be the identity in $F(X)$ and $\nu([\zeta U])$ to be $\tau(U^{-1})\nu([\zeta])\tau(U)$ for $\zeta$ in $Z'$ and $U$ a collected word on $Y$. For every nonzero $m$ in $M$ such that $m$ is not of the form $\zeta U$, $m$ can be written as $\sum_{i=1}^{l} c_i \zeta U_i$ for $l > 1, c_i \in \mathbb{Z}$, $\zeta_i$ in $Z'$ and $U_i$ a collected word on $Y$. We define $\nu([m])$ to be $\prod_{i=1}^{l} (\nu([\zeta_i U_i]))^{c_i}$ and extend $\nu$ to a homomorphism from $F(Z)$ to $F(X)$.

For $1 \leq i \leq r$, $\nu(\mu(x_i)) = \nu([z_i]^{-1}\sigma(x_i)) = x_i \tau(\sigma(x_i))^{-1}\nu(\sigma(x_i))$. Since $\nu(a) = \tau(a)$ for $a$ in $Y$ and $\sigma(x_i)$ is in $F(Y)$, $\nu(\sigma(x_i)) = \tau(\sigma(x_i))$ and $\nu(\mu(x_i)) = x_i$. So $\nu \circ \mu$ is the identity on $F(X)$. In particular, $\nu$ is surjective. Let $\pi : G \to G/N'$ be the projection map. The homomorphism $\pi_Z = \pi \circ \pi_X \circ \nu$ is surjective since $\pi$, $\pi_X$ and $\nu$ are. We have defined an epimorphism from $F(Z)$ to $G/N'$. Since $\pi_Z \circ \mu = \pi \circ \pi_X \circ \nu \circ \mu = \pi \circ \pi_X$ and $\pi \circ \pi_X \circ \nu = \pi_Z$, $(\mu, \nu)$ is a defining pair for $G/N'$. Next we will find the kernel of $\pi_Z$.

**Proposition 4.1.** The group $N'$ is generated as a subgroup by the set $\pi_X(P)$, where $P = \{[W_1^{W_1}, W_2^{W_2}] \mid W_1, W_2 \in Q, W_1^W, W_2^W \text{ are words on } X\}$.

**Proof.** By Proposition 2.1, $N$ is generated as a normal subgroup by $\pi_X(Q)$. So $N$ is generated as a subgroup by $\{\pi_X(W^{W'}) \mid W \in Q, W' \text{ is a word on } X\}$ and $N'$ is generated as a normal subgroup by $\pi_X(P)$. Observe that the set $P$ is closed under conjugation by elements in $F(X)$. So $N'$ is generated as a subgroup by $\pi_X(P)$. \(\square\)

As a result of this proposition, the kernel of the homomorphism $\pi \circ \pi_X$ is the normal subgroup generated by $P \cup R$.

**Lemma 4.2.**

1. If $W$ is a word on $Y$, then $\nu(W) = \tau(W)$.
2. The element $(\pi_X \circ \nu)([m])$ is in $N$ for every $m$ in $M$.
3. If $W_1$ and $W_2$ are words in $Y$ such that $\pi_Y(W_1) = \pi_Y(W_2)$, then $(\pi_X \circ \nu)(W_1) = (\pi_X \circ \nu)(W_2)$ modulo $N$.

**Proof.**

1. This is true since by definition of $\nu$, $\nu(a) = \tau(a)$ for every $a$ in $Y$.
2. By inspecting the definition of $\nu$, it is clear from Proposition 2.1 that for every $\zeta$ in $Z'$, $(\pi_X \circ \nu)([\zeta])$ is contained in the normal subgroup $N$. If $\zeta$ is in $Z'$ and $U$ is a collected word on $Y$, then $(\pi_X \circ \nu)([\zeta U]) = \pi_X(\nu(U)^{-1}\nu([\zeta])\nu(U))$ is in $N$. The result for general $m$ in $M$ follows from the definition of $\nu([m])$.
3. Since $\pi_Y = \pi \circ \pi_X \circ \tau$, $(\pi \circ \pi_X \circ \tau)(W_1) = (\pi \circ \pi_X \circ \tau)(W_2)$. Now $W_1, W_2$ are words in $Y$. So $\tau(W_1) = \nu(W_1)$ and $\tau(W_2) = \nu(W_2)$. Recall that $\pi$ is the quotient map from $G$ to $G/N$. Therefore $(\pi_X \circ \nu)(W_1)$ is equal to $(\pi_X \circ \nu)(W_2)$ modulo $N$. \(\square\)

Since $\pi_Z$ is onto, a presentation for $G/N'$ can be obtained using the following theorem.

**Theorem 4.3.** The kernel of $\pi_Z$ is generated by the following relations:

\[
\begin{align*}
(1) \quad [m_1, m_2] & = [m_1 + m_2], \quad m_1, m_2 \in M, \\
(2) \quad a[m] & = [ma^{-1}]a, \quad m \in M, a \in Y^\pm, \\
(3) \quad U_i & = [z_i]V_i, \quad 1 \leq i \leq s, \\
(4) \quad a & = \mu(\nu(a)), \quad a \in Y, \\
(5) \quad \mu(W) & = 1, \quad W \in R.
\end{align*}
\]
PROOF. Let $L_1$ be the normal subgroup generated by the above relations and let $L_2$ be the kernel of $\pi_Z$.
First we will show that $L_1 \subseteq L_2$.

By Lemma 4.2 part 2, for every $m$ in $M$, $\pi_Z([m]) = (\pi \circ \pi_X \circ \nu)([m])$ is in $N/N'$. So if $m_1$ and $m_2$ are elements in $M$, then $\pi_Z([m_1])$ and $\pi_Z([m_2])$ commute. The relations of type (1) follow from the definition of $\nu([m])$ for $m$ in $M$ and the fact that $M$ is free.
To show the relations of type (2) are in $L_2$, it suffices to show they are true when $m$ is $\zeta U$ for $\zeta$ in $Z'$ and $U$ is a collected word on $Y$ since relations of type (1) hold. Assume that $a$ is in $Y^k$. Let $V$ be the collected form of $Ua^{-1}$. Then $\pi_Y(V) = \pi_Y(Ua^{-1})$ and by Lemma 4.2 part 3, $\pi_Z(V)h = \pi_Z(Ua^{-1})$ for some $h$ in $N/N'$. Therefore
\[
\pi_Z(a[\zeta U]a^{-1}) = \pi_Z(a)\pi_Z(U^{-1})\pi_Z(\zeta)\pi_Z(U)\pi_Z(a^{-1})
\]
\[
= h^{-1}\pi_Z(V)^{-1}\pi_Z(\zeta)\pi_Z(V)h = h^{-1}\pi_Z(\zeta)V)h = \pi_Z(\zeta(V)a^{-1}).
\]
By definition, $\nu([\zeta]) = \zeta(U_iV_i^{-1}) = \nu(U_iV_i^{-1})$. So $\zeta(U_i) = \nu([\zeta])V_i$ and
\[
\pi_Z(U_i) = (\pi \circ \pi_X \circ \nu)(U_i) = (\pi \circ \pi_X \circ \nu)([\zeta])V_i = \pi_Z([\zeta])V_i.
\]
So the relations of type (3) are true in $L_2$. The relations of types (4) and (5) are true in $L_2$ since $\nu \circ \mu$ is the identity homomorphism in $F(X)$. For every $a$ in $Y$,
\[
\pi_Z(\mu \circ \nu)(a) = (\pi \circ \pi_X \circ \nu \circ \mu \circ \nu)(a) = (\pi \circ \pi_X \circ \nu)(a) = \pi_Z(a).
\]
For every $W$ in $R$,
\[
\pi_Z(\mu(W)) = (\pi \circ \pi_X \circ \nu \circ \mu)(W) = (\pi \circ \pi_X)(W) = 1.
\]
Next we will show that $L_2 \subseteq L_1$. First, we claim that every element in $F(Z)$ is equal to its own image under $\mu \circ \nu$ modulo $L_1$. Second, we claim that the image of the kernel of $\pi \circ \pi_X$ under $\mu$ is a subset of $L_1$. For the first claim, let $1 \leq i \leq 3$. We have $\mu(\nu([\zeta])) = \mu(\nu(U_iV_i^{-1})) = \nu(U_iV_i^{-1})$. So by the relations of types (3) and (4), $\mu([\zeta]) = U_iV_i^{-1} = \pi_Z(\zeta)$ modulo $L_1$. Now let $1 \leq i \leq 7$, we have $\mu(\nu([\zeta])) = \mu(\pi(\nu(x_0)))\mu(x_0^{-1}) = \mu(\nu(\pi(x_0)))\mu(x_0^{-1})$. Since $\pi(x_0)$ is a word on $Y$, $\mu(\nu(\pi(x_0))) = \pi(x_0)$ modulo $L_1$. So $\mu(\nu([\zeta])) = \pi(x_0)$ modulo $L_1$. Every element in $Y \cup Z'$ is, modulo $L_1$, equal to its own image under $\mu \circ \nu$. By inspecting the relations of types (1) and (2), we see that $Y \cup Z'$ generates $F(Z)$ modulo $L_1$. Therefore every element in $F(Z)$ is equal to its own image under $\mu \circ \nu$ modulo $L_1$. Now let us prove the second claim. It suffices to show that $\mu(P \cup R)$ is contained in $L_1$. By the remark following Proposition 4.1, $P \cup R$ generates the kernel of $\pi \circ \pi_X$ as a normal subgroup. By the definition of $\nu$, any element in $Q$ is equal to $\nu([\zeta])$ for some $\zeta$ in $Z'$. Using the first claim, the image of any element in $Q$ under $\mu$ is, modulo $L_1$, equal to $[\zeta]$ for some $\zeta$ in $Z'$. As a result, the image of any element in $P$ under $\mu$ is, modulo $L_1$, equal to a commutator of two elements of the form $[m]$ where $m$ is in $M$. By the relations of type (1), this commutator is in $L_1$. Therefore $\mu(P)$ is a subset of $L_1$. Using the relations of type (5), we can see that $\mu(R)$ is also a subset of $L_1$. This proves the second claim. So if $1 = \pi_Z(W) = (\pi \circ \pi_X \circ \nu)(W)$ for some element $W$ in $F(Z)$, then the second claim implies that $\mu(\nu(W))$ is in $L_1$. Now $W$ is equal to $\mu(\nu(W))$ modulo $L_1$ by the first claim. Therefore $W$ is contained in $L_1$. This proves the theorem. □

Given the relations of types (1), (2) and (3), we can find an extension rewriting system relative to the right $\mathbb{Z}[G/N]$-module $M$ and the elements $z_1', z_2', \ldots, z_r'$. The relations of types (4) and (5) can be rewritten using this rewriting system. Suppose $a$ is an element
in $Y$. Consider the relation $a = \mu(\nu(a))$. The word $\mu(\nu(a))$ can be rewritten to a word of the form $[m_1]W'$ or to $W'$ when $m_1 = 0$, where $W'$ is the collected form of $a$. Similarly, the word $a$ can be rewritten to a word $[m_2]W'$ or to $W'$ when $m_2 = 0$. Let $m = m_1 - m_2$ in this case. Now suppose that $W$ is in $R$. Then $W$ represents the identity element in $G$. When we rewrite the word $\mu(W)$, we obtain the element $[m]$ for some $m$ in $M$, or simply the identity in $K$ when $m = 0$. Let $M_0$ be the submodule of $M$ generated by the overlap module and the set of module elements $m$ found by rewriting the relations of types (4) and (5) as described above.

**Theorem 4.4.** With the notation described above, $M/M_0$ and $N/N'$ are isomorphic as right $\mathbb{Z}[G/N]$-modules.

**Proof.** Consider the extension rewriting system relative to the right $\mathbb{Z}[G/N]$-module $M/M_0$ and the elements $z_1'M_0, z_2'M_0, \ldots, z_i'M_0$. Let $\mathcal{G}$ be the group associated with this rewriting system. The groups $F(Z)/L_1$ and $\mathcal{G}$ are generated by the same alphabet set and it is routine to check that they satisfy the same relations. Therefore they are isomorphic via the map $\rho$ taking the image of $[m + M_0]$ in $\mathcal{G}$ to $[m]L_1$ for $m$ in $M$, and the image of $a$ in $\mathcal{G}$ to $aL_1$ for $a$ in $Y$. Let $\pi$ be the quotient map induced from $\pi_G$. Then $\pi$ gives an isomorphism between the groups $F(Z)/L_1$ and $G/N'$. Therefore $\pi \circ \rho$ is an isomorphism from $\mathcal{G}$ to $G/N'$. It is easy to see that under $\pi \circ \rho$, $M/M_0$ is mapped to $N/N'$. So $M/M_0$ and $N/N'$ are isomorphic as right $\mathbb{Z}[G/N]$-modules. □

5. Group Reduction Orderings

In this section, we begin to study the Gröbner basis method in finitely generated right $\mathbb{Z}[G]$-modules. A good reference for Gröbner bases is Becker and Weispfenning (1991).

Let $S$ be a set. A partial ordering $\preceq$ is a reflexive, antisymmetric and transitive relation on $S$. We call a set $S$ with a partial ordering $\preceq$ defined a poset. Let $S'$ be a subset of a poset $S$. We define $\text{min}(S')$ to be $\emptyset$ if $S' = \emptyset$. Otherwise $\text{min}(S')$ is the set $\{s \in S' \mid \forall s' \in S', s' \not\preceq s\}$. A descending chain in $S$ is a sequence of elements $s_1, s_2, \ldots$ such that $s_i \succ s_j$ whenever $i < j$. An antichain $S'$ in $S$ is a subset of $S$ such that whenever $s_1$ and $s_2$ are in $S'$, $s_1 \not\preceq s_2$.

**Theorem 5.1.** Let $S$ be a poset with partial ordering $\preceq$. The followings conditions on $S$ are equivalent.

1. For any infinite sequence $s_1, s_2, \ldots$ of elements in $S$, there exists $i, j$ such that $i < j$ and $s_i \not\preceq s_j$.
2. If $S'$ is a nonempty subset of $S$, then $\text{min}(S')$ is nonempty and finite.
3. The poset $S$ has no infinite descending chain and no infinite antichain.

**Proof.** Condition 1 clearly implies condition 3. Any infinite descending chain and any infinite antichain will give an infinite sequence $s_1, s_2, \ldots$ such that $s_i \not\preceq s_j$ for $i < j$. To prove that condition 3 implies condition 2, we take a nonempty subset $S'$ of $S$. Take an element $s_1 \in S'$. If $s_1 \not\preceq \text{min}(S')$, then there exists $s_2$ in $S'$ with $s_1 \succ s_2$. If $s_2 \not\preceq \text{min}(S')$, then there exists $s_3$ in $S'$ with $s_1 \succ s_2 \succ s_3$. Continue this process until we have found an element in $\text{min}(S')$. Since $S$ has no infinite descending chain, this process must stop. So $\text{min}(S')$ is nonempty. It is clear that $\text{min}(S')$ is an antichain and so $\text{min}(S')$ is finite.
It remains to show that condition 2 implies condition 1. Let \( s_1, s_2, \ldots \) be a sequence of elements in \( S \). We may assume that the elements in the sequence are distinct. Let \( S_k = \{ s_i \mid i \geq k \} \). Since \( \min(S_1) \) is finite, \( S_l \cap \min(S_l) \) is empty for some \( l \). Let \( s_j \) be in \( \min(S_l) \). Now \( s_j \notin \min(S_1) \). So there exists \( i < l \) such that \( s_i \preceq s_j \). Since \( s_j \in \min(S_l) \), 
\( l \leq j \). This proves the theorem. \( \Box \)

**Definition 5.1.** A poset \( S \) is a well-poset if it satisfies any of the equivalent conditions in Theorem 5.1. If \( S \) is a well-poset, then the partial ordering on \( S \) is called a well-partial-ordering.

