# A POLYCYCLIC QUOTIENT ALGORITHM

BY EDDIE H. LO

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Mathematics

Written under the direction of

Professor Charles C. Sims

and approved by

New Brunswick, New Jersey

May, 1996

## ABSTRACT OF THE DISSERTATION

# A Polycyclic Quotient Algorithm

by Eddie H. Lo

Dissertation Director: Professor Charles C. Sims

Suppose $G$ is a group given by a finite presentation. Let $G^{(n)}$ denote the n-th term in the derived series of $G$. In 1981, Baumslag, Cannonito and Miller described an algorithm to determine whether the quotient $G/G^{(n)}$ is polycyclic and to find $G/G^{(n)}$ when it is polycyclic. However, in its original form, the algorithm is not practical for computer implementation. Here, a practical algorithm to determine whether $G/G^{(n)}$ is polycyclic and to find it when it is polycyclic is described. The algorithm involves a generalization of the Gröbner basis method of commutative ring theory to the integral group ring of a polycyclic group. An implementation of this algorithm in the language C has been developed and its efficiency is encouraging.

# Preface

This thesis is written in the area of computational group theory. As the name suggests, computational group theorists design and implement practical algorithms to compute in groups. My idea of a practical algorithm is one which is worth implementing. It will usually give results for inputs of small size and has a chance to terminate with results for large inputs. Even when the problem is too hard, the algorithm will produce some output which may help the user solve the problem. A practical algorithm is one which will work well on average, even though in the worst case, the run time may not be polynomial. In this respect, the general Gröbner basis algorithm, which has a doubly exponential worst case behaviour but a good average case behaviour, will be considered a practical algorithm.

# Acknowledgements

With great pleasure, I would like to thank the following people who have helped and supported me during my doctoral study.

First and most important of all, I would like to thank my thesis advisor Professor Charles C. Sims. Professor Sims has been my advisor ever since I came to Rutgers. In the first two years he was my academic advisor and gave me excellent advice on choosing courses and planning my path to the Ph.D. degree. He also taught a Computational Algebra course which led me to my research area Computational Group Theory. It was kind of him and fortunate of me that he took me as a Ph.D. student. As a thesis advisor he is knowledgeable, insightful and very helpful. I learn something every time we discuss mathematics. I enjoy every single one of our weekly meetings these last four years and really appreciate the precious time he spent on me. Without his help, this thesis would not have come to its existence. Professor Sims also has a positive influence on me as a person. I admire his positive, professional and meticulous attitude towards work. He has truly been a father figure to me during these six years of study. I cannot imagine I could have a better thesis advisor.

My sincere thanks go to Gretchen Ostheimer. For the past four years, she has been the perfect friend, colleague and office-mate to me. As a friend, she is a person whom I can talk to on many things. I treasure her every advice and every word of encouragement. As a colleague in mathematics, she is helpful and constructively critical. She is extremely patient in our conversation and always raises interesting mathematical issues during discussion. As an office-mate, she shows extreme patience and tolerance to her noisy neighbor. She is like a sister to me during these past four years. Her presence has made my study much more enjoyable. My sincere thanks also go to Emmi Schatz for her help on computers. She widens my scope on many other subjects as well.

I would like to thank Professor Richard Lyons, Professor Mike O'Nan and Professor Gilbert Baumslag from CUNY for carefully reading my thesis. Special thanks go to Professor Lyons for his help both in mathematics and in academic matters. I would like to thank Professor Michael Saks and Professor Jeff Tunnell for all their help over my six years at Rutgers, Dr. Laszlo Kovacs for his helpful comments on my thesis, and Werner Nickel for introducing me to the gnu compiler and other useful computer programs.

I would like to thank the Rutgers University Department of Mathematics for its financial support and the stimulating academic environment it has created.

Here, I would like to thank my family. My parents have been a constant source of care and encouragement, despite being five thousand miles away. Whenever I need to talk and I need help, they are always there for me. I would like to thank them for taking up the financial burden for these six years when I am studying at Rutgers. I wouldn't be able to finish my study if otherwise. My thanks also go to my sister Amy Lo whose sisterly love and care have always been present whether she is in Baltimore or in Hong Kong. She has been the perfect elderly sister for me, giving me valuable advice when I need them. Whenever we were together on the East coast, we had a lot of fun. My thanks also extend to my brother-in-law Alfred Wong.

I would like to thank the Chim's family which has been like my family here in New Jersey. Leung and I met in my earliest days at Rutgers. As a friend he is understanding, patient and wise. I enjoyed all the good times we have in the first three years and all the hard times we went through together in the last few years. In particular, Leung and Trang has given me emotional support for the last two years. I also thank baby Ivana for all the joy I felt every time I saw her.

Here I would like to thank a few special friends. Jin-yen Patrick Tai has been the perfect roommate. He is generous, efficient and understanding. Our cordial partnership is really strong. I would like to thank Ka-Leun Cheung for helping me get through the qualifying examinations, and John Shareshian, Siu-Hung Ng and Sau-Lai Chan for their help and the many mathematics discussions over the years. My sincere thanks go to

Stanley Chow, Hon and Melissa Sheung, Lok Sheung, Gregory Shiu, Eddie Li and my many friends in Hawaii who help my parents when I am there. Special thanks go to Stephen and Florence Lam for their friendship and their useful advice.

Last but not least, I would like thank my grandma Kim-Siu Yeung and my aunt Judy Au for their kind words of encouragement during my study.

# Dedication

This thesis is dedicated to my father Kwok-Lum Lo and my mother Ka-Min Lo.

# Table of Contents

# Chapter 1

# Introduction

The primary objective of this thesis is to describe an algorithm to compute polycyclic quotients of a finitely presented group. Algorithms of this kind are known as polycyclic quotient algorithms. The exact definition of this algorithm will appear in Chapter 3. In Chapter 2, we will look at some basic properties of polycyclic groups. The main tool used in this algorithm, a generalization of the Gröbner basis method to the integral group ring of a polycyclic group, will be examined in Chapter 4. In Chapter 5, we will see what we can deduce from a Gröbner basis. A C program implementing this algorithm has been written. Some results obtained from using this program will be listed in Chapter 6 and a few issues on the implementation will be discussed in the Appendix.

## 1.1 History

A *polycyclic group* is a group which has a subnormal series with only cyclic factors. Polycyclic groups were introduced by K.A. Hirsch in his series of papers [Hir38a], [Hir38b], [Hir46], [Hir52], [Hir54]. These groups form a class of groups in which many group-theoretic problems have algorithmic solutions. For example, the generalized word problem for polycyclic groups was solved by Mal'cev in [Mal58]. Readers interested in the algorithmic theory of polycyclic groups are referred to [BCRS91], [Seg90] and [Can91]. Many group-theoretic questions which are decidable in the class of polycyclic groups do not have algorithmic solutions in bigger classes. For example, it was shown in [Kha81] that the word problem is in general not solvable for finitely presented solvable groups.

After we know that a certain problem is solvable, the next step is to try to construct a practical algorithm to solve the problem. It turns out that all polycyclic groups have very nice presentations called consistent polycyclic presentations with which we can perform calculations in the groups. For example, if we have a consistent polycyclic presentation for a polycyclic group $G$, 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 [Sim94]. Getting a consistent polycyclic presentation is usually the first step in computing in polycyclic groups.

Given a finite presentation for a group $G$, the polycyclic quotient algorithm attempts to find the quotient $G/G^{(n)}$ where $G^{(n)}$ is the $n$-th term in the derived series of $G$. Mathematicians have been trying to compute solvable quotients for nearly a century. Determining the abelian quotient $G/G'$ was discussed in [Tie08] in 1908. In [CFL58] Chen, Fox and Lyndon showed that the isomorphism types of the quotients in the lower central series of a finitely presented group can be determined. Effective algoithms for determining $p$-quotients began with the work of Macdonald in [Mac73], [Mac74]. His techniques were extended in [SW73a], [SW73b], [Wam74], [BKW74], [New76a], [New76b] and [HN80]. In his thesis [Nic93] in 1993, Nickel developed and implemented an algorithm to compute nilpotent quotients of finitely presented groups. As for the quotients $G/G^{(n)}$, in 1981, Baumslag, Cannonito and Miller in their papers [BCM81a] and [BCM81b] developed an algorithm to decide whether these quotients are polycyclic and to find these quotients when they are polycyclic. However, their algorithm is not practical for computer implementation. More research on the polycyclic (or solvable) quotient algorithm can be found in [Wam77], [HJ82], [LG84] and [Ple87]. In [Sim90], Sims gave a practical algorithm to compute the metabelian quotient $G/G^{(2)}$. The method he used is a generalization of the Gröbner basis method to the integral group ring of a finitely generated abelian group. In [Hal54], P.Hall proved that the group ring of a polycyclic group is right Noetherian. This result suggests that some form of Gröbner basis computation may be possible in such a ring and a generalization of Sims' method may be applied to compute the quotients $G/G^{(n)}$ for bigger $n$. In Chapter 4, we will see that such an approach is in fact feasible. See [Wei92] for an example of a non-Noetherian ring $R$ in which each finitely generated ideal has a finite Gröbner basis.

In [Sim87] Sims 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 $G$.

# Chapter 2

# The Basics

In this chapter, we will establish our notation and take a brief look at rewriting systems. We will also look at polycyclic groups and mention some facts about them.

## 2.1 Free Monoids

Let $X$ be a set. Define $X^*$ to be $\{x_1x_2\ldots x_l \mid l \geq 0, x_1, x_2 \ldots, x_l \in X\}$, the set of strings formed from $X$. The empty string in $X^*$ is denoted by $\epsilon$. We can give $X^*$ a monoid structure by defining multiplication to be the usual concatenation of strings. The identity in this monoid is the empty string $\epsilon$. We call $X^*$ the *free monoid* on the set $X$.

Let $X$ be a set. A *congruence* on $X^*$ is an equivalence relation $\sim$ on $X^*$ such that for every $U$ and $V$ in $X^*$, $U \sim V$ implies that $UW \sim VW$ and $WU \sim WV$ for every $W$ in $X^*$. Let $\mathcal{R}$ be a set of ordered pairs of strings in $X^*$. The *congruence generated by* $\mathcal{R}$ is the smallest congruence in $X^*$ containing $\mathcal{R}$. It is not hard to see that $X^*/\!\sim$, the set of equivalence classes under an equivalence relation $\sim$, is again a monoid. If $W$ is a string in $X^*$ and $c$ is in $X^*/\!\sim$ such that $W$ is in the equivalence class $c$, then we say that $W$ represents $c$ and call $c$ the *image* of $W$ in $X^*/\!\sim$.

Let $X$ be a set. Define $X^- = \{x^{-1} \mid x \in X\}$ to be a set of formal inverses of elements in $X$. Define $X^\pm$ to be $X \cup X^-$ and $X^{\pm *}$ to be $(X^\pm)^*$. Let $x$ be an element in $X$. Notice that the strings $xx^{-1}$ and $x^{-1}x$ in $X^{\pm *}$ are not the same as the empty string $\epsilon$. Strings in $X^{\pm *}$ will be called *words on* $X$. Every word $U$ on $X$ is of the form $x_1^{\epsilon_1}x_2^{\epsilon_2}\ldots x_l^{\epsilon_l}$ for some integer $l \geq 0$, $x_1, x_2, \ldots, x_l$ in $X$ and $\epsilon_1, \epsilon_2, \ldots, \epsilon_l$ in $\{-1, 1\}$. We define $U^{-1}$ to be the string $x_l^{-\epsilon_l}\ldots x_2^{-\epsilon_2}x_1^{-\epsilon_1}$. If $\approx$ is the congruence generated by $\{(xx^{-1}, \epsilon) \mid x \in X^\pm\}$ in the monoid $X^{\pm *}$, then the monoid $X^{\pm *}/\!\approx$ is the free group $F(X)$ on $X$.

Let $\mathcal{R}$ be a set of ordered pairs of words on $X$. The set of equivalence classes under the congruence generated by $\approx \cup \mathcal{R}$ forms a group, which is exactly the group $F(X)/N_{\mathcal{R}}$, where $N_{\mathcal{R}}$ is the normal subgroup generated by the image of $U^{-1}V$ in $F(X)$ as $(U, V)$ ranges over $\mathcal{R}$. We say that a group $G$ has a *presentation* $\langle X \mid \mathcal{R} \rangle$ if there exists an epimorphism $\pi : F(X) \to G$ with kernel $N_{\mathcal{R}}$. The epimorphism $\pi$ is called the *projection map* of the presentation. We call an ordered pair $(U, V)$ in $\mathcal{R}$ a *relation* and $\mathcal{R}$ a *set of relations*. We usually write a relation $(U, V)$ as $U = V$ and call $U$ and $V$ the *left side* and the *right side* of the relation, respectively. Sometimes it is convenient to think of $\mathcal{R}$ as the set of words $\{U^{-1}V \mid (U, V) \in \mathcal{R}\}$. When there is no confusion caused, the notion of $\mathcal{R}$ as a set of ordered pairs and the notion of $\mathcal{R}$ as a set of words will be used interchangeably.

## 2.2 Rewriting Systems

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

**Examples:**

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 string in $(X \cup Y)^*$. 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 strings in $X^*$. Let $V$ be another string in $(X \cup Y)^*$. Suppose that the corresponding decomposition of $V$ is $C_0d_1C_1d_2 \ldots C_{s-1}d_sC_s$. 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 $(X \cup Y)^*$ and is called the *reverse-wreath-product ordering*

on $(X \cup Y)^*$ relative to $\leq_X$ and $\leq_Y$.

Suppose that $\leq$ is a rewriting ordering on $X^*$. Let $\mathcal{R}$ be a relation on $X^*$, $\sim$ be the congruence generated by $\mathcal{R}$ and $(P, Q)$ be in $\mathcal{R}$. If $P = Q$, then deleting $(P, Q)$ from $\mathcal{R}$ does not change $\sim$. If $P \neq Q$, then replacing $(P, Q)$ by $(Q, P)$ in $\mathcal{R}$ also does not change $\sim$. So we may assume that $P > Q$ for each $(P, Q)$ in $\mathcal{R}$. We call such $\mathcal{R}$ a *rewriting system* with respect to the rewriting ordering $\leq$. We call a pair $(P, Q)$ in $\mathcal{R}$ a *rewriting rule* in $\mathcal{R}$. We write $U \to V$ and say that $U$ *reduces to* $V$ if $U = APB$, $V = AQB$ and $(P, Q)$ is a rewriting rule in the rewriting system $\mathcal{R}$. A string $U$ is *irreducible* with respect to $\mathcal{R}$ if there does not exist any string $V$ with $U \to V$. We write $U \xrightarrow{*} V$ 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 $U_i \to U_{i+1}$ for $0 \leq i \leq l-1$. In words, we say that $U$ *can be reduced to* $V$. The rewriting system $\mathcal{R}$ is *confluent* if given any strings $U, V, W$ in $X^*$ such that $W \xrightarrow{*} U$ and $W \xrightarrow{*} V$, there exists a string $Q$ in $X^*$ such that $U \xrightarrow{*} Q$ and $V \xrightarrow{*} Q$.

Proofs for the next three propositions can be found in Chapter 2 of [Sim94].