**Examples 5.1.**
1. Let \( N \) be the set of nonnegative integers and for \( s_1, s_2 \) in \( N \), define \( s_1 \preceq s_2 \) if \( s_1 \leq s_2 \) under the usual ordering on \( N \). Under \( \preceq \), \( N \) is a well-poset.
2. Let \( S \) be a set of \( n \geq 1 \) elements. For every \( s_1, s_2 \) in \( S \), define \( s_1 \preceq s_2 \) if \( s_1 = s_2 \). In other words, \( s_1 \neq s_2 \) for every distinct pair \( s_1, s_2 \) in \( S \). Under \( \preceq \), \( S \) is a well-poset. In fact, any finite poset is a well-poset. We call this poset the discrete poset of \( n \) elements and denote it by \( \Phi_n \).
3. Let \( S \) be a poset with partial ordering \( \preceq \). A subposet of \( S \) is a subset \( T \) of \( S \) with a partial ordering, again denoted by \( \preceq \), defined by \( t_1 \preceq t_2 \) in \( T \) if \( t_1 \preceq t_2 \) in \( S \). If \( T \) is a subposet of a well-poset \( S \), then \( T \) is a well-poset also. On the other hand, if \( T \) is a subposet of a poset \( S \), \( T \) is a well-poset, and \( |S - T| \) is finite, then \( S \) is a well-poset.

Let \( S \) and \( T \) be two posets with partial orderings \( \preceq_S \) and \( \preceq_T \), respectively. The product poset \( S \times T \) is defined as the usual Cartesian product with the partial ordering \( \preceq \) defined by \((s_1, t_1) \preceq (s_2, t_2)\) if \( s_1 \preceq_S s_2 \) and \( t_1 \preceq_T t_2 \).

**Theorem 5.2.** The product poset of two well-posets is a well-poset.

The proof of this theorem is clear using characterization (3) in Theorem 5.1.

By induction, if \( S_1, S_2, \ldots, S_k \) are well-posets, then so is their product poset \( S_1 \times S_2 \times \cdots \times S_k \). For any poset \( S \), we write \( S^k \) to denote the product poset \( S \times S \times \cdots \times S \).

**Proposition 5.3.** Let \( S \) be a well-poset and let \( S' \) be a nonempty subset of \( S \). For any \( s \) in \( S' \), there exists \( s' \) in \( \min(S') \) such that \( s' \preceq s \).

The proof of this proposition is also clear using characterization (3) in Theorem 5.1.

**Definition 5.2.** Let \( \preceq \) be a partial ordering on a set \( S \). Suppose \( \preceq \) is a total ordering on \( S \) such that whenever \( s_1 \prec s_2 \) in \( S \), \( s_1 \prec s_2 \). Then we say that \( \preceq \) is a linear extension of \( \preceq \).

**Proposition 5.4.** Let \( S \) be a well-poset with partial ordering \( \preceq \). Any linear extension \( \preceq \) of \( \preceq \) is a well-ordering.

**Proof.** Let \( T \) be a nonempty subset of \( S \). Since \( \min(T) = \{ s \in T \mid \forall t \in T, t \neq s \} \) is nonempty and finite, it is possible to find an element \( s \) in \( \min(T) \) such that \( s \preceq s' \) for each \( s' \) in \( \min(T) \). Now if \( t \) is an element in \( T \), then by Proposition 5.3, \( t \preceq s' \) for some \( s' \) in \( \min(T) \). So \( t \preceq s' \preceq s \). \( \Box \)
Suppose $\preceq$ is a partial ordering defined on a set $S$. Let $s_1$ and $s_2$ be elements in $S$. Denote by $\text{lcm}(s_1, s_2)$ the set $\min\{s \in S \mid s_1 \preceq s \preceq s_2\}$. If $S$ is a well-poset with partial ordering $\preceq$, then for any elements $s_1, s_2$ in $S$, $\text{lcm}(s_1, s_2)$ is finite.

**Definition 5.3.** A poset $S$ with partial ordering $\preceq$ is said to have the unique LCM property if for every $s_1, s_2$ in $S$, $\text{lcm}(s_1, s_2)$ has at most one element. Suppose $S$ has the unique LCM property. If $s_1, s_2$ are in $S$ and $\text{lcm}(s_1, s_2) = \{s\}$, then the element $s$ is called the lcm of $s_1$ and $s_2$ in $S$. By abuse of notation, we write $s$ as $\text{lcm}(s_1, s_2)$.

If two posets $S$ and $T$ have the unique LCM property, then so does the product poset $S \times T$. In fact, if $(s_1, t_1), (s_2, t_2)$ are in $S \times T$, then $\text{lcm}((s_1, t_1), (s_2, t_2)) = (\text{lcm}(s_1, s_2), \text{lcm}(t_1, t_2))$ if it exists.

**Definition 5.4.** A group $G$ with a partial ordering $\preceq$ and a total ordering $\preceq$ defined is said to be a reduction group if

1. $\preceq$ is a well-partial-ordering with the unique LCM property,
2. $\preceq$ is a well-ordering, and,
3. for every $x_1, x_2, x'$ in $G$, if $x_1 \preceq x_2$ and $x' \preceq x_1$, then $x' x_1^{-1} x_2 < x_2$.

We call $\preceq$ a group reduction ordering with respect to $\preceq$.

**Examples 5.2.**

1. Let $G$ be the infinite cyclic group generated by $a$. Define $a^0 < a^1 < a^2 < \cdots$ and $a^0 < a^{-1} < a^{-2} < \cdots$. With this partial ordering, $G$ is a well-poset with the unique LCM property. In fact, referring to Examples 5.1, we can see that $G$ as a poset has the same structure as $(\mathbb{Z} \times \mathbb{Z}) \cup \{0\}$. If we define $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \cdots$, then $G$ is a reduction group. Another group reduction ordering on $G$ is given by $a^0 < a^1 < a^2 < \cdots < a^{-1} < a^{-2} < \cdots$. In fact, any linear extension of $\preceq$ is a group reduction ordering.

2. Let $G$ be the finite cyclic group of order $m$. Suppose $a$ generates $G$. Define $a^0 < a^1 < a^2 < \cdots < a^m$. Clearly $\preceq$ is a well-partial-ordering with the unique LCM property. One group reduction ordering $\preceq$ in this case is given by $a^0 < a^1 < a^2 < \cdots < a^{m-1}$.

3. Let $G$ be a finite group of order $m$. We may give $G$ the discrete poset structure. Again, $G$ is a well-poset with the unique LCM property. The group $G$ is a reduction group given any total ordering.

Let $G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \cdots \triangleright G_{c+1} = 1$ be a subnormal series of $G$. From here on in this section, $i$ is an integer and $1 \leq i \leq c$. For each $i$, let $Y_i$ be a set of coset representatives of $G_{i+1}$ in $G_i$ and let $Y = \bigcup_{i=1}^c Y_i$. For all $x$ in $G$, $x$ can be written uniquely as $y_1 y_{e-1} \cdots y_2 y_1$ where each $y_i$ is in $Y_i$. We call $y_e y_{e-1} \cdots y_2 y_1$ the $Y$-form of $x$. Let $H_i$ be the quotient $G_i/G_{i+1}$ and $\pi_i : G_i \rightarrow H_i$ be the quotient map. Assume that for each $i$ we can define a partial ordering $\preceq_i$ on $H_i$. Then we can extend these partial orderings to a partial ordering on $G$. Let $x$ and $x'$ be elements in $G$ with $Y$-forms $y_e y_{e-1} \cdots y_2 y_1$ and $y'_e y'_{e-1} \cdots y'_2 y'_1$, respectively. Define $x \preceq x'$ if $\pi_i(y_e) \preceq_i \pi_i(y'_e)$.
for each $i$. As a poset, $G$ is just the product poset of the posets $H_1, H_2, \ldots, H_e$. If $H_i$ is a well-ordered set with the unique LCM property for each $i$, then so is $G$. For each $i$, let $\leq_i$ be a total ordering on $H_i$. We can extend these total orderings to a total ordering on $G$. Assume that the $Y$-forms of $x$ and $x'$ are as above. We define $x < x'$ if there exists $j$ with $1 \leq j \leq e$ such that $y_1 = y'_1, y_2 = y'_2, \ldots, y_{j-1} = y'_{j-1}$, and $\pi_j(y_j) <_j \pi_j(y'_j)$.

**Definition 5.5.** Using the above notation, we call the partial ordering $\preceq$ and the total ordering $\leq$, respectively, the standard partial ordering and the standard total ordering of $G$ relative to the subnormal series, the set $Y$, the partial orderings $\preceq_i$ and the total orderings $\leq_i$.

For $x_1$ and $x_2$ in $G$, let $\frac{x_2}{x_1}$ denote the element $x_1^{-1}x_2$.

**Theorem 5.5.** Using the notation above, if for each $i$ $H_i$ is a reduction group with partial ordering $\preceq_i$ and total ordering $\leq_i$, then with the standard partial ordering $\preceq$ and the standard total ordering $\leq$, $G$ is a reduction group.

**Proof.** Condition 1 in the definition of a group reduction ordering is easy to see. We prove condition 2 by induction on $e$, the length of the subnormal series. If $e = 1$, then this is trivial. For the induction step, suppose that condition 2 holds when the length of the subnormal series is equal to $e-1$. Let $X$ be a nonempty subset of $G$. Let $y_1$ be such that $\pi_1(y_1)$ is the minimal element in $\pi_1(X)$ under the total ordering $\leq_1$. Let $X'$ be the set $\{y \in G \mid y y_1 \in X \}$. Then $X'$ is non-empty. By the induction hypothesis the standard total ordering on $G_2$ relative to the subnormal series $G_2 \triangleright G_3 \triangleright \cdots \triangleright G_{e+1} = 1$, the set $\bigcup_{i=2}^e Y_i$ and the orderings $\preceq_i$ and $\leq_i$ are well-orderings. Notice that this standard total ordering coincides with the standard total ordering $\leq$ restricted to $G_2$. Let $y'$ be a minimal element in $X'$. It is now routine to check that $y'y_1$ is the minimal element in $X$ under $\leq$.

We prove condition 3 again by induction on $e$. If $e = 1$ then this is trivial. For the case $e = 2$, the map $\pi_2$ is just the identity map on $G_2$ and will not be written. To simplify the exposition, let us write $\pi$ for $\pi_1$. Assume that $x, x'$ and $x''$ are elements in $G$ with $Y$-forms $y_2y_1$, $y_2y'_1$ and $y_2y''_1$, respectively. We want to show that if $x < x'$ and $x' \preceq x''$, then $x \frac{x''}{x'} < x''$. Since $x < x'$, either $\pi(x) = \pi(y_1) <_1 \pi(y'_1) = \pi(x')$, or $y_1 = y'_1$ and $y_2 < y'_2$. In the first case, we have $\pi(x) <_1 \pi(x')$ and $x' \preceq x''$. Using the fact that the total ordering on $H_1$ is a group reduction ordering, $\pi(x) \frac{\pi(x'')}{\pi(x')} <_1 \pi(x'')$. So $\pi(x) \frac{x''}{x'} <_1 \pi(x'')$ and $x \frac{x''}{x'} < x''$. In the second case, $y_1 = y'_1$ and $y_2 < y'_2$. So $x \frac{x''}{x'} = x(x')^{-1}x'' = y_2y_1(y'_1)^{-1}(y''_2)^{-1}y''_2y''_1 = y_2(y''_2)^{-1}y''_2y''_1 = y_2y''_2y''_1$. If the $Y$-form of $x \frac{x''}{x'}$ is $y_2y''_1$, then $y'_1 = y''_1$ and $y'_2 = y_2y''_2$. So $\pi(x) \frac{x''}{x'} = \pi(y''_1) = \pi(x'')$. Since $y_2 < y'_2$, $y'_2 \preceq y''_2$ and the total ordering on $H_2$ is a group reduction ordering, $y_2 \frac{y''_2}{y'_2} < y''_2$. Therefore $y'_2 < y''_2$ and $x \frac{x''}{x'} < x''$. The case $e = 2$ is proved. For the induction step, assume that the theorem is true when the length of the subnormal series is equal to $e-1$. Therefore with respect to the standard partial and total ordering on $G_2$, $G_2$ is a reduction group. Now consider the subnormal series $G = G_1 \triangleright G_2 \triangleright G_{e+1} = 1$. It is routine to check that the standard total ordering and standard partial ordering on $G$ relative to this series and the set $Y_1 \cup G_2$ coincide with $\leq$ and $\leq$, respectively. By the result for


\[ e = 2, \ G \text{ is a reduction group with the standard partial ordering } \leq \text{ and the standard total ordering } \preceq. \ \Box \]

We remark here that if for each \( i, \preceq_i \) is a linear extension of \( \preceq_i \), then the standard total ordering \( \leq \) is also a linear extension of the standard partial ordering \( \preceq \). Note that all the group reduction orderings in Examples 5.2 are linear extensions of the respective partial orderings.

**Corollary 5.6.** If \( G \) is polycyclic-by-finite, then there exists a group reduction ordering on \( G \) such that the underlying partial ordering has the unique LCM property.

**Proof.** In Example 5.2 group reduction orderings on all finite groups and the infinite cyclic group are illustrated. If \( G \) is polycyclic-by-finite, then \( G \) has a subnormal series such that each quotient is finite or cyclic. Since each quotient is a reduction group, by Theorem 5.5, so is \( G \). \( \Box \)

Let \( G \) be a group. We will consider submodules of a free right \( \mathbb{Z}[G] \)-module \( \mathbb{Z}[G]^k \), \( k \geq 1 \). An element in \( \mathbb{Z}[G]^k \) is called a *polynomial array*. Every polynomial array in \( \mathbb{Z}[G]^k \) is a \( k \)-tuple \((f_1, f_2, \ldots, f_k)\) where \( f_i \) is in \( \mathbb{Z}[G] \) for \( 1 \leq i \leq k \). Let \( \{e_1, e_2, \ldots, e_k\} \) be the standard module basis of \( \mathbb{Z}[G]^k \). Every element in \( \mathbb{Z}[G]^k \) is a finite sum of elements of the form \( ec_i x \), where \( 1 \leq i \leq k \), \( c \) is a nonzero integer and \( x \) is an element in \( G \). We call such an element a *monomial vector*. An element in \( \mathbb{Z}[G]^k \) of the form \( e_i x \) is called a *term vector*. Let \( \mathcal{M} \) be the set of monomial vectors and \( \mathcal{T} \) be the set of term vectors.

**Definition 5.6.** Suppose \( G \) is a reduction group and \( \mathcal{T} \) is the set of term vectors in \( \mathbb{Z}[G]^k \). Let \( \preceq \) be a partial ordering and \( \leq \) be a total ordering on \( \mathcal{T} \) such that

1. \( \preceq \) is a well-partial-ordering with the unique LCM property,
2. \( \leq \) is a well-ordering,
3. for every \( t_1, t_2 \) in \( \mathcal{T} \) such that \( t_1 \preceq t_2 \), there exists an element \( x \) in \( G \), denoted by \( \frac{t_1}{t_2} \), such that \( t_1 x = t_2 \), and,
4. for every \( t_1, t_2, t' \) in \( \mathcal{T} \), if \( t_1 < t_2 \) and \( t' < t_1 \), then \( t' \frac{t_1}{t_2} < t_2 \).

Then \( \preceq \) is called a reduction partial ordering on \( \mathbb{Z}[G]^k \) and \( \leq \) is called a reduction total ordering on \( \mathbb{Z}[G]^k \) associated with \( \preceq \).

Suppose \( G \) is a reduction group. Let the partial ordering on \( G \) be denoted by \( \preceq \) and the total ordering on \( G \) be denoted by \( \leq \). It is possible to extend \( \preceq \) and \( \leq \) to \( \mathcal{T} \) to give a reduction partial ordering and a reduction total ordering, respectively. Given two term vectors \( t_1 = e_i x_1, t_2 = e_j x_2 \), we define \( t_1 \preceq t_2 \) if \( x_1 \preceq x_2 \) in \( G \) and \( i = j \). This poset structure on \( \mathcal{T} \) is just the product poset of the poset \( G \) and the discrete poset \( \Phi_k \). By Theorem 5.2, \( \mathcal{T} \) is a well-poset with the unique LCM property. One way of extending \( \preceq \) to \( \mathcal{T} \) is to define \( t_1 < t_2 \) if either \( i < j \), or \( i = j \) and \( x_1 < x_2 \) in \( G \). It is routine to check that \( \preceq \) is a reduction partial ordering and \( \leq \) is a reduction total ordering associated with \( \preceq \) on \( \mathbb{Z}[G]^k \).

We extend \( \preceq \) and \( \leq \) further to \( \mathcal{M} \). Let the set of nonzero integers \( \mathbb{Z} - \{0\} \) be ordered as \( 1 \ll 2 \ll \cdots \ll -1 \ll -2 \ll \cdots \). With this partial ordering, \( \mathbb{Z} - \{0\} \) is a well-poset. The set of monomial vectors \( \mathcal{M} \) can be given a poset structure by viewing \( \mathcal{M} \) as the product
poset of \( \mathbb{Z} - \{0\} \) and \( T \) with their respective partial orderings. By Theorem 5.2, \( \mathcal{M} \) is a well-poset. For the total ordering on \( \mathcal{M} \), let \( c_1, c_2 \) be integers and \( t_1, t_2 \) be terms in \( T \). We define \( c_1 t_1 < c_2 t_2 \) if either \( t_1 < t_2 \), or \( t_1 = t_2 \) and \( c_1 < c_2 \). Clearly \( \leq \) is a linear extension of \( \preceq \) in \( \mathcal{M} \). By Proposition 5.4, \( \leq \) is a well-ordering. Let us call the partial orderings and total orderings on \( T \) and \( \mathcal{M} \) defined above the lexicographic extensions of \( \preceq \) and \( \leq \). Unless otherwise stated, given a reduction group \( G \), the lexicographic extensions will be used to define the orderings on \( T \) and \( \mathcal{M} \). Collectively, the partial and total orderings on \( T \) and \( \mathcal{M} \) form a reduction ordering set.

6. Reductions in Integral Group Rings

In this section we will define the notion of a reduction and eventually the notion of a Gröbner basis of a submodule of \( \mathbb{Z}[G]^k \) where \( G \) is a reduction group. Assume that we have a group reduction ordering on \( G \). Then a reduction ordering set can be defined using the lexicographic extensions. Let us denote the partial orderings and the total orderings on \( T \) and \( \mathcal{M} \) by \( \preceq \) and \( \leq \), respectively.

Every nonzero \( f \) in \( \mathbb{Z}[G]^k \) is of the form \( c_1 t_1 + c_2 t_2 + \cdots + c_r t_r \), where \( r \) is a positive integer, \( c_1, c_2, \ldots, c_r \) are nonzero integers and \( t_1 > t_2 > \cdots > t_r \) are term vectors. This expression will be called the standard form of \( f \). We define \( \text{lc}(f) = c_1 \) to be the leading coefficient of \( f \), \( \text{lt}(f) = t_1 \) to be the leading term vector of \( f \) and \( \text{lm}(f) = c_1 t_1 \) to be the leading monomial vector of \( f \).

**Definition 6.1.** Let \( g \) be a nonzero polynomial array in \( \mathbb{Z}[G]^k \) and \( x \) be an element in \( G \). We say that \( x \) is aligned with \( g \) if \( \text{lt}(g) \preceq \text{lt}(g)x \). Let \( m = ax \) be a monomial. We say that \( m \) is aligned with \( g \) if \( x \) is.

If \( g \) is a nonzero polynomial array in \( \mathbb{Z}[G]^k \), \( x \) is in \( G \) such that \( x \) is aligned with \( g \), and \( t \) is a term vector such that \( t < \text{lt}(g) \), then \( tx < \text{lt}(g)x \). In particular, all term vectors which appear in the standard form of \( gx \) are no bigger than \( \text{lt}(g)x \) with respect to \( \leq \). Clearly \( \text{lt}(gx) = \text{lt}(g)x \), \( \text{lm}(gx) = \text{lm}(g)x \), and \( \text{lc}(gx) = \text{lc}(g) \).

**Definition 6.2.** Let \( P \) be a set of nonzero polynomial arrays of \( \mathbb{Z}[G]^k \) with \( \text{lc}(f) > 0 \) for each \( f \) in \( P \). Define an elementary \( P \)-product to be a polynomial array of the form \( fm \) where \( f \) is in \( P \) and \( m \) is a monomial. An elementary product \( fm \) is a proper \( P \)-product if \( m \) is aligned with \( f \). If it is clear from the context what \( P \) is, then the reference to \( P \) will be dropped.

**Definition 6.3.** Let \( g_1 \) and \( g_2 \) be nonzero polynomial arrays in \( \mathbb{Z}[G]^k \) and \( P \) be a set of nonzero polynomial arrays in \( \mathbb{Z}[G]^k \) with \( \text{lc}(f) > 0 \) for each \( f \) in \( P \). We write \( g_1 \rightarrow_P g_2 \) if there exists \( f \) in \( P \) such that \( \text{lm}(g_1) \geq \text{lm}(f) \) and

\[
g_2 = g_1 - \left( \frac{\text{lc}(g_1)}{\text{lc}(f)} \right) fx,
\]

where \( x \) is the unique element in \( G \) with \( \text{lt}(f)x = \text{lt}(g_1) \). We say that \( g_1 \) is reduced to \( g_2 \) in one step by \( P \). Performing one such computation with \( g_1 \) is called a reduction.

There are two remarks we can make about this definition. First, if \( t \) is a term vector which appears in the standard form of \( f \), then \( t \leq \text{lt}(f) \). Since \( \text{lt}(f) \leq \text{lt}(g_1) = \text{lt}(f)x \),
that two polynomial arrays

Let

Theorem 6.1.

with

\lm(g_1) \geq \lm(f), \text{ either } \lc(g_1) = \lc(f) \text{ or } \lc(g_1) > \lc(f) \text{. In both cases } \left\lceil \frac{\lc(g_1)}{\lc(f)} \right\rceil \neq 0. \text{ So } g_1 \neq g_2 \text{. In fact, if } \\
\lt(g_1) = \lt(g_1), \text{ then } \lc(g_2) < \lc(g_1) \text{. Therefore } \lm(g_2) < \lm(g_1) \text{ whenever } g_1 \rightarrow_P g_2.

For a nonzero polynomial array \( g_1 \), define \( g_1 \rightarrow^* P g \) if \( g_1 = g \) or if there exists a succession of \( s \geq 1 \) reduction(s) \( g_1 \rightarrow_P g_2, g_2 \rightarrow_P g_3, \ldots, g_s \rightarrow_P g_{s+1} = g \). We say that \( g_1 \) can be reduced to \( g \) by \( P \). We cannot reduce the polynomial array 0 and we can only reduce a nonzero polynomial array if its leading monomial vector is preceded with respect to \( \preceq \) by the leading monomial vector of some polynomial array in \( P \). For each \( i, 1 \leq i \leq s \), let \( g_i = g_{i+1} - a_i h_i x_i \) be the equation of the reduction. So \( g_1 = g + a_1 h_1 x_1 + a_2 h_2 x_2 + \cdots + a_s h_s x_s \). In addition, \( \lt(h_1 x_1), \lt(h_2 x_2), \ldots, \lt(h_s x_s) \) are all \( \leq \lt(g_1) \). Clearly, \( g_1 - g \) is in the submodule generated by \( P \).

Given \( g_1 \) as above, we can only perform a finite number of reductions with \( g_1 \). Any sequence of reductions \( g_1 \rightarrow_P g_2, g_2 \rightarrow_P g_3, \ldots, \) must stop eventually. If not, then \( \lm(g_1) > \lm(g_2) > \cdots \) is an infinite sequence of strictly decreasing monomials, and this is not possible since \( \leq \) is a well-ordering on \( \mathcal{M} \).

**Definition 6.4.** Let \( \mathcal{I} \) be a nonzero submodule of \( \mathbb{Z}[G]^k \). A Gröbner basis of \( \mathcal{I} \) is a subset \( \mathcal{B} \) of \( \mathcal{I} \) such that if \( f \) is a nonzero polynomial array in \( \mathcal{I} \), then there exists \( g \in \mathcal{B} \) with \( \lm(g) \preceq \lm(f) \).

**Theorem 6.1.** Let \( \mathcal{B} \) be a Gröbner basis of \( \mathcal{I} \). For a polynomial array \( g \) in \( \mathbb{Z}[G]^k \), \( g \) is in \( \mathcal{I} \) iff \( g \rightarrow^* \mathcal{B} 0 \).

**Proof.** If \( g \rightarrow^* \mathcal{B} 0 \), then clearly \( g \) is in \( \mathcal{I} \). Now suppose that \( g \) is in \( \mathcal{I} \) and \( g \neq 0 \). In finitely many reductions, \( g \) can be reduced to a polynomial array \( g' \) in \( \mathcal{I} \) which cannot be reduced any more by \( \mathcal{B} \). If \( g' \neq 0 \), then since \( \mathcal{B} \) is a Gröbner basis, \( \lm(g') \geq \lm(f) \) for some \( f \) in \( \mathcal{B} \). Therefore \( g' \) can be reduced by \( \mathcal{B} \). This is a contradiction. So \( g \rightarrow^* \mathcal{B} 0 \). \( \square \)

**Corollary 6.2.** Let \( \mathcal{B} \) be a Gröbner basis of \( \mathcal{I} \). Then \( \mathcal{B} \) generates \( \mathcal{I} \) as a right \( \mathbb{Z}[G] \)-module.

### 7. Finding Gröbner Bases

Suppose \( P = \{ f_1, f_2, \ldots, f_s \} \) is a finite set of nonzero polynomial arrays in \( \mathbb{Z}[G]^k \). The problem of deciding membership in the submodule generated by \( P \) can now be reduced to first finding a Gröbner basis of the submodule and then performing reductions. The main goal of this section is to prove Theorem 7.3 which gives a sufficient condition for a set \( P \) to be a Gröbner basis of the submodule generated by itself. First we will establish some notation.

Let \( i \) and \( j \) be integers and \( 1 \leq i < j \leq s \). Denote \( \lt(f_i) \) by \( t_i \) and \( \lt(f_j) \) by \( t_j \). We say that two polynomial arrays \( f_i \) and \( f_j \) in \( P \) are aligned if there exists a term vector \( t' \) such that \( t' \succ t_i \) and \( t' \succ t_j \). Suppose \( f_i \) and \( f_j \) are aligned. Let \( t \) be \( \text{lcm}(t_i, t_j) \) which exists and is unique since \( \preceq \) has the unique LCM property. Let \( a_i \) be \( \text{lcm}(f_i, f_j) \), and \( p \) and \( q \) be integers such that \( pa_i + qa_j = \gcd(a_i, a_j) \). Let \( a \) be \( \text{lcm}(a_i, a_j) \). Define \( \alpha_{ij} \) to be \( pf_i \frac{a}{a_i} + qf_j \frac{a}{a_j} \) and \( \beta_{ij} \) to be \( \frac{a}{a_i} f_i - \frac{a}{a_j} f_j \). The \( \beta_{ij} \)'s are similar to the \( S \)-polynomials (see Becker and Weispfenning, 1991, Section 5.3) in the study of Gröbner bases in polynomial rings over fields. The \( \alpha_{ij} \)'s are needed because we are working with integer coefficients instead of over a field.
Definition 7.1. Suppose \( P \) is a set of nonzero polynomial arrays. Let \( g \) be a nonzero polynomial array and \( t \) be a term vector.

1. We say that \( g \) is lower\(_P\)-t if \( g \) can be written as \( h_1 m_1 + h_2 m_2 + \cdots + h_s m_s \) where for each \( i \) with \( 1 \leq i \leq s \), \( h_i m_i \) is a proper \( P \)-product and \( \text{lt}(h_i m_i) < t \).
2. We say that \( g \) is lower\(_P^\ast\)-t if \( g \) can be written as \( h_1 m_1 + h_2 m_2 + \cdots + h_s m_s \) where for each \( i \) with \( 1 \leq i \leq s \), \( h_i m_i \) is a proper \( P \)-product and \( \text{lt}(h_i m_i) \leq t \).
3. We say that \( g \) can be \( P \)-weakly reduced if \( g = h_1 m_1 + h_2 m_2 + \cdots + h_s m_s + h \) where \( h \) is lower\(_P\)-\( \text{lt}(g) \) and for each \( i \) with \( 1 \leq i \leq s \), \( h_i m_i \) is a proper \( P \)-product, \( \text{lt}(h_i m_i) = \text{lt}(g) \) and \( \text{lc}(h_i m_i) \) has the same sign as \( \text{lc}(g) \).

The reference to \( P \) will be dropped if it is clear from the context what \( P \) is.

Let \( a \) be a nonzero integer and let \( P \) and \( g \) be as in Definition 7.1. If \( g \) is lower\(_P\)-t, then \( ag \) is lower\(_P\)-t. Similarly, if \( g \) is lower\(_P^\ast\)-t, then so is \( ag \). It is also clear that if \( g \) can be \( P \)-weakly reduced, then \( ag \) can be \( P \)-weakly reduced.

Lemma 7.1. Let \( P \) be a set of nonzero polynomial arrays in \( \mathbb{Z}[G]^k \). If \( g \to \ast \, P \, 0 \) and \( \text{lc}(g) > 0 \), then \( g \) can be weakly reduced.

Proof. Suppose \( g \to \ast \, P \, 0 \). Let \( g_1 = g \) and the chain of reductions performed consists of \( g_2 = g_1 - h_1 m_1 \), \( g_3 = g_2 - h_2 m_2 \), \ldots, \( g_{s+1} = g_s - h_s m_s \) and \( g_s \neq 0 \). For all \( i \) with \( 1 \leq i \leq s \), \( h_i m_i \) is a proper product. Now \( g = h_1 m_1 + h_2 m_2 + \cdots + h_s m_s \).

Since \( \text{lt}(g_1) = \text{lt}(h_i m_i) \) for each \( i \) and \( \text{lm}(g_1) > \text{lm}(g_2) > \cdots > \text{lm}(g_s) \), it follows that \( \text{lt}(g_1) \geq \text{lt}(g_2) \geq \cdots \geq \text{lt}(g_s) \) and thus \( \text{lt}(g) \geq \text{lt}(h_i m_i) \) for each \( i \). We are given that \( \text{lc}(g) > 0 \).

Let \( i \) be maximal such that \( \text{lt}(g_i) = \text{lt}(g) \). Since \( \text{lm}(g_1) > \text{lm}(g_2) > \cdots > \text{lm}(g_s) \), \( \text{lc}(g_1) \gg \text{lc}(g_2) \gg \cdots \gg \text{lc}(g_i) \). But \( \text{lc}(g_1) = \text{lc}(g) > 0 \). So \( \text{lc}(g_i) > \text{lc}(g_2) > \cdots > \text{lc}(g) > 0 \) by the definition of \( \ll \). Therefore \( \text{lc}(h_1 m_1), \text{lc}(h_2 m_2), \ldots, \text{lc}(h_s m_s) \) are all positive. This shows that \( g \) can be weakly reduced. \( \square \)

Definition 7.2. A set of nonzero polynomial arrays \( P \) is said to be symmetrized if for any \( f \) in \( P \), \( \text{lc}(f) \) is positive, and for any \( x \) in \( G \), \( fx \) is lower\(_P\)-\( \text{lt}(fx) \).

Lemma 7.2. Let \( P \) be a symmetrized set of polynomial arrays in \( \mathbb{Z}[G]^k \). Let \( g \) be a nonzero polynomial array in \( \mathbb{Z}[G]^k \) and \( a \) be a nonzero integer. Suppose \( t \) is a term vector and \( x \) is in \( G \).