**Proposition 2.2.1** *Let $\mathcal{R}$ be a confluent rewriting system on $X^*$. If $U$, $V$ and $W$ are strings in $X^*$ such that $U \xrightarrow{*} V$, $U \xrightarrow{*} W$ and both $V$ and $W$ are irreducible, then $V = W$.*

We say that *local confluence* holds at a string $W$ if for all strings $U, V$ in $X^*$ such that $W \to U$ and $W \to V$, there exists a string $Q$ such that $U \xrightarrow{*} Q$ and $V \xrightarrow{*} Q$.

**Proposition 2.2.2** *A rewriting system $\mathcal{R}$ is confluent if and only if local confluence holds at all strings.*

**Proposition 2.2.3** *Let $\mathcal{R}$ be a rewriting system. Suppose local confluence does not hold at a string $W$ but holds at all proper substrings of $W$. Then 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 substring.*

3. *$W$ can be written as $ABC$, where $A$, $B$, and $C$ are nonempty strings and both $AB$ and $BC$ are left sides in $\mathcal{R}$.*

If $W$ is the left side of a rule in $\mathcal{R}$ and contains the left side of another rule as a proper substring, then $W$ is called a *type 2 string* of the two rules. If $W$ can be written as $ABC$, where $A$, $B$, and $C$ are nonempty strings and both $AB$ and $BC$ are left sides of two rules in $\mathcal{R}$, then $W$ is called an *overlap* of the two rules. Now let $\mathcal{R}$ be a rewriting system. Assume that in $\mathcal{R}$ no two rewriting rules have the same left side. By Proposition 2.2 and Proposition 2.3, to check that $\mathcal{R}$ is confluent, we only need to make sure that local confluence holds at all type 2 strings and all overlaps. This fact will be used in Section 3.2.

## 2.3 Polycyclic Groups

Much of the exposition in this section is taken from Chapter 9 of [Sim94]. Readers are referred to that book for details and proofs.

As stated earlier, a group $G$ is polycyclic if there is a subnormal series $G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \ldots \triangleright G_{n+1} = 1$ such that for each $i$, $1 \leq i \leq n$, the quotient $G_i/G_{i+1}$ is cyclic. Such a series is called a *polycyclic series* of $G$.

**Examples:**

1. Finitely generated abelian groups are polycyclic.
2. Finitely generated nilpotent groups are polycyclic.
3. Finite solvable groups are polycyclic. This follows from Theorem 3.1 below.
4. Every solvable subgroup of $GL(n, \mathbb{Z})$ is polycyclic. In fact, every solvable group of automorphisms of a polycyclic group is polycyclic. For a proof see Chapter 2 of [Seg83].

The next theorem gives a list of several commonly used properties of polycyclic groups.

**Theorem 2.3.1**

1. *Subgroups and quotients of polycyclic groups are polycyclic.*
2. *If $N$ is a normal subgroup of a group $G$ and both $N$ and $G/N$ are polycyclic, then $G$ is polycyclic.*
3. *A group $G$ is polycyclic if and only if it is solvable and all of its subgroups are finitely generated.*

Let $G$ be a polycyclic group and let $G = G_1 \rhd G_2 \rhd G_3 \rhd \ldots \rhd G_{n+1} = 1$ be a polycyclic series of $G$. Let $1 \leq i \leq n$. Assume that $a_i$ is an element in $G_i$ such that the coset $a_iG_{i+1}$ generates the quotient $G_i/G_{i+1}$. The sequence $a_1, a_2, \ldots, a_n$ will be called a *polycyclic generating sequence* for $G$. It is clear that for every $x$ in $G$, $x$ can be written in the form $a_n^{\lambda_n} a_{n-1}^{\lambda_{n-1}} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$ where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are integers. Let $I$ be the set of integers $i$ such that $1 \leq i \leq n$ and $G_i/G_{i+1}$ is finite. For each $i$ in $I$, let $m_i$ be the order of $G_i/G_{i+1}$. In the expression $a_n^{\lambda_n} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$ for $x$, if we impose the condition that $0 \leq \lambda_i < m_i$ for each $i$ in $I$, then the sequence of integers $\lambda_n, \ldots, \lambda_2, \lambda_1$ is unique and we call the word $a_n^{\lambda_n} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$ the *collected form* of $x$. In particular, if $x$ is in $G_i$ where $1 \leq i \leq n$ and the collected form of $x$ is $a_n^{\lambda_n} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$, then $\lambda_1 = \lambda_2 = \ldots = \lambda_{i-1} = 0$. Notice that this definition of the collected form is different from the conventional definition. Usually, a collected form looks like $a_1^{\mu_1} a_2^{\mu_2} \ldots a_n^{\mu_n}$ for integers $\mu_1, \mu_2, \ldots, \mu_n$. The reason for this nonstandard terminology is that when working in right modules of integral group rings of polycyclic groups, $a_n^{\lambda_n} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$ is the form which works. Suppose that $x$ is an element in $G$ and $x$ is not the identity. We say that the *leading generator* of $x$ is $a_i$ if the collected form of $x$ is $a_n^{\lambda_n} \ldots a_{i+1}^{\lambda_{i+1}} a_i^{\lambda_i}$ and $\lambda_i \neq 0$.

We are now ready to define conjugacy relations and power relations. Suppose $i$ is in $I$. Then $a_i^{m_i}$ is contained in $G_{i+1}$. Let the collected form of $a_i^{m_i}$ be $W_i$ and that of $a_i^{-1}$ be $W_i^- a_i^{m_i-1}$. Either $W_i$ is the identity or the index of the leading generator of $W_i$ is bigger than $i$. We call the relations of the forms $a_i^{m_i} = W_i$ and $a_i^{-1} = W_i^- a_i^{m_i-1}$ *power relations.* Now let $1 \leq i < j \leq n$, then $a_j^{a_i}, (a_j^{-1})^{a_i}, a_j^{(a_i^{-1})}$ and $(a_j^{-1})^{a_i^{-1}}$ are all elements in $G_{i+1}$. Let $W_{ij}^{-+}, W_{ij}^{--}, W_{ij}^{++}, W_{ij}^{+-}$ be their respective collected forms. The index of the leading generator of each one of the above elements is at least $i+1$ if the element is not the identity. The *conjugacy relations* consist of the following relations for every $i, j$ with $1 \leq i < j \leq n$:

$$\begin{aligned} a_i a_j &= W_{ij}^{++} a_i, & & \\ a_i a_j^{-1} &= W_{ij}^{+-} a_i, & &\text{for } j \notin I, \\ a_i^{-1} a_j &= W_{ij}^{-+} a_i^{-1}, & &\text{for } i \notin I, \\ a_i^{-1} a_j^{-1} &= W_{ij}^{--} a_i^{-1}, & &\text{for } i, j \notin I. \end{aligned}$$

The conjugacy relations do not include those rules with $a_i^{-1}$ or $a_j^{-1}$, where $i, j$ are in $I$, appearing on the left side. It is enough to have the rule $a_i^{-1} = W_i^- a_i^{m_i - 1}$ among the power relations for each $i$ in $I$.

Notice that for each conjugacy relation and each power relation, the word on the right side is exactly the collected form of the element on the left side. The conjugacy and power relations constitute a group presentation for $G$. We call this presentation the *standard polycyclic presentation* relative to $a_1, a_2, \ldots, a_n$. In fact, the relations in a standard polycyclic presentation of $G$, together with the relations $a_i a_i^{-1} = e$ and $a_i^{-1} a_i = e$ for $1 \leq i \leq n$, give a confluent rewriting system using a combination of reverse-wreath-product orderings on $\{a_1, a_2, \ldots, a_n\}^{\pm *}$. Using this rewriting system and given a word on $\{a_1, a_2, \ldots, a_n\}$, we can find the collected form of the element represented by the word. The rewriting process is known as collection. Different strategies of collection can be found in [Fel76], [HN76], [VL90] and [LGS90].

A group presentation which consists of conjugacy and power relations is called a *polycyclic presentation*. Any polycyclic presentation defines a polycyclic group $G$. If a polycyclic presentation is the standard polycyclic presentation for the group it defines, then the presentation is called a *consistent polycyclic presentation*.

# Chapter 3

# Computing Polycyclic Quotients

The central theme of this thesis is to describe a practical polycyclic quotient algorithm. Let $G$ be a finitely presented group and let $e$ be a positive integer. The quotient $G/G^{(e)}$ is solvable but not necessarily polycyclic. The primary purpose of the polycyclic quotient algorithm is to decide whether $G/G^{(e)}$ is polycyclic, and if so, give a useful description of the quotient.

Now let us look at a slightly more general problem. Suppose that $G$ is finitely presented and $N$ is a normal subgroup of $G$ such that $G/N$ is polycyclic. Let $N'$ be the commutator subgroup $[N, N]$, which is normal in $G$. Assume that we have a description of the quotient $G/N$. We want to decide whether $G/N'$ is polycyclic, and if so, return a description of $G/N'$. This problem will be called the *extension problem*. We will describe an algorithm called the *extension algorithm* to solve the extension problem.

## 3.1 Describing Quotient Groups

Here we define what we mean by a description of a polycyclic quotient of a finitely presented group. Much of the exposition in this section is taken from Chapter 11 of [Sim94].

It should be mentioned here that throughout this thesis, the order of composition of functions is from right to left.

Let $\langle X \mid R\rangle$ be a finite presentation for $G$ and let $N$ be a normal subgroup of $G$. Assume that we have obtained a presentation $\langle Y \mid S\rangle$ for $G/N$. To describe $G/N$ effectively, we would like to somehow describe the normal subgroup $N$ also. Let $F(X)$ and $F(Y)$ be the free groups on $X$ and $Y$, respectively. Let $\pi_X : F(X) \to G$ and

$\pi_Y : F(Y) \to G/N$ be the quotient maps. Suppose that $\sigma : F(X) \to F(Y)$ 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 $\langle X \mid R \rangle$ and $\langle Y \mid S \rangle$. Furthermore, if $\tau : F(Y) \to F(X)$ 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 $\langle X \mid R \rangle$ and $\langle Y \mid S \rangle$. Given a defining pair, we can find a finite set of normal generators for $N$.

**Proposition 3.1.1** *Suppose $(\sigma, \tau)$ is a defining pair for $G/N$ relative to $\langle X \mid R \rangle$ and $\langle Y \mid S \rangle$. Let $Q$ be the set $\{x^{-1}(\tau \circ \sigma)(x) \mid x \in X\} \cup \{\tau(s) \mid s \in S\}$. The normal closure of $\pi_X(Q)$ in $G$ is $N$.*

**Proof:** Let $M$ be the normal closure of $\pi_X(Q)$ in $G$. First we will show that $M$ is a subgroup of $N$. Since $\overline{\sigma}$ is the projection map from $G$ to $G/N$, it suffices to show that the image of each element in $Q$ under $\overline{\sigma} \circ \pi_X = \pi_Y \circ \sigma$ is the identity in $G/N$. Let $x$ be in $X$. We have $(\pi_Y \circ \sigma)(x^{-1}(\tau \circ \sigma)(x)) = (\pi_Y \circ \sigma)(x^{-1})(\pi_Y \circ \sigma \circ \tau \circ \sigma)(x) = (\pi_Y \circ \sigma)(x^{-1})(\pi_Y \circ \sigma)(x) = 1$ since $\pi_Y \circ \sigma \circ \tau = \pi_Y$. If $s$ is in $S$, then $(\pi_Y \circ \sigma \circ \tau)(s) = \pi_Y(s) = 1$. Therefore $M \leq N$.

Now we will show that $N$ is a subgroup of $M$. Let $W$ be an element in $F(X)$ such that $\pi_X(W)$ is in $N$. We will see that $\pi_X(W)$ is in $M$. Suppose that $W = x_1 x_2 \ldots x_r$ where $x_1, x_2, \ldots, x_r$ are elements in $X^{\pm}$. For each $x$ in $X$, $\pi_X(x)$ is in $(\pi_X \circ \tau \circ \sigma)(x)M$. Therefore $\pi_X(W)$ is contained in $(\pi_X \circ \tau \circ \sigma)(W)M$. Since $\pi_X(W)$ is in $N$ and $M \leq N$, $(\overline{\sigma} \circ \pi_X \circ \tau \circ \sigma)(W) = 1$. As a result, $(\pi_Y \circ \sigma)(W) = (\pi_Y \circ \sigma \circ \tau \circ \sigma)(W) = 1$. Therefore $\sigma(W) = (s_1^{\pm 1})^{U_1}(s_2^{\pm 1})^{U_2} \ldots (s_l^{\pm 1})^{U_l}$ for a nonnegative integer $l$, elements $U_1, U_2, \ldots, U_l$ in $F(Y)$ and elements $s_1, s_2, \ldots, s_l$ in $S$. For $1 \leq i \leq l$, $(\pi_X \circ \tau)((s_i^{\pm 1})^{U_i})$ is contained in $M$. So $(\pi_X \circ \tau \circ \sigma)(W)$ is in $M$, $\pi_X(W)$ is in $M$ and $N$ is a subgroup of $M$. $\Box$

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 $G$.

**Definition:** Let $H$ be a group and let $K$ be a normal abelian subgroup of $H$. Then $K$ has a right $\mathbb{Z}[H]$-module structure where the action of $H$ on $K$ is by conjugation. Moreover, since $K$ acts trivially, we can view $K$ as a right $\mathbb{Z}[H/K]$-module. We call this $\mathbb{Z}[H/K]$-module structure of $K$ the *induced module structure* of $K$ from $H$.

Since $N/N'$ is a normal abelian subgroup of $G/N'$, we can give $N/N'$ the induced module structure from $G/N$. If $Q$ generates $N$ as a normal subgroup, then $QN'$ generates $N/N'$ as a right $\mathbb{Z}[G/N]$-module. This leads to the following corollary.

**Corollary 3.1.2** *$N/N'$ is generated as a right $\mathbb{Z}[G/N]$-module by $x^{-1}\tau(\sigma(x))N'$ where $x$ is in $X$ and $\tau(s)N'$ where $s$ is in $S$. If $S$ is finite, then $N/N'$ is finitely generated as a right $\mathbb{Z}[G/N]$-module.*

In the remaining part of this section, we will formally describe the extension algorithm and the polycyclic quotient algorithm. For the extension algorithm, its inputs consist of a finite presentation $\langle X \mid R\rangle$ of a group $G$, a consistent polycyclic presentation $\langle Y \mid S\rangle$ of a polycyclic quotient $G/N$ of $G$ and a defining pair $(\sigma, \tau)$ for $G/N$ relative to the presentations $\langle X \mid R\rangle$ and $\langle Y \mid S\rangle$. Given these inputs, the extension algorithm decides whether the quotient $G/N'$ is polycyclic, and if so, it returns a consistent polycyclic presentation $\langle Y' \mid S'\rangle$ for $G/N'$ and a defining pair $(\sigma', \tau')$ for the quotient $G/N'$ relative to the presentations $\langle X \mid R\rangle$ and $\langle Y' \mid S'\rangle$.