1. If \( g \) is lower-t and \( x \) is aligned with \( t \), then \( agx \) is lower-tx.
2. If \( g \) is lower\(_t^\ast\)-t and \( x \) is aligned with \( t \), then \( agx \) is lower\(_t^\ast\)tx.
3. If \( g \) can be weakly reduced and \( x \) is aligned with \( g \), then \( agx \) can be weakly reduced.

Proof.
1. Assume that \( g \) is lower-t for some term vector \( t \), then \( g = h_1 m_1 + h_2 m_2 + \cdots + h_s m_s \) where for each \( i \) with \( 1 \leq i \leq s \), \( h_i m_i \) is a proper product and \( \text{lt}(h_i m_i) < t \).

By the remark following Definition 7.1, we may assume that \( a = 1 \). Now \( g' = h_1 m_1 x + h_2 m_2 x + \cdots + h_s m_s x \). Let \( i \) be an integer with \( 1 \leq i \leq s \). It suffices to show that \( h_i m_i x \) is lower-tx. Since \( x \) is aligned with \( t \), \( t \geq tx \). Any term vector \( t' \) which appears in the standard form of \( h_i m_i x \) satisfies \( t' \leq \text{lt}(h_i m_i) < t \). So \( t' x < tx \) and \( \text{lt}(h_i m_i x) < tx \). Since \( P \) is symmetrized, \( h_i m_i x \) is lower\(_t^\ast\)-\( \text{lt}(h_i m_i x) \). Therefore \( h_i m_i x \) is lower-tx.
Theorem 7.3. Let $P$ be a symmetrized set of polynomial arrays in $\mathbb{Z}[G]^k$. If for every pair of aligned polynomial arrays $f_i, f_j$ in $P$, $\alpha_{ij}$ can be weakly reduced and $\beta_{ij}$ is lower-$\lcm(\lt(f_i), \lt(f_j))$, then $P$ is a Gröbner basis of the submodule generated by $P$.

Proof. Assume the hypothesis. Let $f$ be in the submodule generated by $P$. Then $f$ can be written as a sum of products of the form $hm$ where $h$ is in $P$ and $m$ is a monomial. If any of these products $hm$ is not a proper product, then since $P$ is symmetrized, $hm$ can be written as a sum of proper products. In other words, $f$ can be written as a sum of proper products. Let $f$ be written as $h_1m_1 + h_2m_2 + \cdots + h_km_k$, where $s \geq 1$, $h_1$ is lower-$\lt(h_1m_1)$ and for $1 \leq i \leq s$, $h_i$ is a proper product and $\lt(h_1m_1) = \lt(h_1m_1)$. We will call this expression of $f$ a proper sum for $f$. Each proper sum for $f$ defines two parameters. It defines a term vector parameter $\lt(h_1m_1)$ and an integer parameter $|\lcm(h_1m_1)| + |\lcm(h_2m_2)| + \cdots + |\lcm(h_km_k)|$. Among all proper sums for $f$, consider only those such that the term vector parameter is minimal in $\leq$, and among all such, pick one with a minimal integer parameter. We will show that in this expression of $f$, the leading coefficients of the $h_i$’s all have the same sign. If this is true, then $\lt(h_1) \leq \lt(h_1m_1) = \lt(f)$. It is not hard to see that either $\lcm(f) \gg \lcm(h_1)$ or $\lcm(f) = \lcm(h_1)$. So $\lcm(h_1) = \lt(f)$ and $P$ is a Gröbner basis.

Now we proceed to show that the leading coefficients of the $h_i$’s all have the same sign. If $s = 1$ then clearly we are done. Suppose $s \geq 2$. Without loss of generality, we assume that $\lcm(h_1m_1)$ and $\lcm(h_2m_2)$ have different signs. Since $m_1$ is aligned with $h_1$ and $m_2$ is aligned with $h_2$, $\lt(h_1) = \lt(h_1m_1)$ and $\lt(h_2) = \lt(h_2m_2)$. But $\lt(h_1m_1) = \lt(h_2m_2)$. Therefore the polynomial arrays $h_1$ and $h_2$ are aligned. Let $h_1$ be $f_i$ and $h_2$ be $f_j$ in $P$. Exchanging the indices of $h_1$ and $h_2$ if necessary, we assume that $i < j$. Let $t_i$ be $\lt(f_i)$, $t_j$ be $\lt(f_j)$ and $t$ be $\lcm(t_i, t_j)$. Let $x_i$ be $\frac{t_i}{t}$ and $x_j$ be $\frac{t_j}{t}$. Let $d$ be $\lcm(\lcm(f_i), \lcm(f_j))$. By the definitions of $\alpha_{ij}$ and $\beta_{ij}$, for some integers $p$ and $q$ such that $p \lcm(f_i) + q \lcm(f_j) = \gcd(\lcm(f_i), \lcm(f_j))$,

\[
\begin{bmatrix}
\frac{d}{\lt(f_i)} & \frac{\lcm(f_i)}{\lt(f_i)} \\
\frac{\lcm(f_j)}{\lt(f_j)} & \frac{d}{\lt(f_j)}
\end{bmatrix}
\begin{bmatrix}
f_ix_i \\
f_jx_j
\end{bmatrix}
= \begin{bmatrix}
\alpha_{ij} \\
\beta_{ij}
\end{bmatrix}.
\]

The 2-by-2 matrix in the above equation is unimodular. In fact,

\[
\begin{bmatrix}
f_ix_i \\
f_jx_j
\end{bmatrix}
= \begin{bmatrix}
\frac{d}{\lt(f_j)} & \frac{\lcm(f_j)}{\lt(f_j)} \\
\frac{\lcm(f_i)}{\lt(f_i)} & \frac{d}{\lt(f_i)}
\end{bmatrix}
\begin{bmatrix}
\alpha_{ij} \\
\beta_{ij}
\end{bmatrix}.
\]

Therefore any integer linear combination of $f_ix_i$ and $f_jx_j$ can be written as an integer linear combination of $\alpha_{ij}$ and $\beta_{ij}$.

Let $x$ be the group element in $G$ such that $tx = \lt(h_1m_1)$. Now $h_1m_1$ is an integer multiple of $f_ix_i$ and $h_2m_2$ is an integer multiple of $f_jx_j$. So $h_1m_1 + h_2m_2 = px +
q’βi,jx for some integers p’ and q’. Since t ≥ lt(h1m1) = tx, x is aligned with t. So
lt(αi,jx) = tx and lt(βi,jx) < tx. By Lemma 7.2, αi,jx can be weakly reduced and βi,jx is
lower-tx. So h1m1 + h2m2 = h1′m1′ + h2′m2′ + ⋯ + h′r m′r + h’ where s’ is nonnegative.
h’ is lower-tx and for 1 ≤ i ≤ s’, h′i m′i is a proper product with leading term vector tx
and leading coefficient the same sign as lc(h1m1 + h2m2). The case s’ = 0 occurs when
p’ = 0. As a result, h1′m1′ + h2′m2′ + ⋯ + h′r m′r + h3m3 + h4m4 + ⋯ + hsm + (h + h’)
is another proper sum for f with the term vector parameter smaller than or equal to
the term vector parameter in the proper sum h1m1 + h2m2 + ⋯ + hsm + h. We have a
contradiction if the term vector parameter in the new proper sum is strictly smaller than
the term vector parameter in the old proper sum. So let us assume that the term vector
parameters in the two proper sums are equal. In the new proper sum for f, the integer
parameter

\[
|\text{lcm}(h_{i1}m_{i1})| + \cdots + |\text{lcm}(h_{ir}m_{ir})| + |\text{lcm}(h_3m_3)| + |\text{lcm}(h_4m_4)| + \cdots + |\text{lcm}(h_sm_s)|
\]

\[
< |\text{lcm}(h_1m_1)| + |\text{lcm}(h_2m_2)| + \cdots + |\text{lcm}(h_sm_s)|.
\]

This implies that the integer parameter is not minimal in the proper sum h1m1 + h2m2 + ⋯ + hsm + h. We have arrived at a contradiction. Therefore all the leading coefficients
of h1m1, h2m2, ⋯, hsm are of the same sign. This finishes the proof. □

8. Symmetrization

One of the hypotheses in Theorem 7.3 is that P is a symmetrized set of polynomial
arrays. The group G is not necessarily finite. In order to show that P is symmetrized,
there are potentially infinitely many elementary products fx we need to check to make
sure that they are lower**-lt(fx). In this section, we will see that if we have a reduction
ordering set on \( \mathbb{Z}[G]^k \), then there are only a finite number of elementary products we
need to check.

Assume that we have a reduction ordering set on \( \mathbb{Z}[G]^k \). Let \( S \) be a subset of \( T \). For
each \( s \) in \( S \), define \( U_G(s, S) \) to be \( \text{min}(\{sx \mid x \in G \text{ and } sx \geq tx \forall t \in S\}) \) and \( V_G(s, S) \) to
be \( \{x \in G \mid sx \in \mathcal{U}_G(s, S)\} \). The subscript \( G \) will be dropped if it is clear what the group
is from the context. Using characterization (2) in Theorem 5.1, we can see that the sets
\( U_G(s, S) \) and \( V_G(s, S) \) are finite. By Proposition 5.3, if \( s \) is in \( S \) and if there exists \( x \in G \)
such that \( sx \geq tx \) for every \( t \) in \( S \), then there exists an element \( sx' \in \mathcal{U}_G(s, S) \) such that
\( sx \geq sx' \). Let \( f \) be a nonzero polynomial array in \( \mathbb{Z}[G]^k \). Define \( T_f \) to be the set of term
vectors which appear in the standard form of \( f \). Let \( \text{Symm}_G(f) \) be \( \{fx \mid x \in \mathcal{V}_G(s, T_f) \}
for some \( s \in T_f \} \). Again, if we know what the group \( G \) is from the context, the subscript
\( G \) will be dropped.

THEOREM 8.1. Assume that \( P \) is a set of nonzero polynomial arrays. If for every \( f \) in
\( P \), every element \( fx \) in \( \text{Symm}(f) \) is lower**-lt(fx), then \( P \) is a symmetrized set.

PROOF. We will prove by contradiction. Assume that there exists a polynomial array
\( f \) in \( P \) and an element \( x \) in \( G \) such that \( fx \) is not lower**-lt(fx). Let us call such an
\( fx \) a counterexample. Since \( \leq \) is a well-ordering on \( T \), we can pick \( f \) and \( x \) such that
among the counterexamples, \( \text{lt}(fx) \) is minimal in \( \leq \). Let the standard form of \( f \) be
c1t1 + c2t2 + ⋯ + cr tr. Therefore \( T_f = \{t_1, t_2, \ldots, tr\} \). Assume that \( \text{lt}(fx) = t_1x \). So
$t_i x \geq t_j x$ for $1 \leq j \leq r$. By the definition of $U(t_i, T_f)$ and Proposition 5.3, there exists an element $x'$ in $G$ such that $t_i x' \leq t_j x$ and $t_i x'$ is contained in $U(t_i, T_f)$. So $f x'$ is in $\text{Symm}(f)$ and is lower$^*$-$lt(f x')$. We can write $f x'$ as $h_1 m_1 + h_2 m_2 + \cdots + h_s m_s$ such that for $1 \leq j \leq s$, $h_j m_j$ is a proper product and $lt(h_j m_j) \leq lt(f x') = t_i x'$. Now

$$f x = (h_1 m_1 + h_2 m_2 + \cdots + h_s m_s) \frac{t_i x}{t_i x'}$$

$$= h_1 m_1 \frac{t_i x}{t_i x'} + h_2 m_2 \frac{t_i x}{t_i x'} + \cdots + h_s m_s \frac{t_i x}{t_i x'}.$$

It suffices to show that for $1 \leq j \leq s$, $h_j m_j \frac{t_i x}{t_i x'}$ is lower$^*$-$t_i x$.

Let $j$ be an integer with $1 \leq j \leq s$. Every term vector in $T_{h_j m_j}$ is no bigger than $t_i x'$ since $lt(h_j m_j) \leq t_i x'$. Let the monomial $m_j$ be $c x_j$, where $c$ is a nonzero integer and $x_j$ is a group element. In the case when $lt(h_j m_j) = t_i x'$, since $h_j m_j$ is a proper product, $t_i x' = lt(h_j m_j) = lt(h_j) x_j \geq lt(h_j)$. We have $lt(h_j) \leq t_i x' \leq t_i = lt(h_j) x_j \frac{t_i x}{t_i x'}$. So $h_j m_j \frac{t_i x}{t_i x'}$ is a proper product with leading term vector $t_i x$. Therefore $h_j m_j \frac{t_i x}{t_i x'}$ is lower$^*$-$t_i x$).

In the case when $lt(h_j m_j) < t_i x'$, every term vector in $T_{h_j m_j}$ is smaller than $t_i x'$ in $\leq$. So $lt(h_j m_j \frac{t_i x}{t_i x'}) < t_i x$. Using the minimality of $lt(f x') = t_i x$, we conclude that $h_j m_j \frac{t_i x}{t_i x'}$ is a proper product with leading term vector $t_i x$. Therefore $h_j m_j \frac{t_i x}{t_i x'}$ is lower$^*$-$t_i x$. Therefore $f x$ is lower$^*$-$lt(f x')$ and this is a contradiction. □

By Theorem 8.1, to show that $P$ is a symmetrized set, it suffices to show that for each $f$ in $P$, each polynomial array in $\text{Symm}(f)$ can be reduced to $0$. The set $\text{Symm}(f)$ is finite because $|V(s, T_f)| = |U(s, T_f)|$ is finite for every $s$ in $T_f$ and $T_f$ is finite. Moreover,

$$|\text{Symm}(f)| = \sum_{s \in T_f} |V(s, T_f)|.$$

To prove this, it suffices to show that the sets $V(s, T_f)$ where $s$ is in $T_f$ are disjoint. This is clear because if $x$ is contained in both $V(s, T_f)$ and $V(s', T_f)$, then $sx \geq s'x$ and $s'x \geq sx$. Therefore $sx = s'x$ and $s = s'$.

Now the remaining question is: how do we find $\text{Symm}(f)$ for a given nonzero $f$ in $\mathbb{Z}[G]^k$? Usually, the set $\text{Symm}(f)$ depends on the choice of the partial ordering $\leq$ and the total ordering $\leq$ on $G$. In the remaining part of this section, we will look at $\text{Symm}(f)$ for $f$ in $\mathbb{Z}[G]^1$ when $G$ is cyclic or finite. Term vectors in $\mathbb{Z}[G]^1$ are viewed here as group elements. Some routine calculation is skipped.

**Examples 8.1.**

1. Let $G$ be the infinite cyclic group $\mathbb{Z}$ generated by $a$ with the partial ordering given by $a^0 < a^1 < a^2 < \cdots$ and $a^0 < a^{-1} < a^{-2} < \cdots$, and the total ordering given by $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \cdots$. Any nonzero element $f$ of $\mathbb{Z}[G]$ can be written uniquely as a sum $c_1 a^{\lambda_1} + c_2 a^{\lambda_2} + \cdots + c_s a^{\lambda_s}$, where $s \geq 1$, $c_1, c_2, \ldots, c_s$ are nonzero integers and $\lambda_1 > \lambda_2 > \cdots > \lambda_s$ are integers in the usual ordering. If $s = 1$, then for each $x$ in $G$, $f x = c_1 a^{\lambda}$ for some integer $\lambda$. In this case, $U(a^\lambda, T_f) = \min \{ t_j \mid j \in \mathbb{Z} \} = \{ a^0 \}$ and $V(a^\lambda, T_f) = \{ a^{-\lambda} \}$. Therefore, $\text{Symm}(f)$ has only the monomial vector $c_1 a^{\lambda}$ in it. Now assume that $s \geq 2$. Let $\lambda$ be an integer. Any polynomial array $f a^{\lambda}$ is of the form $c_1 a^{\lambda_1+\lambda} + c_2 a^{\lambda_2+\lambda} + \cdots + c_s a^{\lambda_s+\lambda}$. It is not hard to see that for $2 \leq i < s - 1$, $a^{\lambda_i+\lambda} < a^{\lambda_i+\lambda}$ or $a^{\lambda_i+\lambda} < a^{\lambda_i+\lambda}$ in the total ordering on $G$. Therefore the sets $U(a^{\lambda_i}, T_f)$ and $V(a^{\lambda_i}, T_f)$ are empty if
2 \leq i \leq s - 1. We need to find \( V(a^{\lambda_1}, T_f) \) and \( V(a^{\lambda_s}, T_f) \). In \( G \), \( a^{\lambda_1+\lambda} \geq a^{\lambda_1+\lambda} \) iff \( \lambda_1 + \lambda + \lambda_3 + \lambda \geq 1 \), which is true as long as \( \lambda \geq \lfloor 1-\frac{2}{3}\lambda_3-\lambda_2 \rfloor \). Let \( \mu_1 = \lfloor 1-\frac{2}{3}\lambda_3-\lambda_2 \rfloor \). Therefore \( U(a^{\lambda_1}, T_f) = \min\{\{a^{\lambda_1+\mu_1} \mid \lambda \geq \mu_1\}\} \). Routine calculation shows that this set is just \( \{a^{\lambda_1+\mu_1}\} \). So \( V(a^{\lambda_1}, T_f) = \{a^{\mu_1}\} \). On the other hand, \( a^{\lambda_1+\lambda} \leq a^{\lambda_1+\lambda} \) iff \( \lambda_1 + \lambda + \lambda_3 + \lambda \leq 0 \), which is true as long as \( \lambda \leq \lfloor -\frac{2}{3}\lambda_3-\lambda_2 \rfloor \). Let \( \mu_s = \lfloor -\frac{2}{3}\lambda_3-\lambda_2 \rfloor \). Therefore \( U(a^{\lambda_s}, T_f) = \min\{\{a^{\lambda_s+\mu_s} \mid \lambda \leq \mu_s\}\} \). Routine calculation shows that this set is just \( \{a^{\lambda_s+\mu_s}\} \). So \( V(a^{\lambda_s}, T_f) = \{a^{\mu_s}\} \).

Summarizing the result for \( s \geq 2 \), we have \( \operatorname{Symm}(f) = \{fa^{\mu_1}, fa^{\mu_s}\} \).

2. Let \( G \) be again the infinite cyclic group generated by \( a \) with the same partial ordering as above. This time we take the total ordering \( a^0 < a^1 < a^2 < \cdots < a^{-1} < a^{-2} < \cdots \). Let \( f \) be a nonzero polynomial array with standard form \( c_1 a^\lambda + c_2 a^{\lambda_2} + \cdots + c_r a^{\lambda_r} \) as in above. For \( s = 1, f = c_1 a^\lambda \) and \( \operatorname{Symm}(f) = \{c_1 a^0\} \). It is not hard to see that if \( 2 \leq i \leq s - 1 \), then \( U(a^{\lambda_i}, T_f) \) is empty. Performing an analysis similar as in the first example, we get \( U(a^{\lambda_1}, T_f) = \{a^{\lambda_1-\lambda_2}\} \) and \( U(a^{\lambda_s}, T_f) = \{a^{-1}\} \). Therefore \( \operatorname{Symm}(f) = \{fa^{-\lambda_i}, fixa^{\lambda_1-\lambda_2}\} \).

3. Let us take a finite cyclic group as our third example. Suppose \( G \) is generated by \( a \) and \( G \) has order \( \geq 1 \) with partial ordering \( a^0 < a^1 < a^2 < \cdots < a^{m-1} \) and total ordering \( a^0 < a^1 < a^2 < \cdots < a^{m-1} \). Let \( f = c_1 a^\lambda + c_2 a^{\lambda_2} + \cdots + c_r a^{\lambda_r} \), where \( s \) is a positive integer, \( c_1, c_2, \ldots, c_r \) are nonzero integers and \( m > \lambda_3 > \lambda_2 > \cdots > \lambda_s \geq 0 \). If \( s = 1 \), then again \( \operatorname{Symm}(f) = \{c_1 a^0\} \). Suppose \( s \geq 2 \). It is tedious but routine to check that \( U(a^{\lambda_1}, T_f) = \{a^{\lambda_1-\lambda_2}\} \) and \( U(a^{\lambda_s}, T_f) = \{a^{\lambda_1-\lambda_2-\cdots-\lambda_{s-1}-m}\} \) for \( 2 \leq i \leq s \).

4. In the case when \( G \) is a finite group and \( G \) has the discrete poset structure, for a given nonzero \( f \) in \( \mathbb{Z}[G] \), \( |\operatorname{Symm}(f)| = |G| \) exactly. Since the sets \( V(t, T_f) \) are disjoint for \( t \) in \( T_f \), \( |\operatorname{Symm}(f)| \leq |G| \). Moreover, for every \( x \) in \( G \), \( x \) is in \( V(t, T_f) \) for some \( t \) in \( T_f \). So \( |\operatorname{Symm}(f)| \geq |G| \). We have \( \operatorname{Symm}(f) = \{fx \mid x \in G\} \). In general, if \( G \) is a finite group, then theoretically speaking, we can compute the polynomial arrays \( fx \) for every \( x \) in \( G \) and find \( \operatorname{Symm}(f) \) from these polynomial arrays by comparing their leading term vectors in \( \preceq \).

Now we would like to find an algorithm to compute \( \operatorname{Symm}(f) \) for \( f \) in \( \mathbb{Z}[G]^k \) when \( G \) is polycyclic-by-finite. Assume that the standard total ordering defined in Section 5 is used for the total ordering on \( G \). We begin with the case \( k = 1 \).

Assume that \( G \) is polycyclic-by-finite and that \( G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \cdots \triangleright G_{e+1} = 1 \) is a subnormal series of \( G \). For each \( i \) with \( 1 \leq i \leq e \), let \( H_i \) be the quotient \( G_i / G_{i+1} \) and \( \pi_i \) be the projection map from \( G_i \) to \( H_i \). Assume that we have, for each \( i \), a partial ordering \( \preceq_i \) and a group reduction ordering \( \triangleleft_i \) in \( H_i \). Let \( Y_i \) be a set of coset representatives for \( G_{i+1} \) in \( G_i \). Then we can define the standard partial ordering \( \preceq \) and the standard total ordering \( \preceq \) for \( G \) with respect to these parameters. By Theorem 5.5, \( \preceq \) is a group reduction ordering with respect to \( \preceq \).

Let \( U_G(s, S), V_G(s, S) \) and \( \operatorname{Symm}_G(f) \) be defined as before. Our object is to find \( U_G(s, S) \) for any finite set \( S \) in \( T \) and element \( s \) in \( S \). It is clear from the definition that if this can be done, then we can find \( \operatorname{Symm}_G(f) \) for every \( f \) in \( \mathbb{Z}[G] \).

For \( 1 \leq i \leq e \), denote by \( T_i \) the set of all term vectors in \( \mathbb{Z}[H_i] \). Denote by \( T' \) the set of all term vectors in \( \mathbb{Z}[G_2] \) and by \( T \) the set of all term vectors in \( \mathbb{Z}[G] \).

Assume that for each \( i \) such that \( 1 \leq i \leq e \) we know how to find \( U_{H_i}(s, S) \) where \( S \) is a finite subset of \( T_i \) and \( s \) is in \( S \). We would like to devise a scheme to find \( U_G(s, S) \) for \( s \) in a finite subset \( S \) of \( T \). To simplify the notation, let us denote \( H_1 \) by \( H \) and \( \pi_1 \) by \( \pi \). By induction on \( e \), assume that we know how to find \( U_G(s', S') \) for every
finite subset $S'$ of $T'$ and every element $s'$ in $S'$. Let $S$ be the set \{x_1y_1, x_2y_2, \ldots, x_r y_r\}

where $x_1, x_2, \ldots, x_r$ are elements in $G_2$ and $y_1, y_2, \ldots, y_r$ are coset representatives in $Y_1$. Consider $U_G(x_1y_1, S) = \min(x_1y_1xy \mid x \in G_2, y \in Y_1$ and $x_1y_1xy > x_1y_1xyxyj, 2 \leq j \leq r)$. By the definition of the standard total ordering, $x_1y_1xy > x_2y_2xy$ if either $\pi(y_1y) > _H \pi(y_2y)$ in $H$, or $y_1 = y_2$ and $x_1x_2y_2 > x_1x_2y_3$ in $G_2$. Let $S_1$ be the set \{\pi(y_1y), \pi(y_2y), \ldots, \pi(y_ry)\} and $S'$ be the set \{x \mid x \in S\}. We can find $U_H(\pi(y_1y), S_1)$ and so we can find the set \{\pi(y) \in V_2(\pi(y_1y), S_1)\}, which we will call $V_1$. By the induction hypothesis we can find $U_{G_2}(x_1, S')$ and so we can find $V_{G_2}(x_1, S')$, which we will call $V'$.

**Proposition 8.2.** \[U_G(x_1y_1, S) = \{x_1y_1xy \mid y \in V_1$ and $x_2y_2 \in V'\} and $V_G(x_1y_1, S) = \{xy \mid y \in V_1 and x_2y_2 \in V'\}.$

**Proof.** If $x_1y_1xy$ is in $U_G(x_1y_1, S)$, we want to show that $y$ is in $V_1$ and $x_2y_2 \in V'$. Since $x_1y_1xy$ is in $U_G(x_1y_1, S)$, $x_1y_1xy \geq x_1y_1xy$ and $\pi(y_1y) \geq \pi(y_2y)$ for every $i$ with $1 \leq i \leq r$. So there exists $y'$ in $V_1$ such that $\pi(y_1y') \leq \pi(y_1y)$. Now for each $i$ such that $1 \leq i \leq r$, it can be seen that $x_1y_1xy = x_1x_2y_2y' \geq x_1x_2y_2y' = x_1y_1xy$. In addition, $x_1y_1xy = x_1x_2y_2y' \leq x_1x_2y_2y' = x_1y_1xy$. Therefore, using the definition of $U_G(x_1y_1, S)$ and the fact that $x_1y_1xy$ is in $U_G(x_1y_1, S)$, we conclude that $x_1y_1xy' = x_1y_1xy$ and $y = y'$. Also, since $x_1y_1xy$ is in $U_G(x_1y_1, S)$, $x_1x_2y_2 \geq x_1x_2y_2$ whenever $1 \leq i \leq r$ and $y_1 = y_2$. In other words, $x_1x_2y_2 \leq x_1x_2y_2$ whenever $x_1$ is in $S$. Therefore there exists $(x'y')y_1$ in $V'$ such that $x_1(x'y')y_1 \leq x_1x_2y_2$. Again, we can be seen that $x_1y_1xy' = x_1(x'y')y_1y_1y_2 \geq x_1(x'y')y_1y_2 = x_1y_1xy$ for $1 \leq i \leq r$. Since $x_1y_1xy' \leq x_1y_1xy$, $x_1y_1xy' = x_1y_1xy$ and $x = x'$. Therefore $x_2y_2$ is in $V'$.

Conversely, if $y$ is in $V_1$ and $x_2y_2$ is in $V'$, we want to show that $x_1y_1xy$ is in $U_G(x_1y_1, S)$. It can be seen that $x_1y_1xy = x_1x_2y_2y_2 = x_1x_2y_2y_2 = x_1y_1xy$ for each $i$ with $1 \leq i \leq r$. So there exists $x'y'$ in $V_G(x_1y_1, S)$ with $x_1y_1xy' \geq x_1y_1xy$. Therefore $x_1(x'y')y_1y_2 \geq x_1x_2y_2y_2y_2 = x_1y_1xy$. Using the first part of this proof, since $x_1y_1xy' \in U_G(x_1y_1, S)$, we see that $y'$ is in $V_1 and (x'y')y_1$ is in $V'$. Since $y$ is in $V_1$ and $\pi(y_1y') \leq \pi(y_1y), \pi(y_1y') = \pi(y_1y)$ and $y' = y$. Since $x_2y_2$ is in $V'$ and $x_1(x'y')y_1y_2 \leq x_1x_2y_2y_2 = x_1x_2y_2 = x_1y_1xy$ and $x = x'$.

As a result, if we know how to find $U_{H_i}(s, S_i)$ for each finite subset $S_i$ of $H_i$ and each $s$ in $S_i$, then we know how to find $\text{Symm}_{G_i}(f)$. A recursive algorithm is given on the next page to find $\text{Symm}_{G_i}(f)$. This algorithm can also be written as a nonrecursive algorithm. There may be a lot of recursive calls to the algorithm Symmetrize but the depth of recursion never exceeds $e$.

Proposition 8.2 gives us a bound on the size of $\text{Symm}_{G_i}(f)$ for $f$ in $\mathbb{Z}[G]$. Suppose $1 \leq i \leq e$. Assume that for every $f$ in $\mathbb{Z}[H_i], |\text{Symm}_{H_i}(f)|$ is bounded by $M_i$, then for $f$ in $\mathbb{Z}[G], |\text{Symm}_{G_i}(f)|$ is bounded by $M_1 M_2 \cdots M_e$. For example, in the previous section, we saw that when $H$ is the infinite cyclic group generated by $a$ with group reduction ordering $\leq$ given by $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \cdots$ or by $a^0 < a^1 < a^2 < \cdots < a^{-1} < a^{-2} < \cdots$, then $|\text{Symm}_{H}(f)| = 2$ for all $f$ in $\mathbb{Z}[H]$. Therefore if $G$ has a subnormal series in which each quotient is infinite cyclic, then $|\text{Symm}_{G}(f)| \leq 2^e$ for every $f$ in $\mathbb{Z}[G]$, where $e$ is the length of the subnormal series.

Since we know how to find $\text{Symm}(f)$ for $f$ in $\mathbb{Z}[G]$, we will move on to the case
For any element \( x \) in \( G \), we know how to symmetrize polynomials in \( G \)-arrays in finite time by characterization (1) in Theorem 5.1. In finite time, the procedure will be a subnormal series of \( G \)-module of \( m \)-monomial vectors \( f(lm(P)) = \{f(x_1, f(x_2, \ldots, f(x_r)\}, \) then \( Symm(f) = \{f(x_1, f(x_2, \ldots, f(x_r)\}. Since we know how to symmetrize polynomial arrays in \( G \), we know how to symmetrize polynomial arrays in \( Z[G]^k \).

**Algorithm.**

**Procedure Symmetrize** \((G, f)\)

**Input:** \( G : \) A group together with a subnormal series

\[ G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \cdots \triangleright G_{e+1} = 1, \]

quotients \( H_i \) and coset representatives \( Y_i \) as described above;

\( f : \) A nonzero polynomial.

**Output:** Symm\(_G(f)\).

**Begin**

If \( G \) is the trivial group, return \( \{f\}; \)

(* Let us write \( H \) for \( H_1 \) and \( \pi \) for \( \pi_1 \). *)

Let \( f = f_1 y_1 + f_2 y_2 + \cdots + f_r y_r \), where

\( y_1, y_2, \ldots, y_r \) are distinct elements in \( Y_1 \),

\( f_1, f_2, \ldots, f_r \) are nonzero polynomials in \( Z[G_2]\);

Let \( S := \{\pi(y_1), \pi(y_2), \ldots, \pi(y_r)\}; \)

\( SP := 0; \)

For \( i := 1 \) to \( r \) do

\( V := V_H(\pi(y_i), S); \)

\( SP' := \text{Symmetrize}(G_2, f_i); \)

\( SP := SP \cup \{gy \mid g \in SP', \pi(y) \in V\}; \)

end;

Return \( SP; \)

end.