For the polycyclic quotient algorithm, its inputs consist of a finite presentation $\langle X \mid R\rangle$ for a group $G$ and a positive integer $e$. The algorithm decides whether the quotient $G/G^{(e)}$ is polycyclic, and if so, it returns a consistent polycyclic presentation $\langle Y \mid S\rangle$ for $G/G^{(e)}$ and a defining pair $(\sigma, \tau)$ for $G/G^{(e)}$ relative to $\langle X \mid R\rangle$ and $\langle Y \mid S\rangle$. Once we have the extension algorithm, the polycyclic quotient algorithm follows. It is not hard to find a consistent polycyclic presentation for $G/G'$ and a defining pair for it. Applying the extension algorithm to this set of data with $N = G'$, we can decide whether $G/G^{(2)}$ is polycyclic and if so, we can construct a consistent polycyclic presentation and a defining pair for $G/G^{(2)}$. In general, suppose that $G/G^{(e)}$ is polycyclic and we have a consistent polycyclic presentation and a defining pair for $G/G^{(e)}$. Using the extension algorithm with $N = G^{(e)}$, we can decide whether $G/G^{(e+1)}$ is polycyclic and if it is, we can find a consistent polycyclic presentation and a defining pair for $G/G^{(e+1)}$.

Suppose that we are given a defining homomorphism $\sigma$ for $G/N$ relative to a presentation $\langle X \mid R \rangle$ for $G$ and a consistent polycyclic presentation $\langle Y \mid S \rangle$ for $G/N$. It is possible to compute a homomorphism $\tau : F(Y) \to F(X)$ such that $(\sigma, \tau)$ is a defining pair for $G/N$. This computation is similar to finding Hermite normal forms for integer matrices but with collection in place of the usual row operations. However, such computation can be time-consuming and $\tau$ is usually constructed together with $\sigma$.

## 3.2 Extension Rewriting Systems

In this section we take a digression and look at a particular kind of rewriting system. Let $\langle Y \mid S \rangle$ be a consistent polycyclic presentation for a polycyclic group $G$. Suppose that $S$ 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 $Z = Y \cup M$. To avoid confusion between the multiplication in $K$ and the module action of $G$ on $M$, elements in $M$, when considered as elements in $K$, will be enclosed in square brackets. Within the brackets, multiplication denotes the module action. Outside the brackets, multiplication is the group operation 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 elements in $K$ represented by $[m]$ and $U$. Similarly, $[mU^{-1}]$ denotes the element in $K$ represented by the module element which is the result of $m$ acted on by the element in $G$ represented by $U^{-1}$. 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 $U$. Therefore a collected word on $Y$ is also 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:

$$
\begin{array}{llcll}
(1) & [m_1]\,[m_2] & = & [m_1+m_2], & m_1, m_2 \in M, \\
(2) & a[m] & = & [ma^{-1}]a, & m \in M, a \in Y^{\pm}, \\
(3) & U_i & = & [\mu_i]V_i, & 1 \leq i \leq s.
\end{array}
$$

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 $\phi(\pi_Z(a)) = \pi_Y(a)$ for every $a$ in $Y$. It can be seen that $\phi$ is an epimorphism with kernel equal to the subgroup generated by $\pi_Z(M)$. 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 $M$. In Chapter 4 we will see an example of such a well-ordering when $M$ is free of finite rank. Extend this well-ordering to a well-ordering 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 $S$ give a confluent rewriting system. Define $\leq$ on $Z^{\pm *}$ to be the reverse-wreath-product ordering relative to $\leq_Y$ and $\leq_M$. Let us consider the rewriting system $\mathcal{R}$ on $Z^{\pm *}$ with the following rules:

$$
\begin{array}{llcll}
(1) & [m_1]\,[m_2] & \to & [m_1+m_2], & m_1, m_2 \in M, \\
(2) & [m]^{-1} & \to & [-m], & m \in M, \\
(3) & aa^{-1} & \to & \epsilon, & a \in Y^{\pm}, \\
(4) & a[m] & \to & [ma^{-1}]a, & m \in M, a \in Y^{\pm}, \\
(5) & U_i & \to & [\mu_i]V_i, & 1 \leq i \leq s, \\
(6) & [0] & \to & \epsilon. &
\end{array}
$$

This rewriting system 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 $M$. By using the rules of type (2), we can rewrite any word $W$ on $Z$ to the form $[m_1]W_1[m_2]W_2 \ldots [m_r]W_r[m_{r+1}]$, where $r$ is a positive integer, $m_1, m_2, \ldots, m_{r+1}$ are in $M$ and $W_1, W_2, \ldots, W_r$ 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 $[m]W_1W_2 \ldots W_k$ with $m$ in

$M$, or simply $W_1W_2\dots W_k$ when $m=0$. In other words, in a finite number of steps, we can rewrite any word on $Z$ to a word $[m]W'$, where $m$ is in $M$ and $W'$ is a word on $Y$, or to simply a word $W'$ on $Y$. We are done if $W'$ is a collected word on $Y$. Otherwise we can use the rules of type (3) or (5). If we use a rule $U_i \to [\mu_i]V_i$, where $1 \le i \le s$, we introduce an element $[\mu_i]$. However, we can move such an element 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 simply a collected word $W''$ on $Y$. In either case, in a finite number of steps, we will obtain a collected word on $Z$ which cannot be rewritten any further. Also, since for each rewriting rule, the image under $\phi$ of the left side is equal to the image under $\phi$ of the right side, $\phi(W)=\phi(W'')$ is exactly the element in $G$ represented by $W''$.

We would like this rewriting system to be confluent. This rewriting system is confluent if local confluence holds at all strings $W$ of the three types described in Proposition 2.2.3. So $\mathcal{R}$ is confluent if local confluence holds for all type 2 strings and all overlaps. Our approach now is to form all possible type 2 strings and overlaps and check whether local confluence holds there. Let us consider overlaps first. We will work out a few of them just to get the flavor of reducing overlaps. The rest is very similar.

Consider an overlap formed by two rules of type (1) in $\mathcal{R}$. 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 $[m_1][m_2] \to [m_1+m_2]$ first or using the rule $[m_2][m_3] \to [m_2+m_3]$ first. To indicate the order of reduction, we underline the substring which we reduce first. Since $\underline{[m_1][m_2]}[m_3] \to [m_1+m_2][m_3] \to [m_1+m_2+m_3]$ and $[m_1]\underline{[m_2][m_3]} \to [m_1][m_2+m_3] \to [m_1+m_2+m_3]$, local confluence holds at the overlap $[m_1][m_2][m_3]$.

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 $\underline{a[m_1]}[m_2] \to [m_1a^{-1}]a[m_2] \to [m_1a^{-1}][m_2a^{-1}]a \to [m_1a^{-1}+m_2a^{-1}]a$ and $a\underline{[m_1][m_2]} \to a[m_1+m_2] \to [(m_1+m_2)a^{-1}]a$, local confluence holds at this overlap.

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 $Ua[m]$ and there are two ways to process this overlap. We can reduce it as $\underline{Ua}[m]$ which gives $[\mu_i + mV_i^{-1}]V_i$, or we can reduce it as $U\underline{a[m]}$ which yields the same result.

It turns out that there are only two kinds of overlaps at which local confluence may not hold. They are the overlaps formed by a rule of type (3) and a rule of type (5), and the overlaps formed by two rules of type (5). In both cases the overlaps are words on $Y$. Let $W$ be such an overlap. Processing $W$ in two different ways, we obtain two collected words on $Z$. Suppose that the words are $[m_1]W_1$ or $W_1$ when $m_1 = 0$, and $[m_2]W_2$ or $W_2$ when $m_2 = 0$. Let $m = m_1 - m_2$. Since $\phi(W_1) = \phi(W) = \phi(W_2)$ and both $W_1$ and $W_2$ are collected words on $Y$, $W_1 = W_2$. Therefore local confluence holds if $m = 0$ in $M$.

Let us consider type 2 strings now. A type 2 string can be formed by a rule of type (1) and the rule (6). For example, if $m_1$ is in $M$, then $\underline{[m_1][0]} \rightarrow [m_1 + 0] = [m_1]$ and $[m_1]\underline{[0]} \rightarrow [m_1]$. So local confluence holds at $[m_1][0]$. It is easy to see that local confluence holds at the type 2 strings $[0][m_2]$ and $a[0]$ also, where $m_2$ is in $M$ and $a$ is in $Y^{\pm}$. The nontrivial case occurs when a type 2 string is formed by a rule of type (3) and a rule of type (5), or by two rules of type (5). Suppose that the polycyclic generating sequence for $G$ is $a_1, a_2, \ldots, a_n$. Let the polycyclic series of $G$ corresponding to this polycyclic generating sequence be $G = G_1 \triangleright G_2 \triangleright \ldots \triangleright G_{n+1} = 1$. Denote by $I$ the set of integers $i$ such that $1 \leq i \leq n$ and $G_i/G_{i+1}$ is finite. For each $i$ in $I$, let $m_i$ be the order of $G_i/G_{i+1}$. Then for each $i$ in $I$, the words $a_i a_i^{-1}$ and $a_i^{-1}a_i$ are type 2 strings. We can rewrite the word $a_i a_i^{-1}$ as $a_i\underline{a_i^{-1}}$ and as $\underline{a_i a_i^{-1}}$. If in addition $m_i = 1$, then the word $a_i a_i^{-1}$ can also be rewritten as $\underline{a_i}a_i^{-1}$. As in the case of overlaps, by rewriting a type 2 string in two different ways, we will obtain an element $m$ in $M$ such that local confluence holds if $m = 0$.

We saw previously how to obtain the module elements $m$ from overlaps and from type 2 strings. There are only finitely many rules of types (3) and (5). So there are only finitely many such $m$'s. The right $\mathbb{Z}[H]$-submodule generated by the set of all such module elements will be called the *overlap module* for the extension rewriting system $\mathcal{R}$. From the above discussion, $\mathcal{R}$ is confluent if and only if the overlap module for $\mathcal{R}$

is 0. This gives the following proposition.

**Proposition 3.2.1** *If $\mathcal{R}$ is the extension rewriting system relative to a right $\mathbb{Z}[H]$-module $M$ and module elements $\mu_1, \mu_2, \ldots, \mu_s$, and $M_0$ is a submodule of $M$ containing the overlap module for $\mathcal{R}$, 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.*

## 3.3 Forming Module Generators

In this section we continue our discussion of the extension algorithm. We will use the same notation as in Section 1. Assume that $G$ has a finite presentation $\langle X \mid R \rangle$, $X = \{x_1, x_2, \ldots, x_r\}$ and $G/N$ is given by a consistent polycyclic presentation $\langle Y \mid S \rangle$. Let $S$ consist of the relations $U_1 = V_1, U_2 = V_2, \ldots, U_s = V_s$. Let $\pi_X : F(X) \to G$ and $\pi_Y : F(Y) \to G/N$ be 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 $Q = \{x^{-1}(\tau \circ \sigma)(x) \mid x \in X\} \cup \{\tau(U_i V_i^{-1}) \mid 1 \leq i \leq s\}$ under $\pi_X$. Let $Z' = \{z'_1, z'_2, \ldots, z'_s, z_1, z_2, \ldots, z_r\}$ be a set of symbols, $M$ be the free right $\mathbb{Z}[G/N]$-module on $Z'$ and $Z$ be the set $Y \cup M$. We would like to find a presentation $\langle Z \mid T \rangle$ for $G/N'$. The bracketing notation established in Section 2 will again be used in this section.

First we would like to construct homomorphisms $\mu : F(X) \to F(Z)$ and $\nu : F(Z) \to F(X)$ 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 $\tau(\sigma(x_i))x_i^{-1}$ for $1 \leq i \leq r$ and $\nu([z'_i])$ to be $\tau(U_i V_i^{-1})$ 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 $Y$. Now for every nonzero $m$ in $M$ such that $m$ is not of the form $\zeta U$, $m$ can be written as $\sum_{i=1}^{l} c_i \zeta_i U_i$ 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 $\Pi_{i=1}^{l}(\nu([\zeta_i U_i]))^{c_i}$. Now $\nu$ can be extended to a homomorphism from $F(Z)$ to $F(X)$. Since for $1 \leq i \leq r$, $\nu(\mu(x_i)) = \nu([z_i])^{-1}\nu(\sigma(x_i)) = x_i(\tau(\sigma(x_i)))^{-1}\tau(\sigma(x_i)) = x_i$, $\nu \circ \mu$ is the identity on $F(X)$. In particular, $\nu$ is surjective. Let $\overline{\mu} : G \to G/N'$ be the projection map. The homomorphism $\pi_Z = \overline{\mu} \circ \pi_X \circ \nu$ is surjective since $\overline{\mu}$, $\pi_X$ and $\nu$ are. We have defined an epimorphism from $F(Z)$ to $G/N'$. It is routine to check that $(\mu, \nu)$ is a defining pair for $G/N'$. Next we will find the kernel of $\pi_Z$.

**Proposition 3.3.1** *$N'$ is generated as a subgroup by the images under $\pi_X$ of the elements in $P = \{[W_1^{W_1'}, W_2^{W_2'}] \mid W_1, W_2 \in Q, W_1', W_2'$ are words on $X\}$.*

This is clear since $N$ is generated as a normal subgroup by $\pi_X(Q)$. As a result of this proposition, the kernel of the homomorphism $\overline{\mu} \circ \pi_X$ is the normal subgroup generated by $P \cup R$.

**Lemma 3.3.2** *1. If $W$ is a word on $Y$, then $\tau(W) = \nu(W)$.*

*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 follows from the definition of $\nu$.
2. It is clear from the definition of $\nu$ that for every $\zeta$ in $Z'$, $(\pi_X \circ \nu)([\zeta])$ is contained in $N$. Now $N$ is normal in $G$. Therefore 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 $\pi_Y = \overline{\sigma} \circ \pi_X \circ \tau$, $(\overline{\sigma} \circ \pi_X \circ \tau)(W_1) = (\overline{\sigma} \circ \pi_X \circ \tau)(W_2)$. Now $W_1, W_2$ are words in $Y$. So $\tau(W_1) = \nu(W_1)$ and $\tau(W_2) = \nu(W_2)$. In other words, $(\pi_X \circ \nu)(W_1)$ is equal to $(\pi_X \circ \nu)(W_2)$ modulo $N$. □

The following theorem gives a presentation for the group $G/N'$.

**Theorem 3.3.3** *The kernel of $\pi_Z$ is generated by the following relations:*

$$\begin{array}{lrcll}
(1) & [m_1]\,[m_2] & = & [m_1 + m_2], & m_1, m_2 \in M, \\
(2) & a[m] & = & [ma^{-1}]a, & m \in M, a \in Y^{\pm}, \\
(3) & U_i & = & [z_i']V_i, & 1 \le i \le s, \\
(4) & a & = & \mu(\nu(a)), & a \in Y, \\
(5) & \mu(W) & = & 1, & W \in R.
\end{array}$$

**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 3.2 part 2, for every $m$ in $M$, $\pi_Z([m])$ is in $N/N'$. So for every $m_1$ and $m_2$ in $M$, $\pi_Z([m_1])$ 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 $U$ a collected word on $Y$. Assume that $a$ is in $Y^{\pm}$. Let $V$ be the collected form of $Ua^{-1}$. So $\pi_Y(V) = \pi_Y(Ua^{-1})$ and by Lemma 3.2 part 3, $\pi_Z(V)h = \pi_Z(Ua^{-1})$ for some $h$ in $N/N'$. Therefore $\pi_Z(a[\zeta U]a^{-1}) = \pi_Z(a)\pi_Z(U^{-1})\pi_Z([\zeta])\pi_Z(U)\pi_Z(a^{-1}) = h^{-1}\pi_Z(V)^{-1}\pi_Z([\zeta])\pi_Z(V)h = h^{-1}\pi_Z([\zeta V])h = \pi_Z([\zeta Ua^{-1}])$. By definition of $\nu([z_i'])$, the relations of type (3) are true in $L_2$. The relations of types (4) and (5) are clearly true in $L_2$ since $\nu \circ \mu$ is the identity homomorphism in $F(X)$.