9. The Gröbner Basis Algorithm

At this point, we are ready to look at the full Gröbner basis algorithm in \( Z[G]^k \) when \( G \) is polycyclic-by-finite. The algorithm, listed on the next page, takes as input a set of nonzero polynomial arrays \( P \) and computes a Gröbner basis \( P' \) for the right \( Z[G]\)-submodule of \( Z[G]^k \) generated by \( P \), denoted by \( \langle P \rangle \). Let \( G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \cdots \triangleright G_{e+1} = 1 \) be a subnormal series of \( G \) such that each quotient is cyclic or finite. Assume that we have a reduction ordering set for \( Z[G]^k \).

In the algorithm, we reduce the polynomial arrays in \( Symm(f_i) \) and the polynomial arrays \( \alpha_{ij}, \beta_{ij} \) such that \( 1 \leq i < j \) and \( f_i, f_j \) are aligned. If the result of the reduction is not zero, we add it into the set \( P' \) and add appropriate polynomial arrays into the set \( S \).

If the procedure terminates, then by Theorem 7.3, \( P' \) is a Gröbner basis of the submodule \( \langle P' \rangle = \langle P \rangle \). The procedure will terminate in finite time. Suppose \( m_1 = \text{lm}(f_{|P|+1}), m_2 = \text{lm}(f_{|P|+2}) \ldots \). Since \( f_{|P|+2} \) cannot be reduced by \( f_{|P|+1}, m_2 \not\leq m_1 \). Similarly, \( f_{|P|+3} \) cannot be reduced by \( f_{|P|+1} \) and \( f_{|P|+2} \) and so \( m_3 \not\leq m_1, m_2 \). Inductively the sequence of monomial vectors \( m_1, m_2, \ldots \) satisfies the property that \( m_i \not\leq m_j \) whenever \( i < j \). This sequence is finite by characterization (1) in Theorem 5.1. In finite time, the procedure will
stop adding new polynomial arrays to \( P' \) and reduce the remaining polynomial arrays in \( S \) to 0.

**Algorithm.**

Procedure Gröbner_Basis\((P,B)\)

Input: \( P \): A set of nonzero polynomial arrays in \( \mathbb{Z}[G]^k \),
\[ P = \{f_1, f_2, \ldots, f_p\} \].

Output: \( P' \): A Gröbner basis of \( P \).

Begin
\[
P' := P;
\]
\[
S := \text{Symm}(f_1) \cup \text{Symm}(f_2) \cup \cdots \cup \text{Symm}(f_p) \cup \{ \alpha_{ij}, \beta_{ij} \mid 1 \leq i < j \leq |P|, f_i \text{ and } f_j \text{ are aligned} \};
\]

While \( S \neq \emptyset \) do begin
(* At all times \( P' = \{f_1, f_2, \ldots, f_p\}, p = |P'|, * )

Take a polynomial array \( g \) from \( S \);

While \( g \neq 0 \) and there exists \( f_i \) in \( P' \) with \( \text{lm}(f_i) \preceq \text{lm}(g) \) do begin
Let \( x \) be such that \( \text{lt}(g) = \text{lt}(f_i)x \);
\[
g := g - \lfloor \frac{\text{lc}(g)}{\text{lc}(f_i)} \rfloor f_i x;
\]
end; If \( g \neq 0 \) then begin
\[
f_{p+1} := g;
\]
\[
S := S \cup \text{Symm}(f_{p+1}) \cup \{ \alpha_{i,p+1}, \beta_{i,p+1} \mid 1 \leq i \leq p, f_i \text{ and } f_{p+1} \text{ are aligned} \};
\]
\[
P' := P' \cup \{f_{p+1}\};
\]
end
end
end.

Now we will see two examples of Gröbner basis calculations in integral group rings of polycyclic groups. Let \( G \) be the direct product of the infinite cyclic group with generator \( a_1 \) and the finite cyclic group of order 3 with generator \( a_2 \). Every element in \( G \) has a collected form \( a_2^k a_1^{\lambda_1} \) for some integers \( \lambda_1, \lambda_2 \). All polynomials shown here will be written in their standard forms. Let us find a Gröbner basis of the ideal in \( \mathbb{Z}[G] \) generated by \( f_1 = a_1^3 - 3a_1 + 2 \) and \( f_2 = a_2^2 - 3a_2 + 2 \). Let \( P \) be \( \{f_1, f_2\} \). It can be deduced that \( \text{Symm}(f_1) = \{f_1, f_1a_1^{-1}\} \) and \( \text{Symm}(f_2) = \{f_2, f_2a_2, f_2a_2^{-1}\} \). Now \( f_1a_1^{-3} = 2a_1^{-1} + a_1 - 3 \) cannot be reduced by \( P \). Therefore we define \( f_3 \) to be \( 2a_1^{-1} + a_1 - 3 \) and set \( P' \) to be \( P \cup \{f_3\} \). Next \( f_2a_2 = -3a_2^2 + 3a_2 + 1 \rightarrow p^* - 7a_2 + 7 \) and \(-7a_2 + 7 \) cannot be reduced any more by \( P' \). So we define \( f_4 \) to be \( 7a_2 - 7 \) and reset \( P' \) to include \( f_4 \). Now \( f_2a_2^2 = 2a_2^2 + a_2 - 3 \rightarrow p^* \). At this point, it can be seen that all the polynomials in the set \( S \) described in the algorithm can be reduced to zero. Therefore a Gröbner basis of \( \langle P \rangle \) is \( P' = \{a_1^3 - 3a_1 + 2, 2a_1^{-1} + a_1 - 3, a_2 - 3a_2 + 2, 7a_2 - 7\} \).

Suppose \( G \) is the group \( \langle a_1, a_2 \mid a_1a_2 = a_2^{-1}a_1 \rangle \). It can be seen that \( G \) is polycyclic. In fact, the presentation given here is essentially a consistent polycyclic presentation for \( G \). The collected form of an element in \( G \) is again \( a_2^k a_1^{\lambda_1} \) for some integers \( \lambda_1, \lambda_2 \). Suppose we would like to find the Gröbner basis of the right ideal in \( \mathbb{Z}[G] \) generated by \( P = \{a_1^3 - 3a_1 + 2, a_2^3 - 3a_2 + 2\} \). Now \( \text{Symm}(f_1) = \{f_1, f_1a_1^{-1}\} \) and \( \text{Symm}(f_2) = \{f_2, f_2a_2^{-1}\} \). As in the last example, \( f_1a_1^{-1} = 2a_1^{-1} + a_1 - 3 \) cannot be reduced by \( P \). So
we define \( f_1 \) to be \( 2a_1^{-1} + a_4 - 3 \) and set \( P' \) to be \( P \cup \{ f_3 \} \). Also \( f_2 a_2^{-1} = 2a_2^{-1} + a_2 - 3 \) cannot be reduced by \( P' \). So define \( f_4 \) to be \( 2a_2^{-1} + a_2 - 3 \) and set \( P' \) to be \( P \cup \{ f_4 \} \). Next we consider \( \beta_{12} = f_1 a_2^{-1} - f_2 a_2^{-1} \in P'\cdot a_2^{-1} \cdot a_1 - 4a_1 + 2a_2^{-2} - 6a_2 + 4 \), which we call \( f_3 \) and add to \( P' \). The symmetrized set of \( f_5 \) is \( \{ f_5 a_2^{-1}, f_5 a_2^{-1} a_1^{-1} \} \). Now \( f_5 a_2^{-1} \in P' \cdot a_2^{-1} + 3a_2^{-1} - 4 \). The symmetrized set of \( a_2^{-1} + 3a_2^{-1} - 4 \) is \( a_2^{-1} - 4a_2 + 3, 3a_2^{-1} - 3a_2 - 1 \). Define \( f_6 \) to be \( a_2^{-1} - 4a_2 + 3 \) and reset \( P' \) to include \( f_6 \). Reducing \( 3a_2^{-1} - 3a_2 - 1 \) we get \( 9a_2 - 9 \). Let \( f_r \) be \( 9a_2 - 9 \) and set \( P' \) to be \( P' \cup \{ f_r \} \). Reducing \( a_{37} = 5f_1 a_2^{-1} - f_3 a_1^{-1} \) we get \( 2a_1^{-1} + 5a_2^{-1} + 15a_2^{-1} \), which we call \( f_k \) and add to \( P' \). At this point, it is routine to check that all other polynomials in the set \( S \) can be reduced to 0 by \( P' \). Therefore \( P' \) is a Gröbner basis of the right ideal generated by \( \{ f_1, f_2 \} \).

10. Finding Group Generators

Let \( G \) be a reduction group. Given a free right \( \mathbb{Z}[G] \)-module \( \mathbb{Z}[G]^k \) and a submodule \( M_0 \) of \( \mathbb{Z}[G]^k \), we would like to find group generators for \( \mathbb{Z}[G]^k / M_0 \) as an abelian group. For each \( m \) in \( \mathbb{Z}[G]^k \), we would like to express \( m + M_0 \) in terms of these group generators. We need to define the notion of a full reduction.

Let \( P \) be a set of nonzero polynomial arrays in \( \mathbb{Z}[G]^k \) such that \( \text{lcm}(f) > 0 \) for all \( f \) in \( P \). Suppose \( g \) is a nonzero polynomial array with standard form \( c_1 t_1 + c_2 t_2 + \cdots + c_r t_r \). We say that \( g \) is fully reduced relative to \( P \) if for every \( f \) in \( P \) and for every \( i \) with \( 1 \leq i \leq r \), \( \text{lcm}(f) \not\subseteq c_i t_i \). We also say that the zero polynomial array is fully reduced. Let \( B \) be the Gröbner basis of a right submodule \( M_0 \) of \( \mathbb{Z}[G]^k \). For any polynomial array \( g \), in a finite number of reductions, we can find a unique polynomial array \( g' \), fully reduced relative to \( B \), such that \( g - g' \) is in \( M_0 \). We denote \( g' \) by \( \text{nf}(g) \) and call \( g' \) the normal form of \( g \) relative to \( B \).

**Proposition 10.1.** Let \( B \) be a Gröbner basis of a right submodule \( M_0 \) of \( \mathbb{Z}[G]^k \). Let \( T \) be the set of term vectors in \( \mathbb{Z}[G]^k \) which cannot be reduced by \( B \). Then \( \mathbb{Z}[G]^k / M_0 \) is generated as an abelian group by the set \( \{ b + M_0 \mid b \in T \} \).

**Proof.** Every element \( g + M_0 \) in \( \mathbb{Z}[G]^k / M_0 \) is equal to \( g' + M_0 \) for some fully reduced polynomial array \( g' \). Let \( c_1 t_1 + c_2 t_2 + \cdots + c_r t_r \) be the standard form of \( g' \). Then \( g + M_0 = c_1 (t_1 + M_0) + c_2 (t_2 + M_0) + \cdots + c_r (t_r + M_0) \). \( \square \)

Suppose \( G \) is a polycyclic group. Let \( G = G_1 \triangleright G_2 \triangleright \cdots \triangleright G_{n+1} = 1 \) be a polycyclic series of \( G \) with \( a_1, a_2, \ldots, a_n \) as the polycyclic generating sequence. Define \( Y_i \) to be \( \{ a_j^i \mid j \in \mathbb{Z} \} \) if \( G_i / G_{i+1} \) is finite and \( \{ a_j^0, a_j^1, \ldots, a_j^{m_j-1} \} \) if \( G_i / G_{i+1} \) is finite of order \( m_j \). Assume that for each \( i \) with \( 1 \leq i \leq n \), the partial ordering \( \preceq_i \) is as described in Examples 5.2 and the partial ordering \( \preceq \) on \( G \) is the standard partial ordering defined using the parameters above.

**Proposition 10.2.** Let \( B \) be a Gröbner basis of a right submodule \( M_0 \) in \( \mathbb{Z}[G]^k \). Let \( K \) be the set of all ordered pairs \( (i, j) \) such that \( 1 \leq i \leq k, 1 \leq j \leq n \) and \( H_j \) is infinite. Then \( \mathbb{Z}[G]^k / M_0 \) is finitely generated as an abelian group iff for each \( (i, j) \) in \( K \), there exists integers \( \alpha_{ij} \geq 0, \beta_{ij} \leq 0 \) such that \( c_i a_j^{\alpha_{ij}} \) and \( c_i a_j^{\beta_{ij}} \) are leading monomial vectors of some polynomial arrays in \( B \).
Proof. Suppose $\alpha^{(ij)}$ and $\beta^{(ij)}$ exist for every pair $(i, j)$ in $K$. Then the set $T$ defined in Proposition 10.1 is a subset of the set $T' = \{ e, a^n \lambda_i \ldots \lambda^2 a^{\lambda_1^1} \in T \mid \forall (i, j) \in K, \beta^{(ij)} < \lambda_j < \alpha^{(ij)} \}$. It is not hard to see that $T'$ is finite. Therefore $T$ is finite and $\mathbb{Z}[G]/M_0$ is finitely generated as an abelian group.

Now assume that $\mathbb{Z}[G]/M_0$ is finitely generated as an abelian group. Let $(i, j)$ be in $K$. Consider the subgroup of $\mathbb{Z}[G]/M_0$ generated by $e_i + M_0, e_i a_j + M_0, e_i a_j^2 + M_0, \ldots$. This subgroup is finitely generated. Therefore for a big enough positive integer $l$, $e_i a_j^l + M_0 = c_{l-1} e_i a_j^{l-1} + \cdots + c_1 e_i a_j + c_0 e_i + M_0$ for some integers $c_0, c_1, \ldots, c_{l-1}$. In other words, the polynomial array $c_i a_j^l = (c_{l-1} e_i a_j^{l-1} + \cdots + c_1 e_i a_j + c_0 e_i)$ is in $M_0$ and so $e_i a_j^l$ can be reduced by $B$. But if $m$ is a monomial vector such that $m \leq e_i a_j^l$, then $m = e_i a_j^l$ where $0 \leq l' < l$. In other words, there exists a polynomial array in $B$ with leading monomial vector $e_i a_j^{l'}$. This gives the existence of $\alpha^{(ij)}$. The existence of $\beta^{(ij)}$ can be proved similarly. \(\square\)

11. Constructing Polycyclic Presentations

In this section we will finish up with the polycyclic quotient algorithm. We will use the notation established in Section 4. By Theorem 4.4, $M/M_0$ and $N/N'$ are isomorphic as $\mathbb{Z}[G/N]$-modules. Since by the discussion in Section 4 we can find finitely many submodules generating $M_0$, it is possible to find a Gröbner basis $B$ for $M_0$. By Proposition 10.2, it is possible to decide whether $M/M_0$, and thus whether $N/N'$ is finitely generated as an abelian group. The quotient $G/N'$ is polycyclic if $N/N'$ is finitely generated as an abelian group. Therefore we can decide whether $G/N'$ is polycyclic. Suppose $G/N'$ is in fact polycyclic. Then the set $T$ in Proposition 10.1 is a finite set of group generators for $M/M_0$. Our goal is to construct a consistent polycyclic presentation for $G/N'$ and to find a defining pair for it.