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 $\overline{\mu} \circ \pi_X$ under $\mu$ lies in $L_1$. For the first claim, let $1 \leq i \leq s$. We have $\mu(\nu([z_i'])) = \mu(\tau(U_iV_i^{-1})) = \mu(\nu(U_iV_i^{-1}))$. By the relations of types (3) and (4), $\mu(\nu([z_i'])) = U_iV_i^{-1} = [z_i']$ modulo $L_1$. Now let $1 \leq i \leq r$. We have $\mu(\nu([z_i])) = \mu(\tau(\sigma(x_i)))\mu(x_i^{-1}) = \mu(\nu(\sigma(x_i)))\mu(x_i^{-1})$. Since $\sigma(x_i)$ is a word on $Y$, $\mu(\nu([z_i])) = \sigma(x_i)\mu(x_i^{-1}) = [z_i]$ modulo $L_1$. Therefore 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. By the remark following Proposition 3.1, $P \cup R$ generates the kernel of $\overline{\mu} \circ \pi_X$ as a normal subgroup. 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 contained in $L_1$. Using the relations of type (5), we can see that $\mu(R)$ is contained in $L_1$ also. This proves the second claim. So if $1 = \pi_Z(W) = \overline{\mu} \circ \pi_X \circ \nu(W)$ for some element $W$ in $F(Z)$, then $\mu(\nu(W))$ is in $L_1$. Now $W$ is equal to $\mu(\nu(W))$ modulo $L_1$. 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 $z'_1, z'_2, \ldots, z'_s$. The relations of types (4) and (5) can be rewritten using this rewriting system. Suppose that $a$ 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, $a$ can be rewritten to a word $[m_2]W'$ or to $W'$ when $m_2 = 0$. Let $m$ be $m_1 - m_2$ 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 above.

**Theorem 3.3.4** *$M/M_0$ and $N/N'$ are isomorphic as right $\mathbb{Z}[G/N]$-modules.*

**Proof:** Consider the extension rewriting system $\overline{\mathcal{R}}$ relative to the right $\mathbb{Z}[G/N]$-module $M/M_0$ and the elements $z'_1 + M_0, z'_2 + M_0, \ldots, z'_s + M_0$. Let $\overline{G}$ be the group associated with $\overline{\mathcal{R}}$. By Proposition 2.1, $\overline{\mathcal{R}}$ is confluent. Therefore if $[m_1 + M_0] = [m_2 + M_0]$ in $\overline{G}$, then $[(m_1 - m_2) + M_0]$ is the identity in $\overline{G}$. None of the rewriting rules in $\overline{\mathcal{R}}$ can be used to reduce $[(m_1 - m_2) + M_0]$. So for it to be equal to the identity in $\overline{G}$, $m_1 - m_2$ must be in $M_0$. The groups $F(Z)/L_1$ and $\overline{G}$ 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 $\overline{G}$ to $[m]L_1$ for $m$ in $M$, and the image of $a$ in $\overline{G}$ to $aL_1$ for $a$ in $Y$. Let $\overline{\pi}$ be the quotient map induced from $\pi_Z$. Then $\overline{\pi}$ gives an isomorphism between the groups $F(Z)/L_1$ and $G/N'$. Let $A$ be the subgroup of $F(Z)/L_1$ generated by the set $\{[m]L_1 \mid m \in M\}$. Using the fact that every element in $F(Z)$ can be written as a collected word on $Z$ modulo $L_1$ and parts 2 and 3 of Lemma 3.2, we can see that $\overline{\pi}(A) = N/N'$. Every element in $A$ is of the form $[m]L_1$ for some $m$ in $M$. Under the isomorphism $\rho^{-1}$, $A$ can be identified with $\{[m + M_0] \mid m \in M\} = M/M_0$ since $\overline{\mathcal{R}}$ is confluent. So $M/M_0$ and $N/N'$ are isomorphic as right $\mathbb{Z}[G/N]$-modules through their respective induced right module structures. But the induced right module structures on $M/M_0$ and $N/N'$ are just the usual right $\mathbb{Z}[G/N]$-module structures on $M/M_0$ and $N/N'$. Therefore $M/M_0$ and $N/N'$ are isomorphic as right $\mathbb{Z}[G/N]$-modules. $\square$

Our goal is to study the module structure of $N/N'$ by looking at $M/M_0$. Notice that $M$ is a free right $\mathbb{Z}[G/N]$-module of finite rank and $M_0$ can be generated by finitely many module elements. We would like to be able to find group generators in $M/M_0$ and for each $m$ in $M$, express $m + M_0$ in terms of the group generators in $M/M_0$. To do these, we will first find a Gröbner basis for the submodule $M_0$ of $M$.

# Chapter 4

# Gröbner Bases

The study of Gröbner bases originates in the thesis of Buchberger [Buc65], written under the supervision of Wolfgang Gröbner in 1965. The most usual setting in which Gröbner bases are used is in $k[X]$, a polynomial ring with a finite set of indeterminates $X$ over a field $k$. By the Hilbert Basis Theorem, $k[X]$ is Noetherian. Every ideal of $k[X]$ is finitely generated. Suppose that we have a finite set $S$ of polynomials in $k[X]$ and we would like to determine membership in $\langle S \rangle$, the ideal generated by $S$. The idea is to find a nice set of ideal generators for $\langle S \rangle$ first. When $X$ consists of only one indeterminate, the ring $k[X]$ is a principal ideal domain and the nice set of ideal generators consists of only one polynomial, which is just the greatest common divisor of the nonzero polynomials in $S$ and can be found by the Euclidean algorithm. In fact, Buchberger's algorithm for computing Gröbner bases can be thought of as a multivariate version of the Euclidean algorithm. Two good introductory references on Gröbner bases are [BW91] and [Mor94].

The generalization of Gröbner bases to noncommutative rings is an area of active research. Let $k\langle X \rangle$ denote the free associative algebra over the field $k$ with $X$ as the set of indeterminates. The natural question is: can Buchberger's algorithm be generalized to $k\langle X \rangle$? The answer to this question is an emphatic "no". In fact, the ideal membership problem for two-sided ideals is already algorithmically unsolvable in the case when $k$ is the field of rational numbers and $X$ consists of two variables. For more details refer to [Mor86],[Mor88] and [KRW90].

Gröbner basis computation has been studied in path algebras [FFG93]. Termination of such computation is not always guaranteed. Two classes of noncommutative rings in which Gröbner basis computation with termination is possible are the class of enveloping algebras of finite dimensional Lie algebras [AL89], and more generally, the class of algebras of solvable type [KRW90]. Gröbner basis computation has also been extended to Clifford algebras and Grassmann algebras in [HT95]. For Gröbner basis computation in non-commutative algebras in general, refer to [Mor89].

In [Sim90], Sims showed that Gröbner basis computation with termination is possible in the integral group ring of a finitely generated abelian group. In this chapter, we will show that Gröbner basis computation with termination can also be performed in the integral group ring of a polycyclic group.

## 4.1 Introduction to Gröbner Bases

In this section, we will briefly describe Gröbner bases in polynomial rings over fields. Readers familiar with this material may wish to move on to the next section.

Let $k$ be a field and $X = \{x_1, x_2, \ldots, x_n\}$ be a finite set of indeterminates. Any ideal $\mathcal{I}$ in $k[X]$ is finitely generated. The basic question is: let $P$ be a nonempty finite subset of $k[X]$ and let $\mathcal{I}$ be the ideal generated by $P$. Can we decide membership in $\mathcal{I}$?

Let $T$ be the set of terms $x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n}$. Then $T$ forms an abelian monoid with the natural multiplication. Let $t_1 = x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n}$ and $t_2 = x_1^{\mu_1} x_2^{\mu_2} \ldots x_n^{\mu_n}$ be elements in $T$. We say that $t_1 \mid t_2$ if $\lambda_i \leq \mu_i$ for every $i$ with $1 \leq i \leq n$.

**Definition:** A total ordering $\leq$ on $T$ is called a *term ordering* if for every $s, t_1, t_2$ in $T$, $s \geq 1$, and $t_1 \leq t_2$ implies $t_1 s \leq t_2 s$.

**Examples:**

1. Consider the monoid $T$. Define the ordering $\leq$ by $x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n} < x_1^{\mu_1} x_2^{\mu_2} \ldots x_n^{\mu_n}$ if there exists an integer $j$ with $1 \leq j \leq n$ such that for each $i$ with $1 \leq i < j$, $\lambda_i = \mu_i$, and $\lambda_j < \mu_j$. Then $\leq$ is a term ordering and is called the *lexicographic ordering* on $T$.

2. Let $\leq$ be the lexicographic ordering on $T$ as above. Define an ordering $\leq'$ by

$x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n} <' x_1^{\mu_1} x_2^{\mu_2} \ldots x_n^{\mu_n}$ if

$$\sum_{i=1}^{n} \lambda_i < \sum_{i=1}^{n} \mu_i,$$

$$\text{or if} \qquad \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \mu_i \qquad \text{and} \qquad x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n} < x_1^{\mu_1} x_2^{\mu_2} \ldots x_n^{\mu_n}.$$

Then $\leq'$ is a term ordering and is called the *total degree ordering* on $T$.

It is easy to see that if $t_1 \mid t_2$, then $t_1 \leq t_2$. The fact that any term ordering is a well-ordering on $T$ will be shown in Proposition 2.6.

Let $k[X]^{\times}$ be the set of nonzero polynomials in $k[X]$. Fix a term ordering $\leq$ on $T$. For a polynomial $f$ in $k[X]^{\times}$, $f$ can be written uniquely as $a_1 t_1 + a_2 t_2 + ... + a_r t_r$ where $r$ is a positive integer, $a_1, a_2, \ldots, a_r$ are nonzero elements in $k$, and $t_1 > t_2 > \ldots > t_r$ are terms. This way of writing $f$ will be called the *standard form* of $f$. Define $\mathrm{lc}(f) = a_1$ to be the *leading coefficient* of $f$, $\mathrm{lm}(f) = a_1 t_1$ to be the *leading monomial* of $f$, and $\mathrm{lt}(f) = t_1$ to be the *leading term* of $f$.

For any subset $P$ of $k[X]$, let $\langle P \rangle$ be the ideal generated by $P$ and $\mathrm{lt}(P)$ be the set of all terms $t$ such that there exists $f$ in $P$ with $\mathrm{lt}(f) = t$. Let $\mathcal{I}$ be a nonzero ideal of $k[X]$. Define $\mathrm{et}(\mathcal{I})$, the set of *essential terms* in $\mathcal{I}$, to be the set $\{t \in \mathrm{lt}(\mathcal{I}) \mid \forall t' \in \mathrm{lt}(\mathcal{I}) - \{t\}, t' \nmid t\}$. Since $\mathcal{I}$ is not trivial, the set $\mathrm{lt}(\mathcal{I})$ is nonempty, but so is $\mathrm{et}(\mathcal{I})$ since the smallest term in $\mathrm{lt}(\mathcal{I})$ in the well-ordering $\leq$ is always in $\mathrm{et}(\mathcal{I})$. Lemma 1.1 below implies that $\mathrm{et}(\mathcal{I})$ is finite. Proofs of Lemma 1.1 and Lemma 1.2 will be given in a more general setting in Section 2.

**Lemma 4.1.1** *If $t_1, t_2, \ldots$ is a sequence of terms such that $t_i \mid t_j$ only when $j \leq i$, then this sequence is finite.*

**Lemma 4.1.2** *For all nonzero $f$ in $\mathcal{I}$, there exists a term $t$ in $\mathrm{et}(\mathcal{I})$ such that $t \mid \mathrm{lt}(f)$.*

**Definition:** Let $\mathcal{I}$ be a nonzero ideal. Let $\mathrm{et}(\mathcal{I}) = \{t_1, t_2, \ldots, t_r\}$. A *minimal Gröbner basis* $\mathcal{B}$ of $\mathcal{I}$ is a set of polynomials $\{f_1, f_2, \ldots, f_r\}$ in $k[X]^{\times}$ such that $\mathrm{lt}(f_i) = t_i$ for $1 \leq i \leq r$.

**Theorem 4.1.3** *Let $\mathcal{B}$ be a minimal Gröbner basis of an ideal $\mathcal{I}$. Then $\mathcal{B}$ is a set of ideal generators for $\mathcal{I}$.*

Fix a set $P$ of polynomials in $k[X]^{\times}$. For $g_1$ in $k[X]^{\times}$, we write $g_1 \rightarrow_P g_2$ if there exists $f$ in $P$ such that $\mathrm{lt}(f) \,|\, \mathrm{lt}(g_1)$ and

$$g_2 = g_1 - \frac{\mathrm{lm}(g_1)}{\mathrm{lm}(f)} f.$$

Such an operation is called a *reduction* on $g_1$. Notice that if $g_1 \rightarrow_P g_2$, then either $g_2 = 0$ or $\mathrm{lt}(g_2) < \mathrm{lt}(g_1)$. Also, $g_1 - g_2$ is in the ideal generated by $P$. We say that $g_1 \xrightarrow{*}_P g$ if $g$ can be obtained from $g_1$ after zero or more reductions. Given $g_1$ in $k[X]^{\times}$, we can only perform a finite number of reductions on $g_1$. More precisely, in a finite number of steps, $g_1$ will either be reduced to 0 or to a nonzero polynomial $g$ where $\mathrm{lt}(g) \not\succeq \mathrm{lt}(f)$ for every $f$ in $P$. Here is a sketch of the proof. Let $g_1 \rightarrow_P g_2, g_2 \rightarrow_P g_3, \ldots$ be successive steps in reduction. Let $t_1 = \mathrm{lt}(g_1)$, $t_2 = \mathrm{lt}(g_2)$, $\ldots$. Then $t_1 > t_2 > \ldots$. Since $\leq$ is a well-ordering, the sequence $t_1, t_2, \ldots$ has to be finite. So only finitely many reductions can be performed.

If $\mathcal{B}$ is a minimal Gröbner basis of a nonzero ideal $\mathcal{I}$, then membership testing can be done by performing reductions with $\mathcal{B}$. Let $g_1$ be a polynomial. In a finite number of steps, $g_1$ can be reduced either to 0 or to a nonzero $g$ which cannot be reduced by $\mathcal{B}$. If $g_1 \xrightarrow{*}_{\mathcal{B}} 0$, then $g_1$ is contained in $\langle \mathcal{B} \rangle \subseteq \mathcal{I}$. For the other case, if $g$ is in $\mathcal{I}$, then $\mathrm{lt}(g) \succeq t$ for some $t$ in $\mathrm{et}(\mathcal{I})$. So $g$ can be reduced further by $\mathcal{B}$ and this is a contradiction. Therefore $g$ is not in $\mathcal{I}$ and since $g_1 - g$ is in $\mathcal{I}$, $g_1$ is not in $\mathcal{I}$.