Now $Y = \{ a_1, a_2, \ldots, a_n \}$. Let the elements in $T$ be denoted by $a_{n+1}, a_{n+2}, \ldots, a_l$ and $Y'$ be $Y \cup T$. To avoid confusion, the relations which we will construct for the presentation of $G/N'$ will be described as new relations. We think of $T$ as both a set of term vectors when we are working in $M$ and as a set of symbols when we try to construct the presentation for $G/N'$. Square brackets are used if we want elements in $T$ to be considered as term vectors. As term vectors, $[a_{n+1}] > [a_{n+2}] > \cdots > [a_l]$. What we will do basically is to transform, similar in principle to Tietze transformation, the presentation given in Theorem 4.3 to construct a consistent polycyclic presentation for $G/N'$. Let $n + 1 \leq i \leq l$. There will be a power relation involving the symbol $a_i$ iff there exists a positive integer $c$ such that the monomial vector $[ca_i]$ can be reduced by $B$. Since the monomial vector $[ca_i]$ cannot be reduced by $B$, we see that $c > 1$. Let $c_i$ be the smallest such $c$. If $\text{nf}([c_i a_i]) = [c'_{i+1} a_{i+1} + c'_{i+2} a_{i+2} + \cdots + c'_{l} a_l]$, then the new power relation involving $a_i$ is $a_i^{c_i} = a_{i+1}^{c'_{i+1}} \ldots a_{i+l}^{c'_{l+1}}$. We also obtain the new power relation with left side $a_i^{-1}$ similarly by finding the normal form of $[-a_i]$. Let $l'$ be the set of all $i$, $n + 1 \leq i \leq l$, such that a new power relation involving $a_i$ exists. The new conjugacy relations involving $a_{n+1}, a_{n+2}, \ldots, a_l$ are easy to get. For $n + 1 \leq i < j \leq l$, these new conjugacy relations are $a_i a_j = a_j a_i, a_i^{-1} a_j = a_j a_i^{-1}$ for $i \not\in l'$, $a_j a_i^{-1} = a_i^{-1} a_j$ for $j \not\in l'$ and $a_i^{-1} a_j^{-1} = a_j^{-1} a_i^{-1}$ for $i, j \not\in l'$. Let $1 \leq i \leq n$ and $n + 1 \leq j \leq l$. The next step is to obtain new conjugacy relations of the form $a_j e_{ij} a_i^{\pm 1}$. We will illustrate here how the rule with left side $a_i a_j$ can be obtained. The rest is similar. As an element in $M$, $[a_j a_i^{-1}]$ is the module element $[a_j]$
act on by \(a_i^{-1}\). If the normal form of \([a_ja_i^{-1}]\) is \([\epsilon'_{n+1}a_{n+1} + \epsilon'_{n+2}a_{n+2} + \cdots + \epsilon'_{l}a_{l}]\), then there is a new conjugacy relation \(a_i a_j = \epsilon'_i \cdots \epsilon'_{n+2} a_{n+1} \). Now we have to tackle the relations inherited from \(S\). Recall that \(|S| = s\). Let \(U_i = V_i^l\) be a relation in \(S\) where \(1 \leq i \leq s\). In Section 4, this relation is modified to \(U_i = [z_i^l]V_i\) where \(z_i^l\) is in \(Z\). Let the normal form of \([z_i^l]\) be \([\epsilon'_{n+1}a_{n+1} + \epsilon'_{n+2}a_{n+2} + \cdots + \epsilon'_{l}a_{l}]\). We obtain a new relation \(U_i = \epsilon'_i \cdots \epsilon'_{n+2} a_{n+1} V_i\). Let \(S'\) be the set consisting of all the new relations described above. Then a consistent polycyclic presentation for \(G/N'\) is \((Y' \mid S')\).

Finally, we have to update the defining pair. Let us call the new defining pair \((\sigma', \tau')\). It suffices to define \(\sigma'\) and \(\tau'\) on \(X\) and \(Y'\), respectively. In Section 4, we define \(\mu(x_i)\) to be \([z_i]^{-1}\sigma(x_i)\) for each \(x_i\) in \(X\). If the normal form of \([-z_i]\) is \([\epsilon'_{n+1}a_{n+1} + \epsilon'_{n+2}a_{n+2} + \cdots + \epsilon'_{l}a_{l}]\), then we define \(\sigma'(x_i)\) to be \(\epsilon'_i \cdots \epsilon'_{n+2} a_{n+1} \sigma(x_i)\). The homomorphism \(\nu\) defined in Section 4 will be used to construct the homomorphism \(\tau'\). For each \(y\) in \(Y'\), \(\tau'(y)\) is defined to be \(\nu(y)\). Then \(\sigma'\) and \(\tau'\) form a defining pair for \(G/N'\).

12. The Full Polycyclic Quotient Algorithm

Here the full extension algorithm is presented.

**Algorithm.**

**Procedure Extension Algorithm**\((\langle X \mid R \rangle, \langle Y \mid S \rangle, \sigma, \tau)\)

Input: \(\langle X \mid R \rangle\) : A finite presentation for a group \(G\),
\(\langle Y \mid S \rangle\) : A polycyclic presentation for a quotient \(G/N\),
\(\sigma, \tau\) : A defining pair for \(G/N\).

Output: flag : A flag to indicate whether \(G/N'\) is polycyclic,
\(\langle Y' \mid S' \rangle\) : A polycyclic presentation for \(G/N'\) if \(G/N'\) is polycyclic,
\(\sigma', \tau'\) : A defining pair for \(G/N'\) if \(G/N'\) is polycyclic,
\(B\) : A Gröbner basis for \(N/N'\) if \(G/N'\) is not polycyclic.

Begin

Construct a defining pair \((\mu, \nu)\) for \(G/N'\) as in Section 4;
Construct the free \(\mathbb{Z}[G/N]\)-module \(M\) and the submodule \(M_0\) as in Section 4;
Compute a Gröbner basis \(B\) for \(M_0\) using the algorithm in Section 9;
Determine whether \(G/N'\) is polycyclic using Proposition 10.2;
If \(G/N'\) is polycyclic, then begin
Set flag to true;
Find group generators for \(N/N'\) using Proposition 10.1;
Construct a polycyclic presentation \(\langle Y' \mid S' \rangle\) for \(G/N'\) as in Section 11;
Update the defining pair \((\mu, \nu)\) to \((\sigma', \tau')\) as in Section 11;
end
else Set flag to false;
end.

If the quotient \(G/N'\) is polycyclic, then the word problem in \(G/N'\) is reduced to computing collected forms of elements using the consistent polycyclic presentation for \(G/N'\). See Felsch (1976), Havas and Nicholson (1976), Vaughan-Lee (1990), Leedham-Green and Soicher (1990) and Section 13.3 for various collection algorithms. If \(G/N'\) is not polycyclic, then the word problem in \(G/N'\) can still be solved. Let us again use the notation in Section 3. Any element in \(G/N'\) can be represented by a collected word on \(Z\).
This collected word can be computed using the extension rewriting system. Suppose the collected word is \([m]W\), where \(W\) is a collected word on \(Y\) and \(m\) is a module element. Now \([m]W\) represents the identity in \(G/N\) iff \(W\) represents the identity in \(G/N\) and \([m]\) represents the identity in \(N/N\), which is equivalent to checking whether \(W\) is the empty word on \(Y\) and whether the normal form of \(m\) is 0. Since the normal form of \(m\) can be computed using the Gröbner basis \(B\), the word problem in \(G/N\) can be solved.

Now the polycyclic quotient algorithm can be described as follows.

**Algorithm.**

Procedure Polycyclic\_Quotient\_Algorithm\((\langle X \mid R \rangle, e)\)

**Input:** \(\langle X \mid R \rangle\): A finite presentation for a group \(G\),

\(e\): A positive integer.

**Output:**

- **flag**: A flag to indicate whether \(G/G^e\) is polycyclic,
- **\(\sigma, \tau\)**: A polycyclic presentation for \(G/G^e\) if \(G/G^e\) is polycyclic,
- **\(B\)**: A Gröbner basis for \(G^{(e-1)}/G^e\) if \(G/G^{(e-1)}\) is polycyclic and if \(G/G^e\) is not.

**Begin**

Compute a polycyclic presentation \(\langle Y \mid S \rangle\) and a defining pair \((\sigma, \tau)\) for \(G/G^d\);

Set \(d := 1\);

Set flag to be true;

While \(d < e\) and flag is true do begin

Use the Extension\_Algorithm with inputs \(\langle X \mid R \rangle, \langle Y \mid S \rangle, \sigma, \tau\) to obtain the outputs flag, \(\langle Z \mid T \rangle, \sigma', \tau'\) and \(B\);

(* Hence we obtain the quotient \(G/G^{(d+1)}\). *)

If flag is true then begin

Set \(\langle Y \mid S \rangle\) to be \(\langle Z \mid T \rangle\);

Set \((\sigma, \tau)\) to be \((\sigma', \tau')\);

end;

Set \(d := d + 1\);

end.

**13. Aspects of the Implementation**

A C program called \texttt{pcqa} has been developed to implement the polycyclic quotient algorithm described in this paper. The program is currently over fifteen thousand lines long. It uses the \texttt{gcc} compiler and its associated multiple precision integer package \texttt{gmp}. Both \texttt{gcc} and \texttt{gmp} can be obtained through anonymous ftp at various sites. The program \texttt{pcqa} itself can be obtained through anonymous ftp at \texttt{math.rutgers.edu}. The directory where the source code can be found is /pub/hlo. The file \texttt{README} in that directory has instructions for installing \texttt{pcqa}. The user’s manual (Lo, 1996b) is also available there.

**13.1. Data structures**

Throughout this section, let \(G\) be a polycyclic group and \(a_1, a_2, \ldots, a_n\) be a polycyclic generating sequence of \(G\). Let \(G_2\) be the subgroup of \(G\) generated by \(a_2, a_3, \ldots, a_n\).
Every element \( x \) in \( G \) can be expressed uniquely as a collected element \( a_1^{x_1} \cdots a_n^{x_n} a_i^{x_i} \). This representation gives a convenient data structure for group elements. We represent the element \( x \) as an array of \( n \) integers \( x_1, \ldots, x_n, x_1 \). Recall that a consistent polycyclic presentation consists of relations with left sides \( a_i a_j, a_i^{-1} a_j, a_i a_j^{-1}, a_i^m \) and right sides their respective collected forms. To describe a consistent polycyclic presentation for \( G \), it is sufficient to store the right sides of these relations in order. Internally, a consistent polycyclic presentation is described by an array of elements.

We have previously discussed two different group reduction orderings for the infinite cyclic group. In this implementation, for the infinite cyclic group generated by \( a \), we use the group reduction ordering \( a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \cdots \). For the finite cyclic group with order \( m \) and generator \( a \), we use the group reduction ordering \( a^0 < a^1 < a^2 < \cdots < a^{m-1} \). Let \( < \) be the total ordering of the integers defined by \( 0 < 1 < -1 < 2 < -2 \). For the polycyclic group \( G \), the group reduction ordering is defined in Section 5 as the standard total ordering relative to some parameters. Given two elements \( x \) and \( x' \) of \( G \). Let their respective collected forms be \( a_1^{x_1} \cdots a_n^{x_n} a_i^{x_i} \) and \( a_1^{x'_1} \cdots a_n^{x'_n} a_i^{x'_i} \). Suppose the standard total ordering is denoted by \( \leq \) and the standard partial ordering is denoted by \( \preceq \).

Then \( x < x' \) if there exists \( i \) with \( 1 \leq i \leq n \) such that \( x_1 = x'_1, x_2 = x'_2, \ldots, x_{i-1} = x'_{i-1} \) and \( x_i < x'_i \). Also \( x \preceq x' \) if for each \( i \) with \( 1 \leq i \leq n \), \( x_i \) and \( x'_i \) have the same sign and \( x_i \leq |x'_i| \).

Next the data structure of polynomials is described. Let \( f \) be a nonzero polynomial in \( \mathbb{Z}[G] \). Then \( f \) can be written in the form \( f_1 a_1^{p_1} + f_2 a_2^{p_2} + \cdots + f_r a_r^{p_r} \) where \( f_1, f_2, \ldots, f_r \) are polynomials in \( \mathbb{Z}[G_2] \) and \( p_1 \gg p_2 \gg \cdots \gg p_r \) as integers. This form of writing \( f \) gives an inductive way to store \( f \). Suppose we know how to store polynomials in \( \mathbb{Z}[G_2] \). To store \( f \) we need to store an array of size \( r \). Each entry in the array is a pair consisting of a polynomial in \( \mathbb{Z}[G_2] \) and an integer \( p \). For example, the polynomial \( f \) written above can be stored as an array of size \( r \) with entries \( (f_1, p_1), (f_2, p_2), \ldots, (f_r, p_r) \). In practice, a more complicated data structure called cell is used to build up a polynomial. A cell consists of five fields, gen, ind, power, hpt and a field defined by a union structure. This field can either be vpt or cpt depending on the value of ind. When the field ind is equal to 1, the fifth field is cpt. When ind is 0, the fifth field is vpt. Each cell can be viewed as an element of the form \( a_i^{x_i} \), in which case the field gen is \( i \) and the field power is \( p \). The fields hpt and vpt are pointers to other cells and the field cpt is a pointer to a multiple precision integer. The name used in the gmp package for a multiple precision integer is MP\_INT.

Let us look again at the polynomial \( f \) described earlier. Here we will describe a scheme to store \( f \) as a linked list of cells. Internally, \( f \) is represented as a pointer to a cell. This pointer is denoted by \( pf \) here. Let us assume, by induction on the number of polycyclic generators, that we have found internal representations for the polynomials \( f_1, f_2, \ldots, f_r \) as pointers \( pf_1, pf_2, \ldots, pf_r \), respectively. Using C notation, if a pointer points to a structure \( c \), then the value of the pointer is given by \&c. To denote the contents of a cell \( c \), we use a 5-tuple with the fields in the order gen, ind, power, hpt and vpt or cpt. Then a polynomial \( f \) can be stored using the following scheme.

If \( f = 0 \), then \( pf \) is the null pointer;
else if \( f = c \) for some integer \( c \neq 0 \), then
let \( pf \) point to a cell with fields \((n + 1, 1, 0, \text{null, \&(MP\_INT} c))\);
else if \( r = 1 \) and \( p_1 = 0 \), then take \( pf = pf_1 \);
else if \( r = 1 \) and \( p_1 \neq 0 \) and \( f_1 = c \) for some integer \( c \), then
let \( pf \) point to a cell with fields \( (1,1,p_1,\text{null},&(\text{MP\_INT}\ c)) \);
else if \( p_r = 0 \) then begin
let \( c_1,c_2,\ldots,c_{r-1} \) be a list of \( r - 1 \) cells;
for \( i := 1 \) to \( r - 2 \) do begin
if \( f_i \) is a constant \( c \), then the fields of \( c_i \) are \( (1,1,p_i,&c_{i+1},&(\text{MP\_INT}\ c)) \);
else the fields of \( c_i \) are \( (1,0,p_i,&c_{i+1},pf_i) \);
end;
if \( f_{r-1} \) is a constant \( c \), then the fields of \( c_{r-1} \) are \( (1,1,p_{r-1},pf_r,&(\text{MP\_INT}\ c)) \);
else the fields of \( c_{r-1} \) are \( (1,0,p_{r-1},pf_r,pf_{r-1}) \);
let \( pf \) point to \( c_1 \);
end;
else begin
let \( c_1,c_2,\ldots,c_r \) be a list of \( r \) cells;
for \( i := 1 \) to \( r - 1 \) do begin
if \( f_i \) is a constant \( c \), then the fields of \( c_i \) are \( (1,1,p_i,&c_{i+1},&(\text{MP\_INT}\ c)) \);
else the fields of \( c_i \) are \( (1,0,p_i,&c_{i+1},pf_i) \);
end;
if \( f_r \) is a constant \( c \), then the fields of \( c_r \) are \( (1,1,p_r,\text{null},&(\text{MP\_INT}\ c)) \);
else the fields of \( c_r \) are \( (1,0,p_r,\text{null},pf_r) \);
let \( pf \) point to \( c_1 \);
end.
end.

Figure 1 illustrates the internal representations of four polynomials \( f_1,f_2,f_3 \) and \( f_4 \). There, a cell is represented by a square box. The element \( a_i^p \) inside the box shows the fields gen and power. The field hpt is represented by a horizontal arrow to another cell. If hpt points to the word null, then it is a null pointer. A vertical arrow from a cell to another cell implies that the ind field is 0 and shows the field vpt. A vertical arrow from a cell to an integer implies that the ind field is 1 and shows the value of the field cpt. Here, the group \( G \) is the free abelian group of rank 3 on the generators \( a_1,a_2,a_3 \).