The above paragraph gives a proof of Theorem 1.3. If $g_1$ is in $\mathcal{I}$, then $g_1 \xrightarrow{*}_{\mathcal{B}} 0$ and so $g_1$ is in $\langle \mathcal{B} \rangle$. Therefore $\mathcal{I} \subseteq \langle \mathcal{B} \rangle$. Clearly $\langle \mathcal{B} \rangle \subseteq \mathcal{I}$.

Let $P = \{f_1, f_2, \ldots, f_r\}$ be a set of polynomials. The problem of deciding membership in $\langle P \rangle$ is now reduced to computing a minimal Gröbner basis of $\langle P \rangle$. Let $1 \leq i < j \leq r$. If $\mathrm{lt}(f_i) = x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n}$ and $\mathrm{lt}(f_j) = x_1^{\mu_1} x_2^{\mu_2} \ldots x_n^{\mu_n}$, then take

$$t = x_1^{\max(\lambda_1,\mu_1)} x_2^{\max(\lambda_2,\mu_2)} \ldots x_n^{\max(\lambda_n,\mu_n)} \quad \text{and} \quad \beta_{ij} = \frac{t}{\mathrm{lm}(f_i)} f_i - \frac{t}{\mathrm{lm}(f_j)} f_j.$$

**Theorem 4.1.4** *If for every $i, j$, $\beta_{ij} \xrightarrow{*}_P 0$, then $P$ contains a minimal Gröbner basis of $\langle P \rangle$.*

**Sketch of proof:** Assume the hypothesis. Let $T'$ be the set of terms $t$ in $\mathrm{lt}(P)$ such that if $t'$ is in $\mathrm{lt}(P)$ and $t'$ divides $t$, then $t' = t$. For each $t$ in $T'$, let $f_t$ be a polynomial in $P$ such that $\mathrm{lt}(f_t) = t$. Let $P'$ be the set of all such $f_t$'s. We want to show that $P'$ is a minimal Gröbner basis of $\langle P \rangle$. Assume not, then there exists a nonzero polynomial $f$ in $\langle P \rangle$ such that $\mathrm{lt}(f_t) \nmid \mathrm{lt}(f)$ for each $f_t$ in $P'$. Since $f$ is in $\langle P \rangle$, $f$ can be written as $\bar{f}_1 h_1 + \bar{f}_2 h_2 + \ldots + \bar{f}_s h_s$, where the $\bar{f}_i$'s are distinct polynomials in $P$ and the $h_i$'s are in $k[X]^{\times}$. Without loss of generality, we may assume that $\mathrm{lt}(\bar{f}_1 h_1) = \mathrm{lt}(\bar{f}_2 h_2) = \ldots = \mathrm{lt}(\bar{f}_q h_q)$ is bigger than all $\mathrm{lt}(\bar{f}_{q+1} h_{q+1}), \mathrm{lt}(\bar{f}_{q+2} h_{q+2}), \ldots, \mathrm{lt}(\bar{f}_s h_s)$. Among all such ways of writing $f$, consider those in which $t = \mathrm{lt}(\bar{f}_1 h_1)$ is smallest in $\leq$, and among them, take one such that $q$ is smallest. Now all $\beta_{ij}$ can be reduced to zero. With some work it can be seen that $q$ has to be equal to 1. Therefore $\mathrm{lt}(f) = \mathrm{lt}(\bar{f}_1 h_1)$ is a multiple of $\mathrm{lt}(\bar{f}_1)$. This contradicts our assumption that $\mathrm{lt}(f_t) \nmid \mathrm{lt}(f)$ for each $f_t$ in $P'$. □

Theorem 1.3 gives us an algorithm to compute a minimal Gröbner basis given a set of ideal generators $P = \{f_1, f_2, \ldots, f_r\}$. Set $S$ to be $\{\beta_{ij} \mid 1 \leq i < j \leq r\}$ and $P'$ to be $P$. While $S$ is nonempty, take a polynomial $\beta_{ij}$ from $S$ and reduce it with respect to $P'$ until it cannot be reduced any further. If the answer $g$ is nonzero, let $f_{|P'|+1} = g$ and $p = |P'|$ and reset $P'$ to be $P' \cup \{g\}$. Reset $S$ to be $S - \{\beta_{ij}\} \cup \{\beta_{i(p+1)} \mid 1 \leq i \leq p\}$. Now loop back to process another polynomial in $S$. To show that this algorithm will terminate, it suffices to show that the list of $g$ produced, $f_{r+1}, f_{r+2}, \ldots$, has to end. For $i \geq 1$, $f_{r+i}$ cannot be reduced by $f_{r+1}, f_{r+2}, \ldots, f_{r+i-1}$. So if $1 \leq j < k$, then $\mathrm{lt}(f_{r+j}) \nmid \mathrm{lt}(f_{r+k})$. By Lemma 1.1, this list has to end eventually. So this computation is finite. Here is the algorithm described above.

Procedure Min_Gröbner_Basis($P$,$\mathcal{B}$)

Input: $P$ : A nonempty set of nonzero polynomials.

Output: $\mathcal{B}$ : A minimal Gröbner basis of $\langle P \rangle$.

Begin

$P' := P$;

$S := \{\beta_{ij} \mid 1 \leq i < j \leq |P|\}$;

While $S \neq \emptyset$ do begin

(* At all times $P' = \{f_1, f_2, \ldots, f_p\}$. *)

Take a polynomial $h$ from $S$;

While $h \neq 0$ and there exists $f_i$ in $P'$ with $\mathrm{lt}(f_i) \mid \mathrm{lt}(h)$ do $h := h - \frac{\mathrm{lm}(h)}{\mathrm{lm}(f_i)} f_i$;

If $h \neq 0$ then begin

$f_{p+1} := h; \qquad P' := P' \cup \{f_{p+1}\}; \qquad S := S \cup \{\beta_{i(p+1)} \mid 1 \leq i \leq p\}$;

end

end;

Let $T' = \{t \in \mathrm{lt}(P') \mid$ if $t' \in \mathrm{lt}(P')$ and $t' \mid t$, then $t' = t\}$;

For each $t$ in $T'$, find a polynomial $f_t$ in $P'$ such that $\mathrm{lt}(f_t) = t$;

Let $\mathcal{B}$ be the set of all such $f_t$'s;

end.

## 4.2 Partially Ordered Sets

Let $S$ be a set. A *partial ordering* $\preceq$ is a reflexive, antisymmetric and transitive relation on $S$. In other words, for every $s_1, s_2, s_3$ in $S$, $s_1 \preceq s_1$, $s_1 \preceq s_2$ and $s_2 \preceq s_1$ imply $s_1 = s_2$, and $s_1 \preceq s_2$ and $s_2 \preceq s_3$ imply $s_1 \preceq s_3$. We call a set $S$ with a partial ordering $\preceq$ defined a *partially ordered set*, or a *poset* for short. Given $s_1, s_2$ in $S$, we define $s_1 \prec s_2$ if $s_1 \preceq s_2$ and $s_1 \neq s_2$. If $s_1 \prec s_2$, we say that $s_1$ *precedes* $s_2$. Clearly, the relations $\preceq$ and $\prec$ uniquely determine one another.

**Definition:** We say that a poset $S$ is a *well-poset* if for any infinite sequence $s_1, s_2, \ldots$ of elements in $S$, there exists $i, j$ such that $i < j$ and $s_i \preceq s_j$. If $S$ is a well-poset, then the partial ordering on $S$ is a *well-partial-ordering.*

Alternatively, a poset $S$ is a well-poset if every sequence $s_1, s_2, \ldots$ in which $s_i \not\preceq s_j$ whenever $i < j$ is finite.

In [BW91], well-partial-orderings are called Dickson partial orders.

**Examples:**

1. Let $N$ be the set of nonnegative integers and for $s_1, s_2$ in $N$, define $s_1 \preceq s_2$ if

$s_1 \le s_2$ under the usual ordering on $N$. Under $\preceq$, $N$ is a well-poset.

2. Let $S$ be a set of $n \ge 1$ elements. For every $s_1, s_2$ in $S$, define $s_1 \preceq s_2$ if $s_1 = s_2$. In other words, $s_1 \not\prec s_2$ 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.

**Lemma 4.2.1** *Let $S'$ be an infinite subset of a well-poset $S$. Then we can find an infinite sequence of elements $s_1 \prec s_2 \prec \ldots$ in $S'$.*

**Proof:** Let $T$ be any infinite subset of $S$ and let $s$ be an element in $S$. Define $U_T(s)$ to be $\{t \in T \mid s \prec t\}$. First we want to show that $U_T(t)$ is infinite for some $t$ in $T$. Assume by contradiction that $U_T(t)$ is finite for every $t$ in $T$. Define a sequence as follow. Take $t_1$ to be any element in $T$, take $t_2$ to be any element in $T - U_T(t_1)$. Inductively, assume that we have chosen $t_1, t_2, \ldots, t_k$. Since $U_T(t_1), U_T(t_2), \ldots, U_T(t_k)$ are finite, the set $T - U_T(t_1) - U_T(t_2) - \ldots - U_T(t_k)$ is nonempty. Pick $t_{k+1}$ to be one of its elements. Therefore we have an infinite sequence of elements such that whenever $1 \le i < j$, $t_i \not\preceq t_j$. This is a contradiction. So $U_T(t)$ is infinite for some $t$ in $T$.

We are given that $S'$ is infinite. By the argument above, $U_{S'}(s_1)$ is infinite for some $s_1$ in $S'$. Let $T$ be $U_{S'}(s_1)$. Using the same argument we see that there exists $s_2$ in $T$ such that $U_T(s_2)$ is infinite. But $U_T(s_2)$ is just $U_{S'}(s_2)$. Inductively, assume that we have found $s_1, s_2, \ldots, s_k$ in $S'$ such that $s_i$ is in $U_{S'}(s_{i-1})$ whenever $2 \le i \le k$, and $U_{S'}(s_k)$ is infinite. Let $T$ be $U_{S'}(s_k)$. There exists $s_{k+1}$ in $T$ such that $U_T(s_{k+1}) = U_{S'}(s_{k+1})$ is infinite. Therefore we can find an infinite sequence $s_1 \prec s_2 \prec \ldots$ in $S'$. $\square$

A *subsequence* of a sequence $s_1, s_2, \ldots$ is a sequence of element $s_{i_1}, s_{i_2}, \ldots$ such that $i_1 < i_2 < \ldots$.

**Corollary 4.2.2** *Let $s_1, s_2, \ldots$ be an infinite sequence of not necessarily distinct elements in $S$, then there exists an infinite subsequence $s_{i_1}, s_{i_2}, \ldots$ such that $s_{i_1} \preceq s_{i_2} \preceq \ldots$.*

**Proof:** Let $T$ be the set of elements $s_i$ in the sequence. If $T$ is finite, then for some $t$ in $S$, $t$ appears in the sequence infinitely many times. Then the subsequence which consists of all the $t$'s from the original sequence is the subsequence we want. Now if $T$ is infinite, then $T$ contains an infinite sequence $t_1 \prec t_2 \prec \ldots$ by Lemma 2.1. For $k \geq 1$, $t_k$ is some $s_{j_k}$ in the sequence $s_1, s_2, \ldots$. The sequence $s_{j_1}, s_{j_2}, \ldots$ may not be a subsequence of $s_1, s_2, \ldots$, but we can certainly pick out a subsequence which is. $\square$

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

**Proof:** Let $S$ and $T$ be two well-posets with partial orderings $\preceq_s$ and $\preceq_t$, respectively. Let $(s_1, t_1), (s_2, t_2), \ldots$ be an infinite sequence of distinct elements in $S \times T$. Consider the infinite sequence $s_1, s_2, \ldots$. By Corollay 2.2 there exists an infinite subsequence $s_{i_1}, s_{i_2}, \ldots$ such that $s_{i_1} \preceq_s s_{i_2} \preceq_s \ldots$. Now consider the infinite sequence $t_{i_1}, t_{i_2}, \ldots$. By Corollary 2.2 again there exists an infinite subsequence $t_{i_{j_1}}, t_{i_{j_2}}, \ldots$ such that $t_{i_{j_1}} \preceq_t t_{i_{j_2}} \preceq_t \ldots$. If $k = i_{j_1}$ and $l = i_{j_2}$, then we have $k < l$, $s_k \preceq_s s_l$, $t_k \preceq_t t_l$ and so $(s_k, t_k) \preceq (s_l, t_l)$. Therefore $S \times T$ is a well-poset. $\square$

By induction, if $S_1, S_2, \ldots, S_k$ are well-posets, then so is their product poset $S_1 \times S_2 \times \ldots \times S_k$. For any poset $S$, we write $S^k$ to denote the product poset $S \times S \times \ldots \times S$. For those readers who have read Section 1, recall that $T$ is the set of terms and $n$ is the number of indeterminates. By defining $t_1 \preceq t_2$ whenever $t_1 \mid t_2$ in $T$, we get a partial ordering on $T$. The poset structure of $T$ is just $N^n$. Since $N$ is a well-poset, so is $T$ and Lemma 1.1 follows. Lemma 1.2 follows from Theorem 2.4 below.

Let $S'$ be a subset of a poset $S$. We define $\min(S')$ to be $\emptyset$ if $S' = \emptyset$. Otherwise $\min(S')$ is the set $\{s \in S' \mid \forall s' \in S', s' \not\prec s\}$.

**Proposition 4.2.4** *Let $S$ be a well-poset. Let the partial ordering in $S$ be $\preceq$ and let $S_1$ be a nonempty subset of $S$. Let $S_2$ be* $\min(S_1)$*. Then $S_2$ is nonempty and finite. In addition, for any $s$ in $S_1$, there exists $s'$ in $S_2$ such that $s' \preceq s$.*

**Proof:** Let us prove the second statement first. Let $s_0$ be in $S_1$. If $s_0$ is in $S_2$ then we are done. Otherwise there exists $s_1$ in $S_1$ such that $s_1 \prec s_0$. If $s_1$ is contained in $S_2$, then again we are done. Otherwise there exists an element $s_2$ in $S_2$ with $s_2 \prec s_1 \prec s_0$. Inductively, we get a sequence $s_0 \succ s_1 \succ s_2 \succ \ldots$. This sequence has to terminate since $S$ is a well-poset. Let $s_i$ be the last term of the sequence. Then $s_i$ is contained in $S_2$ and $s_0 \succeq s_i$. This proves the second statement. The fact $S_2$ is nonempty follows from the second statement and the fact that $S_1$ is nonempty. If $S_2$ is infinite then, by Lemma 2.1, there will be elements $s_1, s_2$ in $S_2$ such that $s_1 \prec s_2$. This is clearly not possible by the definition of $S_2$. □

**Proposition 4.2.5** *Suppose that $S$ is a poset with partial ordering $\preceq$ such that* $\min(S_1)$ *is finite and nonempty for any nonempty subset $S_1$ of $S$. Then $S$ is a well-poset.*

**Proof:** Suppose $s_1, s_2, \ldots$ is an infinite sequence in $S$. We want to show that there are positive integers $i$ and $j$ such that $i < j$ and $s_i \preceq s_j$. We may assume that $s_1, s_2, \ldots$ are all distinct. Let $S_1$ be $\{s_1, s_2, \ldots\}$ and $S_2$ be $\min(S_1)$. By the hypothesis $S_2$ is finite and nonempty. Let $S_2$ be $\{s_{l_1}, s_{l_2}, \ldots, s_{l_k}\}$ where $k > 0$ and $l_1 < l_2 < \ldots < l_k$. Now $\min(\{s_{l_k+1}, s_{l_k+2}, \ldots,\})$ is nonempty. Let $s_j$ be an element in it. Since $s_j$ is not in $S_2 = \min(S_1)$, there is an element $s_i$ such that $s_i \prec s_j$. Using the fact that $s_j$ is in $\min(\{s_{l_k+1}, s_{l_k+2}, \ldots,\})$, we see that $i \leq l_k < j$. □

Proposition 2.4 and Proposition 2.5 give another characterization of a well-poset. A poset $S$ is a well-poset if $\min(S')$ is nonempty and finite for each subset $S'$ of $S$.

In a private communication with Dr. Laszlo Kovacs from the Australian National University, it was pointed out to me that an infinite poset must have either an infinite ascending chain or an infinite descending chain or an infinite antichain or any combination of the three. Thus another characterization of a well-poset is one which has no infinite descending chain and no infinite antichain. This can be seen by Proposition 2.5.

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

**Proposition 4.2.6** *Let $S$ be a well-poset with partial ordering $\preceq$. Any linear extension $\leq$ of $\preceq$ is a well-ordering.*

**Proof:** Let $T$ be a nonempty subset of $S$. We will find an element $s$ in $T$ such that for each $t$ in $T$, $t \geq s$. By Proposition 2.4, $\min(T) = \{s \in T \mid \forall t \in T, t \not\prec s\}$ is nonempty and finite. So it is possible to find an element $s$ in $\min(T)$ such that $s \leq s'$ for each $s'$ in $\min(T)$. Now if $t$ is an element in $T$, by the second statement of Proposition 2.4, $t \succeq s'$ for some $s'$ in $\min(T)$. So $t \geq s' \geq s$. □

**Definition:** Suppose that $\preceq$ is a partial ordering defined on a set $S$. Let $s_1$ and $s_2$ be elements in $S$. Define $lcm(s_1, s_2)$ to be $\min(\{s \in S \mid s_1, s_2 \preceq s\})$.

If $S$ is a well-poset with partial ordering $\preceq$, then for any elements $s_1, s_2$ in $S$, $\mathrm{lcm}(s_1, s_2)$ is finite.

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

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 $\operatorname{lcm}((s_1, t_1), (s_2, t_2)) = (\operatorname{lcm}(s_1, s_2), \operatorname{lcm}(t_1, t_2))$ if it exists.

### 4.3 Group Reduction Orderings

In Section 1, we saw Gröbner bases in polynomial rings over a field. In this and the next few sections, we will attempt to generalize Gröbner bases to submodules of free right modules of the integral group ring $\mathbb{Z}[G]$ of a group $G$ satisfying certain properties. In Section 1, we began by defining appropriate partial ordering and total ordering on the set of terms. Here we use the notion of a reduction ordering set. As we shall see later, if a reduction ordering set exists, then the Gröbner basis method can be generalized.

The *integral group ring* $\mathbb{Z}[G]$ of a group $G$ is the set of functions $f : G \to \mathbb{Z}$ where $f(x)$ is 0 for all but a finite number of elements $x$ in $G$. Define the *support* of $f$, $\operatorname{supp}(f)$, to be the finite set $\{x \in G \mid f(x) \neq 0\}$. We define addition and multiplication in $\mathbb{Z}[G]$ in a natural way. Let $f$ and $g$ be two functions in $\mathbb{Z}[G]$. We define $f + g$ by $(f + g)(x) = f(x) + g(x)$ for each $x$ in $G$. Multiplication is defined as follows:

$$(fg)(x) = \sum_{x_1 x_2 = x} f(x_1) g(x_2)$$

for each $x$ in $G$. For each $x$ in $G$, we associate with it the characteristic function of $\{x\}$, again denoted by $x$, in $\mathbb{Z}[G]$ such that $x(x) = 1$ and $x(x') = 0$ for every $x' \neq x$ in $G$. If $\operatorname{supp}(f) = \{x_1, x_2, \ldots, x_s\}$ and $f(x_i) = c_i$ for $1 \leq i \leq s$, then $f = \sum_{i=1}^{s} c_i x_i$.

The ring $\mathbb{Z}[G]$ is commutative if and only if $G$ is abelian. In this thesis, $G$ is usually not abelian and so $\mathbb{Z}[G]$ is usually not commutative. We need to distinguish between left $\mathbb{Z}[G]$-modules and right $\mathbb{Z}[G]$-modules. From here on, when we refer to a $\mathbb{Z}[G]$-module, we mean a right $\mathbb{Z}[G]$-module, unless otherwise stated.

Suppose that $G$ is a polycyclic group. Let $(a_1, a_2, \ldots, a_n)$ be a polycyclic generating sequence for $G$. Group elements in $G$ can be written as $a_n^{\lambda_n} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$ for some integers $\lambda_n, \ldots, \lambda_2, \lambda_1$. Therefore every element in $\mathbb{Z}[G]$ is an integer linear combination of terms of the form $a_n^{\lambda_n} \ldots a_2^{\lambda_2} a_1^{\lambda_1}$. In comparison, in a polynomial ring $k[X]$ with $X =$

$\{x_1, x_2, \ldots, x_n\}$, every element is an integer linear combination of power products of the form $x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_n^{\lambda_n}$. This similarity prompts us to call elements in $\mathbb{Z}[G]$ *polynomials*, with an abuse of language. Polynomials of the form $cx$ where $c$ is a nonzero integer and $x$ is in $G$ are called *monomials*. In particular, $x$ is also a monomial.

Let $G$ be a group. We will consider submodules of a free right $\mathbb{Z}[G]$-module $\mathbb{Z}[G]^k$, $k \geq 1$. In the case when $k = 1$, the submodules are just the right ideals of $\mathbb{Z}[G]$. We call an element in $\mathbb{Z}[G]^k$ 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 $\mathbb{Z}[G]^k$. In other words, for $1 \leq i \leq k$, $e_i$ is the $k$-tuple $(0, \ldots, 0, 1, 0, \ldots, 0)$ with 1 occurring at the $i$-th position. The element $(f_1, f_2, \ldots, f_k)$ above is equal to $e_1 f_1 + e_2 f_2 + \ldots + e_k f_k$. Now every element in $\mathbb{Z}[G]$ can be written as $c_1 x_1 + c_2 x_2 + \ldots + c_r x_r$ for some nonzero integers $c_1, c_2, \ldots, c_r$ and distinct elements $x_1, x_2, \ldots, x_r$ in $G$. Therefore every element in $\mathbb{Z}[G]^k$ is a finite sum of elements of the form $e_i(cx)$, 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*. Note that monomial vectors and term vectors are polynomial arrays also. The monomial vector $e_i(cx)$ will be written as $ce_ix$. Let $\mathcal{M}$ be the set of monomial vectors and $\mathcal{T}$ be the set of term vectors.

Let us examine what we need to perform the analogue of Gröbner basis computation in $\mathbb{Z}[G]$. In Section 1, we saw two total orderings on $k[X]$, the lexicographic ordering and the total degree ordering. They both have two important properties. First, they are both well-orderings. Second, both of them are consistent with multiplication. What we mean is that, if $t_1, t_2$ and $t$ are terms in $k[X]$, then $t_1 < t_2$ implies $t_1 t < t_2 t$. These two properties are important in making reductions work in $k[X]$. Things can be very different in $\mathbb{Z}[G]$. Let us consider $\mathbb{Z}[G]^1$. Here term vectors can be viewed as elements in $G$. In general there is no well-ordering $\leq$ on $G$ which is consistent with multiplication. In fact, the only exception is when $G$ is the trivial group. To prove this statement, we take a nontrivial cyclic subgroup $H$ of $G$. Let $H$ be generated by $a$. Suppose that $\leq$ is a well-ordering on $G$ and $\leq$ is consistent with multiplication. First, let us consider the case when $H$ is finite with order $m$. If $a^0 < a^1$, then $a^0 < a^1 < a^2 < \ldots < a^{m-1} < a^m = a^0$, which is not possible. We also get a contradiction if $a^0 > a^1$. Now consider the case when $H$ is infinite. If $a^0 < a^1$, then $\ldots < a^{-2} < a^{-1} < a^0 < a^1 < a^2 < \ldots$. So $\leq$ is not a well-ordering. The same conclusion can be obtained if $a^0 > a^1$.

It turns out that it is more important for the total ordering to be a well-ordering. Actually, we can rephrase the consistency with multiplication property to make things work. In $k[X]$, we reduce a polynomial $g_1$ by a polynomial $f$ in $P$ only if $\mathrm{lt}(f)$ divides $\mathrm{lt}(g_1)$. Let $\mathrm{lt}(f)$ be $t_2$ and $\mathrm{lt}(g_1)$ be $t_2 t$. All we need is just the following condition: if $t_2 \prec t_2 t$ and $t_1 < t_2$, then $t_1 t < t_2 t$. We call this condition the *restricted consistency property*. This property ensures that $\mathrm{lt}(g_2) < \mathrm{lt}(g_1)$ if $g_1 \rightarrow_P g_2$. In the case of $k[X]$, since $t_2 \preceq t_2 t$ is always true, the restricted consistency property is a consequence of the fact that the total ordering is consistent with multiplication. In the case of the integral group ring of a polycyclic group, the extra condition $t_2 \preceq t_2 t$ is neccessary to make things work.

We will try to construct a partial ordering $\preceq$ and a total ordering $\leq$ on the set of terms such that $\leq$ is a well-ordering which satisfies the restricted consistency property. Clearly, if $t_2 \prec t_1$ for only a few pairs of terms $t_1$ and $t_2$, then we have only a few terms $t$ such that $t_2 \preceq t_2 t$ and so it is easier for $\leq$ to have the restricted consistency property. However, to make the Gröbner basis computation more effective, we want more pairs in the partial ordering since we can reduce a polynomial more with more pairs.

We will look at the orderings $\preceq$ and $\leq$ more carefully. We will not only consider right ideals in $\mathbb{Z}[G]$ but also submodules in free right $\mathbb{Z}[G]$-modules. But first let us specify the conditions on the group $G$.

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

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

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

Note that the third condition is just the restricted consistency property stated in a slightly different form.

To work in $\mathbb{Z}[G]^k$, we need to define partial orderings and total orderings on $\mathcal{T}$ and $\mathcal{M}$.

**Definition:** Suppose that $G$ 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 $\frac{t_2}{t_1}$, such that $t_1 x = t_2$, and,
4. for every $t_1, t_2, t'$ in $\mathcal{T}$, if $t_1 \prec t_2$ and $t' < t_1$, then $t' \frac{t_2}{t_1} < t_2$.

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 $\preceq$.

It is clear that the element $x$ in condition 3 is unique.

In each of the two definitions above, we assume that the partial ordering has the unique LCM property. However, such an assumption is not necessary. The reason for the extra assumption is to make the exposition simple.

Suppose that $G$ 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 2.3, $\mathcal{T}$ is a well-poset with the unique LCM property. One way of extending $\leq$ 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 will further extend $\preceq$ and $\leq$ to $\mathcal{M}$. Let the set of nonzero integers $\mathbb{Z} - \{0\}$ be ordered as $1 \ll 2 \ll \ldots \ll -1 \ll -2 \ll \ldots$. 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 $\mathcal{T}$ with their respective partial orderings. By Theorem 2.3, $\mathcal{M}$ 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 $\mathcal{T}$. We define $c_1t_1 < c_2t_2$ if either $t_1 < t_2$, or $t_1 = t_2$ and $c_1 \ll c_2$. Clearly $\leq$ is a linear extension of $\preceq$ in $\mathcal{M}$. By Proposition 2.6, $\leq$ is a well-ordering. Let us call the partial orderings and total orderings on $\mathcal{T}$ 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 $\mathcal{T}$ and $\mathcal{M}$. Collectively, the partial and total orderings on $\mathcal{T}$ and $\mathcal{M}$ form a *reduction ordering set.*

## 4.4 Examples of Group Reduction Orderings

In this section, we will look at some examples of group reduction orderings.

**Examples:**

1. Let $G$ be the infinite cyclic group generated by $a$. Define $a^0 \prec a^1 \prec a^2 \prec \ldots$ and $a^0 \prec a^{-1} \prec a^{-2} \prec \ldots$. With this partial ordering, $G$ is a well-poset with the unique LCM property. In fact, referring to the examples in Section 2, we can see that $G$ as a poset has the same structure as $((N - \{0\}) \times \Phi_2) \cup \{0\}$. If we define $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \ldots$, then $G$ is a reduction group. Another group reduction ordering on $G$ is given by $a^0 < a^1 < a^2 < \ldots < a^{-1} < a^{-2} < \ldots$. In fact, any linear extension of $\preceq$ is a group reduction ordering.
2. Let $G$ be the finite cyclic group of order $m$. Suppose that $a$ generates $G$. Define $a^0 \prec a^1 \prec a^2 \prec \ldots \prec a^{m-1}$. Clearly $\preceq$ is a well-partial-ordering with the unique LCM property. One group reduction ordering $\leq$ in this case is given by $a^0 < a^1 < a^2 < \ldots < a^{m-1}$.
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 $G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \ldots \triangleright G_{e+1} = 1$ be a subnormal series of $G$. From here on in this section, $i$ is an integer and $1 \leq i \leq e$. For each $i$, let $Y_i$ be a set of coset representatives of $G_{i+1}$ in $G_i$ and let $Y$ be $\bigcup_{i=1}^{e} Y_i$. For all $x$ in $G$, $x$ can be written uniquely as $y_e y_{e-1} \ldots y_2 y_1$ where each $y_i$ is in $Y_i$. In fact, the map from $G$ to $Y_e \times \ldots \times Y_2 \times Y_1$ taking $x$ to $(y_e, y_{e-1}, \ldots, y_2, y_1)$ is a bijection. We call $y_e y_{e-1} \ldots y_2 y_1$ the *$Y$-form* of $x$. Let $H_i$ be the quotient $G_i/G_{i+1}$ and $\pi_i : G_i \to H_i$ be the quotient map. The map $\pi_i$ gives a one-to-one correspondence between $Y_i$ and $H_i$. 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 $y_e y_{e-1} \ldots y_2 y_1$ and $y'_e y'_{e-1} \ldots y'_2 y'_1$, respectively. Define $x \preceq x'$ if $\pi_i(y_i) \preceq_i \pi_i(y'_i)$ for each $i$. Therefore 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-poset 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:** 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 4.4.1** *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 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 $\{y \in G_2 \mid yy_1 \in X\}$. Then $X'$ is non-empty. By the induction hypothesis the standard total ordering on $G_2$ relative to the subnormal series $G_2 \triangleright G_3 \triangleright \ldots \triangleright G_{e+1} = 1$, the set $\bigcup_{i=2}^{e} Y_i$ and the orderings $\preceq_i$ and $\leq_i$ is a well-ordering. 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 $y_2y_1$, $y_2'y_1'$ and $y_2''y_1''$, 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 $y_2 <_2 y_2'$. In the first case, we have $\pi(x) <_1 \pi(x')$ and $\pi(x') \preceq_1 \pi(x'')$. 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 $y_2 <_2 y_2'$. So $x\frac{x''}{x'} = x(x')^{-1}x'' = y_2y_1(y_1')^{-1}(y_2')^{-1}y_2''y_1'' = y_2(y_2')^{-1}y_2''y_1'' = y_2\frac{y_2''}{y_2'}y_1''$. If the $Y$-form of $x\frac{x''}{x'}$ is $y_2^*y_1^*$, then $y_1^* = y_1''$ and $y_2^* = y_2\frac{y_2''}{y_2'}$. So $\pi(x\frac{x''}{x'}) = \pi(y_1'') = \pi(x'')$. Since $y_2 <_2 y_2'$, $y_2' \preceq_2 y_2''$ and the total ordering on $H_2$ is a group reduction ordering, $y_2\frac{y_2''}{y_2'} <_2 y_2''$. Therefore $y_2^* <_2 y_2''$ 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 $\preceq$, respectively. By the result for $e = 2$, $G$ is a reduction group with the standard partial ordering $\leq$ and the standard total ordering $\preceq$. $\square$

We remark here that if for each $i$, $\leq_i$ 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 in the examples of group reduction orderings we saw previously in this section, the total orderings are linear extensions of the partial orderings. Even though we do not assume that the total ordering is a linear extension of the partial ordering in the definition of a group reduction ordering, most of the time this is the case.