### 13.2. Implementing Symmetrized Sets

In the Gröbner basis algorithm described in Section 9, every time a new nonzero polynomial array \( g \) is found we add to \( S \) the polynomial arrays in the symmetrized set of \( g \) and the appropriate polynomial arrays \( \alpha_{ij} \) and \( \beta_{ij} \). In practice, however, it takes a lot of memory to store \( \alpha_{ij} \) and \( \beta_{ij} \) and so in general, these polynomial arrays are not computed until they are needed. Instead of storing the actual polynomial arrays, we store the pair of integers \( (i,j) \) only. For the same reason the polynomial arrays in Symm(\( g \)) are also not computed until they are needed. Let \( g' \) be in Symm(\( g \)). If \( g' \) is in \( P' \), the set we are building up to be the Gröbner basis in the algorithm in Section 9, then clearly \( g' \) can be reduced to zero by \( P' \). In this implementation, whenever a nonzero polynomial array \( g \) is obtained after reduction, in addition to adding \( g \) to \( P' \), all the polynomial arrays in Symm(\( g \)) are also added to \( P' \). Notice that \( g' = gx \) for some group element \( x \) in \( G \). Suppose \( g \) has been stored. To store \( g' \) all we need to do is to store the element \( x \) and to indicate that \( g' \) can be obtained by multiplying \( g \) by \( x \). In other words, we do not actually store \( g' \) but simply store a way to find \( g' \). This has the advantage of
saving space. Instead of storing several copies of essentially the same polynomial array which may have thousands of terms, we only store just one copy of the polynomial array and several group elements instead. One drawback is that every time we need to use the polynomial array \( g' \), we have to compute it by multiplying \( g \) with \( x \). Suppose \( e_i y \) is the leading term of \( g' \). In this implementation, in addition to storing \( x \), the element \( y x^{-1} \) is also stored. It turns out that by storing this extra element for each \( g' \), in most of the time when we need to use \( g' \), the extra operation it will cost is only one more multiplication of group elements.

13.3. A COLLECTION STRATEGY

One important part of the polycyclic quotient algorithm is in performing group element collection. The collection strategy in this implementation is based on an idea of Sims. Suppose \( a_1, a_2, \ldots, a_n \) is a polycyclic generating sequence. For \( 1 \leq i < n \), we call \( a_{i+1}, a_{i+2}, \ldots, a_n \) the lower generators of \( a_i \). To find the collected form of the element
first we find the automorphism induced by the conjugation of \((a_1^{-1})^{2i}\) on the lower generators \(a_2, a_3, \ldots, a_n\). Notice that finding this automorphism may require collection with lower generators and this may involve finding automorphisms induced by conjugation of lower generators on even lower generators. Sometimes we may have to find products and powers of automorphisms also. Clearly this is a highly recursive process. Once the automorphism induced by conjugation of \((a_1^{-1})^{2i}\) has been found, we can compute its action on the element \(a_n^{2i} \ldots a_2^{2i} a_1^{2i}\). Suppose the result is \(a_n^{2i} \ldots a_2^{2i} a_1^{2i+2n}\).

Now we can recursively perform collection on this element.

In this version of the polycyclic quotient algorithm, a lot of collection is required. When performing collections, it is possible to store partial results along the way. For example, if the automorphism due to conjugation by \(a_2^n\) where \(n\) is a positive integer has been computed, then this result can be stored and used later. Storing this automorphism may speed up future collections.

The collection strategy described here has been implemented in \texttt{pcqa}. At the current stage of development of \texttt{pcqa}, collection is not yet a problem compared to the difficulty of the Gröbner basis completion algorithm. The time spent on collection is just a small fraction of the time spent on reducing polynomials. Sims' strategy has been efficient in \texttt{pcqa} so far.

13.4. others

Here a few more issues of the implementation are briefly discussed. Readers are referred to the appropriate sources for detail.

In this implementation of the polycyclic quotient algorithm, some strategies to reduce the number of critical pairs to process are implemented. It turns out that while Buchberger's first criterion (see Becker and Weispfenning, 1991, Section 5.5) is not applicable, the reduction strategy proposed in Gebauer and Möller (1988) can be adapted. In addition, it is possible to delete some polynomial arrays entirely from the list of polynomial arrays \(P'\), where \(P'\) is as in the algorithm in Section 9. This turns out to be rather useful since with the way symmetrized sets is implemented, a lot of polynomial arrays are added to \(P'\) and many of them are redundant. See Section 4.10 of Lo (1996a) for details.

In the current implementation, a lot of interaction is required between the user and the program. While it takes extra time for users to type in commands, the interaction allows the user to set different options and have more control over the process of forming critical pairs and reducing polynomial arrays in the Gröbner basis algorithm listed in Section 9. In general this will reduce the run-time of the program. A full list of options available is given in Lo (1996b). In Appendix B of Lo (1996a), a general discussion of how these options may vary the performance of the Gröbner basis algorithm can be found.

In addition to computing the quotients of the form \(G/G^{(e)}\), \texttt{pcqa} also has the capability to compute other polycyclic quotients. The basic algorithm in \texttt{pcqa} is an extension algorithm. By adding in some extra relations, the extension algorithm can be modified to compute other quotients. For example, it can be used to compute finite solvable quotients and nilpotent quotients. Readers interested in how this can be done are referred to the sample runs in Lo (1996b). However, in general, \texttt{pcqa} does not compare well with the ANU Nilpotent Quotient Program (\texttt{NQ}) (see Nickel, 1994) when used to compute nilpotent quotients. It also does not compare well with the ANU Soluble Quotient Program.
Gröbner basis method in the integral group rings of polycyclic groups can also be used in other types of computation. For example, it can be used to find matrix representations of polycyclic groups (Ostheimer, 1996). Here, it is noted that \texttt{pcqa} can also be used for general purpose Gröbner basis computations in integral group rings of polycyclic groups. For details please see Lo (1996b).

14. Examples

In this section we look at a few results obtained from using the polycyclic quotient algorithm on some finite presentations. Due to the highly interactive nature of the program, an accurate timing and memory information cannot be obtained. To give the readers some idea about the time and memory requirement of the program, a rating which indicates the level of difficulty is given to each of the problems below. The experiments were performed on a Sun Sparc10 Model 41. A rating of \textit{easy} implies that the problem needs about two megabytes of memory and less than five minutes of interaction time to finish. A rating of \textit{normal} implies that the problem can be finished in thirty minutes using two megabytes of memory. A rating of \textit{hard} implies that the problem needs about five megabytes of memory and takes less than two hours of interacting with the program. A rating of \textit{very hard} implies that the problem needs twenty megabytes of memory and takes less than twenty-four hours of CPU time. Problems harder than these are described as \textit{extremely hard}. It is the author’s intention here to merely give a general feeling of the difficulty of each problem.

In this section, commutators are left-normed. In other words, we denote the commutator \(\ldots[[a_1, a_2], a_3], \ldots, a_n]\) by \([a_1, a_2, \ldots, a_n]\).

**Examples 14.1.**

1. For the first example we will look at the symmetric group \(S_4\) on four letters with the following presentation.

\[
G = \langle a, b, c \mid a^2, b^2, c^2, (ab)^3, (bc)^3, (ac)^2 \rangle
\]

The quotients for the derived series are \(\mathbb{Z}_2, \mathbb{Z}_3\) and \(\mathbb{Z}_2 \times \mathbb{Z}_2\). A consistent polycyclic presentation for \(G\) is constructed using about 1.7 megabytes of memory and three seconds of CPU time. The rating on this problem is easy.

2. For the second example we will look at the following presentation for the free nilpotent group of rank two, class three.

\[
G = \langle a, b \mid [b, a, a, a], [b, a, a, b], [b, a, b, b] \rangle
\]

The elements \([b, a, a, a], [b, a, a, b]\) and \([b, a, b, b]\) which appear as relators in the presentation are precisely the basic commutators of weight four defined in (Hall, 1959). The fact that \(G\) is the free nilpotent group of rank two, class three is proved by Groves and the result appears in Havas and Nicholson (1976).

We use \texttt{pcqa} to compute the derived series of \(G\). However, direct computation is not successful. The program \texttt{pcqa} returns the first two quotients \(\mathbb{Z}^2\) and \(\mathbb{Z}^3\) in almost no time, using about one megabyte of memory. However, showing the next quotient \(G''/G^{(3)}\) is trivial seems to be an extremely hard problem for \texttt{pcqa}. We need to simplify the presentation. Notice that the element \([b, a, a]\) in \(G\) is central in \(G\). Let \(N = \langle [b, a, a] \rangle\) be the subgroup generated by \([b, a, a]\) and \(\overline{G}\) be the quotient group.
A Polycyclic Quotient Algorithm  

Table 1.

<table>
<thead>
<tr>
<th>Group $G$</th>
<th>$G/G'$</th>
<th>$G'/G''$</th>
<th>$G''/G'''(3)$</th>
<th>$G'''/G''''(4)$</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>$Z^2$</td>
<td>$Z \times Z^2$</td>
<td>1</td>
<td>1</td>
<td>Easy</td>
</tr>
<tr>
<td>$G_2$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>not polycyclic</td>
<td>?</td>
<td>Easy</td>
</tr>
<tr>
<td>$G_3$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>not polycyclic</td>
<td>?</td>
<td>Easy</td>
</tr>
<tr>
<td>$G_4$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>not polycyclic</td>
<td>?</td>
<td>Easy</td>
</tr>
<tr>
<td>$G_5$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>not polycyclic</td>
<td>?</td>
<td>Easy</td>
</tr>
<tr>
<td>$G_6$</td>
<td>$Z^2$</td>
<td>$Z$</td>
<td>not polycyclic</td>
<td>?</td>
<td>Easy</td>
</tr>
<tr>
<td>$G_7$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>1</td>
<td>Normal</td>
</tr>
<tr>
<td>$G_8$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>1</td>
<td>Normal</td>
</tr>
<tr>
<td>$G_9$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>1</td>
<td>Normal</td>
</tr>
<tr>
<td>$G_{10}$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>not polycyclic</td>
<td>?</td>
<td>Normal</td>
</tr>
<tr>
<td>$G_{11}$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>$Z \times Z_2 \times Z_{32}$</td>
<td>?</td>
<td>Normal</td>
</tr>
<tr>
<td>$G_{12}$</td>
<td>$Z^2$</td>
<td>$Z^2$</td>
<td>1</td>
<td>1</td>
<td>Very hard</td>
</tr>
</tbody>
</table>

$G_1 = \langle a, b \mid [a^2, b, a], [a, b^2, a], [a, b, a^2], [b, a, b] \rangle$

$G_2 = \langle a, b, c \mid b^2, a, b^3, [b, a]^{a^2}, [b, a]^{a^2-1}, [b, a]^{a^2} \rangle$

$G_3 = \langle a, b, [b, a, b], [b, a, b]^{a^2}, [b, a]^{a^2}, [b, a]^{a^2-1}, [b, a]^{a^2} \rangle$

$G_4 = \langle a, b, [a, b, [b, a, a]], [b, a, b]^{a^2}, [a, b, a]^{a^2}, [b, a, b]^{a^2} \rangle$

$G_5 = \langle a, b, [a, b, a, b^2], [a, b, a, b^2] \rangle$

$G_6 = \langle a, b, [b, a, b], [b, a, b]^{a^2}, [b, a, b]^{a^2-1} \rangle$

$G_7 = \langle a, b, [a, b]^2, [a, b, [b, a, a]], [b, a, b]^{a^2-1}, [b, a, b]^{a^2} \rangle$

$G_8 = \langle a, b, [b, a, b], [b, a, b]^{a^2}, [b, a, b]^{a^2-1}, [b, a, b]^{a^2} \rangle$

$G_{10} = \langle a, b, [b, a, b], [b, a, b]^{a^2}, [b, a, b]^{a^2-1}, [b, a, b]^{a^2} \rangle$

$G_{11} = \langle a, b, [b, a, [b, a, a, a]], [b, a, b, a]^{a^2} \rangle$

$G_{12} = \langle a, b, [b, a, b], [b, a, b]^{a^2}, [b, a, b]^{a^2-1}, [b, a, b]^{a^2} \rangle$

$G / N$. Then $\overline{G}$ can be presented as $\overline{G} = \langle a, b \mid [b, a, a], [b, a, b, b] \rangle$. The first three quotients of the derived series of $\overline{G}$ are free abelian, $G'' \leq G''(3) \cap N$. By inspecting the consistent polycyclic presentation for $G / G''$, we can see that $\overline{G} / G''(3)$ is not in $G''$. Since $G'/G''$ is free abelian, $G'' / G'''(3) = 1$. So $G'''(3) = G''(3)$.

For the third example we look at the group $G = \langle a, b, c \mid abcab = c, bcabc = a, caba = b \rangle$ suggested by W. Nickel. The first quotient $G/G'$ is $Z_2 \times Z_4$ and the second quotient $G'/G''$ is $Z_2 \times Z_4$. The computation takes almost no CPU time. However, computing the third quotient $G''/G'''(3)$ turns out to be rather difficult for pca. By Tietze transformation, the above presentation can be simplified to $(a, b, c \mid a^2 = b^2 = c^2, abcab = c)$. With this presentation, computing the third quotient $G''/G'''(3)$ and the fourth quotient $G'''(3)/G''''(4)$ turns out to be much easier for pca. The quotient $G''/G'''(3)$ is $Z$ and the quotient $G'''(3)/G''''(4)$ is 1. The rating for this problem is normal.

A few more results are given in Table 1.

In Table 1, a question mark "?" means that the quotient is unknown. An unknown quotient appears when a previous quotient is known to be not polycyclic. It also appears if pca cannot come up with an answer in a reasonable amount of time. In our terminology, the problem is extremely hard. For example, the fourth quotient of $G_{11}$ currently cannot be computed. An entry in the column named Rating describes the level of difficulty in
computing the most difficult quotient which is not given by a question mark in that row. For example, the rating for computing the third quotient in $G_{11}$ is normal and this rating does not reflect the level of difficulty in computing the fourth quotient.

15. Conclusion

Experiments show that the polycyclic quotient algorithm outlined here works rather well when the presentation is relatively simple. The program works rather well when the given consistent polycyclic presentation has few power relations. Before computing with this program, if the abelian quotients turns out to have torsion, Reidemeister–Schreier methods (Neubuser, 1982; Schönert et al., 1996; Bosma and Cannon, 1996) may be used to get a presentation for a normal subgroup with finite index. This will reduce the number of power relations. Methods to simplify the presentation, for example, Tietze transformations (Havas et al., 1984; Schönert et al., 1996; Bosma and Cannon, 1996) and the Knuth–Bendix procedure (Sims, 1994, Chapter 2), may also be considered.

At this stage, the memory requirement of this program is not a pressing issue. The way symmetrized sets are implemented allows us to store many different polynomials as basically one polynomial. However, the run-time performance of the algorithm needs to be improved. Methods to reduce the number of critical pairs may be further improved. An intelligent strategy to select a critical pair to process will be desirable. Methods like LLL may be used for performing reduction.

Finally, a few comments on the complexity of this algorithm are needed. Complexity issues in Gröbner basis computations in polynomial rings over fields were discussed in the appendix of Becker and Weispfenning (1991). There, the authors cite a result from Möller and Mora (1984) that Gröbner basis computations have doubly exponential worst-case behavior in the number of variables. Huynh (1986) shows that for fixed number of indeterminates, the ideal membership problem is NP-hard. To complicate things, the Gröbner basis computations outlined here require performing collections in polycyclic groups. Many computational group theorists believe that such collections have exponential run-time behavior. However, we have reasons to be optimistic. The polynomial ring Gröbner basis algorithm works well on average, and the collection does not require much time with the experimentation so far. With more research, the polycyclic quotient algorithm can be made more efficient.

Acknowledgement

I would like to thank my PhD thesis advisor Professor Charles Sims for his generous support and assistance.

References

A Polycyclic Quotient Algorithm 97


Lo, E. (1996b). Using the Polycyclic Quotient Algorithm Program pcqa, Department of Mathematics, Rutgers University, New Jersey ftp@math.rutgers.edu.


Originally received 13 May 1996
Accepted 9 April 1997