**Corollary 4.4.2** *If $G$ is polycyclic-by-finite, then there exists a group reduction ordering on $G$.*

**Proof:** The examples earlier in this section illustrated group reduction orderings on all finite groups and the infinite cyclic group. 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 4.1, so is $G$. $\square$

In Example 3, we saw a way to construct a group reduction ordering on a finite group $G$. The underlying poset structure on $G$ is just the discrete poset $\Phi_{|G|}$. For large $|G|$, this approach is not very desirable because we cannot perform many reductions on elements in $\mathbb{Z}[G]$. If $f_1$ and $f_2$ are in $\mathbb{Z}[G]$, then we can reduce $f_1$ by $f_2$ only when the leading monomial of $f_2$ precedes or is equal to that of $f_1$ in the partial ordering $\preceq$. We cannot perform a lot of reductions if no element in $G$ is preceded by another in $\preceq$, as in the case of a discrete poset. In general, we would like to have many elements related to each other by the partial ordering.

Let $H$ be a finite index subgroup of $G$. Assume that $H$ is a reduction group with partial ordering $\preceq_H$ and total ordering $\leq_H$. Let $X$ be a set of left coset representatives of $H$ in $G$. Every element in $G$ is of the form $xy$ for some $x$ in $X$ and $y$ in $H$. Let $x_1, x_2$ be in $X$ and $y_1, y_2$ be in $H$. Extend the partial ordering to $G$ by defining $x_1y_1 \preceq x_2y_2$ if $x_1 = x_2$ and $y_1 \preceq_H y_2$ in $H$. The poset structure on $G$ is just $\Phi_m \times H$ and therefore $\preceq$ is a well-partial-ordering with the unique LCM property on $G$. Fix an arbitrary total ordering $\leq_X$ on $X$. Define a total ordering on $G$ by $x_1y_1 < x_2y_2$ if $x_1 <_X x_2$, or if $x_1 = x_2$ and $y_1 <_H y_2$.

**Theorem 4.4.3** *The total ordering on $G$ described above is a group reduction ordering with respect to $\preceq$.*

**Proof:** It is easy to see that $\leq$ is a well-ordering. A proof similar to the one given for condition 2 in Theorem 4.1 will work. Assume that $x_1y_1 \preceq x_2y_2$. Notice that $\frac{x_2y_2}{x_1y_1} = y_1^{-1}x_1^{-1}x_2y_2 = \frac{y_2}{y_1}$ since $x_1 = x_2$. Let $x$ be in $X$ and $y$ be in $H$. If $xy < x_1y_1$, then either $x <_X x_1 = x_2$, in which case $xy\frac{x_2y_2}{x_1y_1} = xy\frac{y_2}{y_1} < x_2y_2$, or $x = x_1$ and $y <_H y_1$, in which case we can use the fact that $\leq$ is a group reduction ordering on $H$ to conclude that $y\frac{y_2}{y_1} <_H y_2$ and $xy\frac{x_2y_2}{x_1y_1} = x_2y\frac{y_2}{y_1} < x_2y_2$. $\square$

Polycyclic groups have a subnormal series consisting of only cyclic quotients. The well-poset structure of the infinite cyclic group illustrated earlier has every element related to infinitely many others by the partial ordering. The well-poset structure of a finite cyclic group illustrated earlier has every element related to every other by the partial ordering. We expect to be able to perform a lot of reductions in the group ring of a polycyclic group. If the group is known to have a nonabelian finite simple group $G$ as one of its quotients in a subnormal series, then we can try to find a low index polycyclic subgroup $H$ in $G$. Impose on $H$ the poset structure of $\Phi_{[G:H]} \times H$ as described above. By Theorem 4.3 we get a group reduction ordering on $G$. This allows us to perform more reductions than before.

## 4.5 Reductions in Integral Group Rings

In this section we will define the notion of a reduction and eventually a Gröbner basis of a submodule of $\mathbb{Z}[G]^k$ where $G$ is a reduction group. Recall that $\mathcal{T}$ is the set of all term vectors in $\mathbb{Z}[G]^k$. 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 $\mathcal{T}$ and $\mathcal{M}$ by $\preceq$ and $\leq$, respectively.

Every nonzero $f$ in $\mathbb{Z}[G]^k$ is of the form $c_1t_1 + c_2t_2 + \ldots + c_rt_r$, where $r$ is a positive integer, $c_1, c_2, \ldots, c_r$ are nonzero integers and $t_1 > t_2 > \ldots > t_r$ are term vectors. This expression will be called the *standard form* of $f$. We define $\mathrm{lc}(f) = c_1$ to be the *leading coefficient* of $f$, $\mathrm{lt}(f) = t_1$ to be the *leading term vector* of $f$ and $\mathrm{lm}(f) = c_1t_1$ to be the *leading monomial vector* of $f$.

**Definition:** 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 $\operatorname{lt}(g) \preceq \operatorname{lt}(g)x$. 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 $t < \operatorname{lt}(g)$, then $tx < \operatorname{lt}(g)x$ since $\leq$ is a group reduction ordering. In particular, all term vectors which appear in the standard form of $gx$ are no bigger than $\operatorname{lt}(g)x$ in $\leq$. So $\operatorname{lt}(gx) = \operatorname{lt}(g)x$, $\operatorname{lm}(gx) = \operatorname{lm}(g)x$ and $\operatorname{lc}(gx) = \operatorname{lc}(g)$.

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

$$g_2 = g_1 - \lfloor \frac{\operatorname{lc}(g_1)}{\operatorname{lc}(f)} \rfloor fx,$$

where $x$ is the unique element in $G$ with $\operatorname{lt}(f)x = \operatorname{lt}(g_1)$. 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 $t \leq \operatorname{lt}(f)$. Since $\operatorname{lt}(f) \preceq \operatorname{lt}(g_1) = \operatorname{lt}(f)x$, $fx$ is a proper product and thus $\operatorname{lt}(g_2) \leq \operatorname{lt}(g_1)$. Second, since $\operatorname{lm}(g_1) \succeq \operatorname{lm}(f)$, either $\operatorname{lc}(g_1) = \operatorname{lc}(f)$ or $\operatorname{lc}(g_1) \gg \operatorname{lc}(f)$. In both cases $\lfloor \frac{\operatorname{lc}(g_1)}{\operatorname{lc}(f)} \rfloor \neq 0$. So $g_1 \neq g_2$. In fact, if $\operatorname{lt}(g_2) = \operatorname{lt}(g_1)$, then $\operatorname{lc}(g_2) \ll \operatorname{lc}(g_1)$. Therefore $\operatorname{lm}(g_2) < \operatorname{lm}(g_1)$ whenever $g_1 \to_P g_2$.

For nonzero polynomial arrays $g_1$ and $g$, define $g_1 \xrightarrow{*}_P g$ if $g_1 = g$ or if there exists a succession of $s \geq 1$ reduction(s) $g_1 \to_P g_2$, $g_2 \to_P g_3, \ldots, g_s \to_P g_{s+1} = g$. 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 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. Adding all these equations up, we have $g_1 = g + a_1 h_1 x_1 + a_2 h_2 x_2 + \ldots + a_s h_s x_s$. In addition, $\mathrm{lt}(h_1 x_1), \mathrm{lt}(h_2 x_2), \ldots, \mathrm{lt}(h_s x_s)$ are all $\leq \mathrm{lt}(g_1)$. 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 $g_1 \to_P g_2$, $g_2 \to_P g_3, \ldots$, must stop eventually. If not, then $\mathrm{lm}(g_1) > \mathrm{lm}(g_2) > \ldots$ is an infinite sequence of strictly decreasing monomials, and this is not possible since $\leq$ is a well-ordering on $\mathcal{M}$.

**Definition:** Let $\mathcal{I}$ be a nonzero submodule of $\mathbb{Z}[G]^k$. Let $\mathcal{M}_\mathcal{I}$ be $\min(\{\mathrm{lm}(f) \in \mathcal{M} \mid f \in \mathcal{I}, f \neq 0\})$ with respect to the partial ordering $\preceq$. For each $m$ in $\mathcal{M}_\mathcal{I}$, let $f_m$ be a polynomial array in $\mathcal{I}$ with $\mathrm{lm}(f_m) = m$. The set of all such $f_m$'s, denoted by $\mathcal{B}$, is a *minimal Gröbner basis* of $\mathcal{I}$. A *Gröbner basis* is a subset of $\mathcal{I}$ which contains a minimal Gröbner basis.

By Proposition 2.4, a minimal Gröbner basis of a nonzero submodule is nonempty and finite.

**Theorem 4.5.1** *Let $\mathcal{B}$ be a Gröbner basis of $\mathcal{I}$. For a polynomial array $g$ in $\mathbb{Z}[G]^k$, $g$ is in $\mathcal{I}$ if and only if $g \xrightarrow{*}_\mathcal{B} 0$.*

**Proof:** If $g \xrightarrow{*}_\mathcal{B} 0$, 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 $g'$ which cannot be reduced any more by $\mathcal{B}$. If $g' \neq 0$, then by Proposition 2.4 and the fact that $\mathcal{B}$ is a Gröbner basis, $\mathrm{lm}(g') \succeq \mathrm{lm}(f)$ for some $f$ in $\mathcal{B}$. Therefore $g'$ can be reduced by $\mathcal{B}$. This is a contradiction. So $g \xrightarrow{*}_\mathcal{B} 0$. $\square$

**Corollary 4.5.2** *Let $\mathcal{B}$ be a Gröbner basis of $\mathcal{I}$. Then $\mathcal{B}$ generates $\mathcal{I}$ as a right $\mathbb{Z}[G]$-module.*

**Corollary 4.5.3** *Every right $\mathbb{Z}[G]$-submodule of $\mathbb{Z}[G]^k$ is finitely generated as a right module. In other words, $\mathbb{Z}[G]$ is right Noetherian.*

**Theorem 4.5.4** *If $G$ is a group and $\mathbb{Z}[G]$ is right Noetherian, then every subgroup of $G$ is finitely generated.*

In Chapter 10 of [Pas77], it was proved that if $K$ is a field and $K[G]$ is right Noetherian, then every subgroup of $G$ is finitely generated. Therefore Theorem 5.4 follows from the fact that $\mathbb{Q}[G]$ is a localization of $\mathbb{Z}[G]$. For completeness, a proof of Theorem 5.4 similar to the proof in [Pas77] is given.

**Proof:** Let us prove this theorem by contradiction. Suppose $x_1, x_2, \ldots$ is an infinite sequence of elements in $G$ such that if $H_i$ is the subgroup generated by $x_1, x_2, \ldots, x_i$ for $i \geq 1$, then $H_{i+1}$ contains $H_i$ properly. Let $i$ be a positive integer. Define $\mathcal{I}_i$ to be the right ideal of $\mathbb{Z}[G]$ generated by $\{h-1 \mid h \in H_i\}$. Let $\pi_i$ be the unique $\mathbb{Z}$-linear map from $\mathbb{Z}[G]$ to $\mathbb{Z}$ such that $\pi_i(h) = 1$ for each $h$ in $H_i$ and $\pi_i(g) = 0$ for each $g$ not in $H_i$. Any element in $\mathcal{I}_i$ can be written in the form $a_1(h_1 - 1)g_1 + a_2(h_2 - 1)g_2 + \ldots + a_s(h_s - 1)g_s$ where $a_1, a_2, \ldots, a_s$ are integers, $h_1, h_2, \ldots, h_s$ are elements in $H_i$ and $g_1, g_2, \ldots, g_s$ are elements in $G$. Let $j$ be an integer and $1 \leq j \leq s$. Now $\pi_i(a_j(h_j - 1)g_j) = a_j(\pi_i(h_j g_j) - \pi_i(g_j))$. Since $h_j$ is in $H_i$, $h_j g_j$ is in $H_i$ if and only if $g_j$ is. As a result, $\pi_i(h_j g_j) = \pi_i(g_j)$. Therefore $\pi_i(a_1(h_1 - 1)g_1 + a_2(h_2 - 1)g_2 + \ldots + a_s(h_s - 1)g_s) = 0$ and $\pi_i(\mathcal{I}_i) = \{0\}$. Clearly $\pi_i(x_{i+1} - 1) = -1$. The element $x_{i+1} - 1$ is in $\mathcal{I}_{i+1}$ but not in $\mathcal{I}_i$. So $\{\mathcal{I}_i\}_{i \geq 1}$ gives a chain of right ideals with $\mathcal{I}_{i+1}$ containing $\mathcal{I}_i$ properly for $i \geq 1$. This is a contradiction. $\square$

**Corollary 4.5.5** *If $G$ is a reduction group, then every subgroup of $G$ is finitely generated.*

## 4.6 Finding Gröbner Bases

Suppose that $P = \{f_1, f_2, \ldots, f_s\}$ 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. Let $i$ and $j$ be integers and $1 \leq i < j \leq s$. Denote $\mathrm{lt}(f_i)$ by $t_i$ and $\mathrm{lt}(f_j)$ 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 $t' \succeq t_i$ and $t' \succeq t_j$. Suppose that $f_i$ and $f_j$ are aligned. Let $t$ be $\mathrm{lcm}(t_i, t_j)$ which exists and is unique since $\preceq$ has the unique LCM property. Let $a_i$ be $\mathrm{lc}(f_i)$, $a_j$ be $\mathrm{lc}(f_j)$, and $p$ and $q$ be integers such that $pa_i + qa_j = \gcd(a_i, a_j)$. Let $a$ be $\mathrm{lcm}(a_i, a_j)$. Define $\alpha_{ij}$ to be $pf_i\frac{t}{t_i} + qf_j\frac{t}{t_j}$ and $\beta_{ij}$ to be $\frac{a}{a_i}f_i\frac{t}{t_i} - \frac{a}{a_j}f_j\frac{t}{t_j}$.

**Definition:** Suppose that $P$ 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 $h_1m_1 + h_2m_2 + \ldots + h_sm_s$ where for each $i$ with $1 \leq i \leq s$, $h_im_i$ is a proper $P$-product and $\mathrm{lt}(h_im_i) < t$.
2. We say that $g$ is *lower$^*_P$-$t$* if $g$ can be written as $h_1m_1 + h_2m_2 + \ldots + h_sm_s$ where for each $i$ with $1 \leq i \leq s$, $h_im_i$ is a proper $P$-product and $\mathrm{lt}(h_im_i) \leq t$.
3. We say that $g$ can be *$P$-weakly reduced* if $g = h_1m_1 + h_2m_2 + \ldots + h_sm_s + h$ where $h$ is lower$_P$-$\mathrm{lt}(g)$ and for each $i$ with $1 \leq i \leq s$, $h_im_i$ is a proper $P$-product, $\mathrm{lt}(h_im_i) = \mathrm{lt}(g)$ and $\mathrm{lc}(h_im_i)$ has the same sign as $\mathrm{lc}(g)$.

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

**Lemma 4.6.1** *Let $P$ be a set of nonzero polynomial arrays in $\mathbb{Z}[G]^k$. If $g \xrightarrow{*}_P 0$ and $lc(g) > 0$, then $g$ can be weakly reduced.*

**Proof:** Suppose $g \xrightarrow{*}_P 0$. Let $g_1 = g$ and the chain of reductions performed consists of $g_2 = g_1 - h_1m_1$, $g_3 = g_2 - h_2m_2, \ldots, 0 = g_{s+1} = g_s - h_sm_s$ and $g_s \neq 0$. For all $i$ with $1 \leq i \leq s$, $h_im_i$ is a proper product. Now $g = h_1m_1 + h_2m_2 + \ldots + h_sm_s$. Since $\mathrm{lt}(g_i) = \mathrm{lt}(h_im_i)$ for each $i$ and $\mathrm{lm}(g_1) > \mathrm{lm}(g_2) > \ldots > \mathrm{lm}(g_s)$, it follows that $\mathrm{lt}(g_1) \geq \mathrm{lt}(g_2) \geq \ldots \geq \mathrm{lt}(g_s)$ and thus $\mathrm{lt}(g) \geq \mathrm{lt}(h_im_i)$ for each $i$. We are given that

$\mathrm{lc}(g) > 0$. Let $i$ be maximal such that $\mathrm{lt}(g_i) = \mathrm{lt}(g)$. Since $\mathrm{lm}(g_1) > \mathrm{lm}(g_2) > \ldots > \mathrm{lm}(g_i)$, $\mathrm{lc}(g_1) \gg \mathrm{lc}(g_2) \gg \ldots \gg \mathrm{lc}(g_i)$. But $\mathrm{lc}(g_1) = \mathrm{lc}(g) > 0$. So $\mathrm{lc}(g_1) > \mathrm{lc}(g_2) > \ldots \mathrm{lc}(g_i) > 0$ by the definition of $\ll$. Therefore $\mathrm{lc}(h_1m_1), \mathrm{lc}(h_2m_2), \ldots, \mathrm{lc}(h_im_i)$ are all positive. This shows that $g$ can be weakly reduced. □

**Definition:** A set of nonzero polynomial arrays $P$ is said to be *symmetrized* if for any $f$ in $P$, $\mathrm{lc}(f) > 0$ and for any $x$ in $G$, $fx$ is $\mathrm{lower}^*_P$-$\mathrm{lt}(fx)$.

**Lemma 4.6.2** *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 that $t$ 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$^*$-$t$ and $x$ is aligned with $t$, then $agx$ is lower$^*$-$tx$.*
3. *If $g$ can be weakly reduced and $x$ is aligned with $g$, $agx$ can be weakly reduced.*

**Proof:**

1. Assume that $g$ is lower-$t$ for some term vector $t$, then $g = h_1m_1 + h_2m_2 + \ldots + h_sm_s$ where for each $i$ with $1 \le i \le s$, $h_im_i$ is a proper product and $\mathrm{lt}(h_im_i) < t$. We may assume that $a = 1$. Now $gx = h_1m_1x + h_2m_2x + \ldots + h_sm_sx$. Let $i$ be an integer with $1 \le i \le s$. It suffices to show that $h_im_ix$ is lower-$tx$. Since $x$ is aligned with $t$, $t \preceq tx$. Any term vector $t'$ which appears in the standard form of $h_im_i$ satisfies $t' \le \mathrm{lt}(h_im_i) < t$. So $t'x < tx$ and $\mathrm{lt}(h_im_ix) < tx$. Since $P$ is symmetrized, $h_1m_1x$ is $\mathrm{lower}^*$-$\mathrm{lt}(h_1m_1x)$. Therefore $h_1m_1x$ is lower-$tx$.
2. The proof of this part is similar to the proof of part 1.
3. Assume that $g$ can be weakly reduced. Therefore $g = h_1m_1 + h_2m_2 + \ldots + h_sm_s + h$ where $h$ is lower-$\mathrm{lt}(g)$ and for each $i$ with $1 \le i \le s$, $h_im_i$ is a proper product, $\mathrm{lt}(h_im_i) = \mathrm{lt}(g)$ and $\mathrm{lc}(h_im_i)$ has the same sign as $\mathrm{lc}(g)$. Again, we may assume that $a = 1$. Since $x$ is aligned with $g$, $\mathrm{lt}(g) \preceq \mathrm{lt}(g)x$. If $t$ is a term vector such that $t < \mathrm{lt}(g)$, then $tx < \mathrm{lt}(g)x = \mathrm{lt}(gx)$. This shows that for every $i$ with $1 \le i \le s$, $\mathrm{lt}(h_im_ix) = \mathrm{lt}(h_im_i)x = \mathrm{lt}(g)x$. By part 1, $hx$ is lower-$\mathrm{lt}(gx)$. Since

$gx = h_1m_1x + h_2m_2x + \ldots + h_sm_sx + hx$, $gx$ can be weakly reduced. □

**Theorem 4.6.3** *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. We will start by finding a set $P'$ which we will prove is a minimal Gröbner basis. Let $\mathcal{M}_P$ be the set of all leading monomial vectors of the polynomial arrays in $P$. For any leading monomial vector $m$ in $\min(\mathcal{M}_P)$, let $f_m$ be a polynomial array in $P$ such that $\mathrm{lm}(f_m) = m$. Let $P'$ be the collection of all such $f_m$'s.

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 $h_1m_1 + h_2m_2 + \ldots + h_sm_s + h$, where $s \geq 1$, $h$ is lower-lt($h_1m_1$) and for $1 \leq i \leq s$, $h_im_i$ is a proper product and $\mathrm{lt}(h_im_i) = \mathrm{lt}(h_1m_1)$. 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 $\mathrm{lt}(h_1m_1)$ and an integer parameter $|\mathrm{lc}(h_1m_1)| + |\mathrm{lc}(h_2m_2)| + \ldots + |\mathrm{lc}(h_sm_s)|$. 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 $h_im_i$'s all have the same sign. If this is true, then $\mathrm{lt}(h_1) \preceq \mathrm{lt}(h_1m_1) = \mathrm{lt}(f)$. It is not hard to see that either $\mathrm{lc}(f) \gg \mathrm{lc}(h_1)$ or $\mathrm{lc}(f) = \mathrm{lc}(h_1)$. So $\mathrm{lm}(h_1) \preceq \mathrm{lm}(f)$ and by the definition of $P'$, there is a polynomial array $f_m$ in $P'$ such that $\mathrm{lm}(f_m) \preceq \mathrm{lm}(f)$. Therefore $P'$ is a minimal Gröbner basis.

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

$$\begin{bmatrix} p & q \\ \frac{d}{\mathrm{lc}(f_i)} & \frac{-d}{\mathrm{lc}(f_j)} \end{bmatrix} \begin{bmatrix} f_i x_i \\ f_j x_j \end{bmatrix} = \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix}.$$

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

$$\begin{bmatrix} f_i x_i \\ f_j x_j \end{bmatrix} = \begin{bmatrix} \frac{d}{\mathrm{lc}(f_j)} & q \\ \frac{d}{\mathrm{lc}(f_i)} & -p \end{bmatrix} \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix}.$$

Therefore any integer linear combination of $f_i x_i$ and $f_j x_j$ can be written as an integer linear combination of $\alpha_{ij}$ and $\beta_{ij}$.

Let $x$ be the group element in $G$ such that $tx = \mathrm{lt}(h_1 m_1)$. Now $h_1 m_1$ is an integer multiple of $f_i x_i x$ and $h_2 m_2$ is an integer multiple of $f_j x_j x$. So $h_1 m_1 + h_2 m_2 = p' \alpha_{ij} x + q' \beta_{ij} x$ for some integers $p'$ and $q'$. Since $t \preceq \mathrm{lt}(h_1 m_1) = tx$, $\mathrm{lt}(\alpha_{ij} x) = tx$ and $\mathrm{lt}(\beta_{ij} x) < tx$. Since $\alpha_{ij} x$ can be weakly reduced and $\beta_{ij} x$ is lower-$tx$, $h_1 m_1 + h_2 m_2 = h'_1 m'_1 + h'_2 m'_2 + \ldots + h'_{s'} m'_{s'} + h'$ where $s'$ is nonnegative, $h'$ is lower-$tx$ and for $1 \leq i \leq s'$, $h'_i m'_i$ is a proper product with leading term vector $tx$ and leading coefficient the same sign as $\mathrm{lc}(h_1 m_1 + h_2 m_2)$. The case $s' = 0$ occurs when $p' = 0$. As a result, $h'_1 m'_1 + h'_2 m'_2 + \ldots + h'_{s'} m'_{s'} + h_3 m_3 + h_4 m_4 + \ldots + h_s m_s + (h + h')$ is another proper sum for $f$ with the term vector parameter smaller than or equal to the term vector parameter in the proper sum $h_1 m_1 + h_2 m_2 + \ldots + h_s m_s + h$. 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

$$\begin{aligned} & |\mathrm{lc}(h'_1 m'_1)| + \ldots + |\mathrm{lc}(h'_{s'} m'_{s'})| + |\mathrm{lc}(h_3 m_3)| + |\mathrm{lc}(h_4 m_4)| + \ldots + |\mathrm{lc}(h_s m_s)| \\ = \; & |\mathrm{lc}(h_1 m_1 + h_2 m_2)| + |\mathrm{lc}(h_3 m_3)| + |\mathrm{lc}(h_4 m_4)| + \ldots + |\mathrm{lc}(h_s m_s)| \\ < \; & |\mathrm{lc}(h_1 m_1)| + |\mathrm{lc}(h_2 m_2)| + \ldots + |\mathrm{lc}(h_s m_s)|. \end{aligned}$$

Therefore in the proper sum $h_1 m_1 + h_2 m_2 + \ldots + h_s m_s + h$, the integer parameter is not minimal. We have arrived at a contradiction. This means that the leading coefficients of $h_1m_1$ and $h_2m_2$ must have the same sign. Therefore all the leading coefficients of $h_1m_1, h_2m_2, \ldots, h_sm_s$ are of the same sign. This finishes the proof. $\square$

## 4.7 Symmetrization

One of the hypotheses in Theorem 6.3 is that $P$ is a symmetrized set of polynomial arrays. By definition, for any $f$ in $P$ and $x$ in $G$, $fx$ can be written as a sum of proper products each with leading term vector no bigger than $\mathrm{lt}(fx)$ in the ordering $\leq$. 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$^*$-$\mathrm{lt}(fx)$. 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.

We will follow closely the notation established in the previous section. Assume that we have a reduction ordering set in $\mathbb{Z}[G]^k$. Let $\mathcal{S}$ be a subset of $\mathcal{T}$. For each $s$ in $\mathcal{S}$, define $\mathcal{U}_G(s, \mathcal{S})$ to be $\min(\{sx \mid x \in G \text{ and } sx \geq tx \, \forall \, t \in \mathcal{S}\})$ and $\mathcal{V}_G(s, \mathcal{S})$ to be $\{x \in G \mid sx \in \mathcal{U}(s, \mathcal{S})\}$. The subscript $G$ will be dropped if it is clear what the group is from the context. By Proposition 2.4, the sets $\mathcal{U}(s, \mathcal{S})$ and $\mathcal{V}(s, \mathcal{S})$ are finite. In addition, if $s$ is in $S$ and if there exists $x$ in $G$ such that $sx \geq tx$ for every $t$ in $\mathcal{S}$, then there exists an element $sx'$ in $\mathcal{U}(s, \mathcal{S})$ such that $sx \succeq sx'$. 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 $\mathrm{Symm}_G(f)$ be $\{fx \mid x \in \mathcal{V}_G(s, T_f) \text{ for some } s \in T_f\}$. Again, if we know what the group $G$ is from the context, the subscript $G$ will be dropped.

**Theorem 4.7.1** *Assume that $P$ is a set of nonzero polynomial arrays. If for every $f$ in $P$, every element $fx$ in Symm($f$) is lower$^*$-lt($fx$), then $P$ is a symmetrized set.*

**Proof:** We will prove this theorem by contradiction. Assume that there exists a polynomial array $f$ in $P$ and an element $x$ in $G$ such that $fx$ is not lower$^*$-$\mathrm{lt}(fx)$. Let us call such an $fx$ a counterexample. Since $\leq$ is a well-ordering on $\mathcal{T}$, we can pick $f$ and $x$ such that among the counterexamples, $\mathrm{lt}(fx)$ is minimal in $\leq$. Let the standard form of $f$ be $c_1t_1 + c_2t_2 + \ldots + c_rt_r$. Therefore $T_f = \{t_1, t_2, \ldots, t_r\}$. Assume that $\mathrm{lt}(fx) = t_ix$. So $t_ix \geq t_jx$ for $1 \leq j \leq r$. By definition of $\mathcal{U}(t_i, T_f)$, there exists an element $x'$ in $G$ such that $t_ix' \preceq t_ix$ and $t_ix'$ is contained in $\mathcal{U}(t_i, T_f)$. So $fx'$ is in $\mathrm{Symm}(f)$ and is lower*-$\mathrm{lt}(fx')$. We can write $fx'$ as $h_1m_1 + h_2m_2 + \ldots + h_sm_s$ such that for $1 \leq j \leq s$, $h_jm_j$ is a proper product and $\mathrm{lt}(h_jm_j) \leq \mathrm{lt}(fx') = t_ix'$. Now

$$
\begin{aligned}
fx &= (h_1m_1 + h_2m_2 + \ldots + h_sm_s)\frac{t_ix}{t_ix'} \\
&= h_1m_1\frac{t_ix}{t_ix'} + h_2m_2\frac{t_ix}{t_ix'} + \ldots + h_sm_s\frac{t_ix}{t_ix'}.
\end{aligned}
$$

It suffices to show that for $1 \leq j \leq s$, $h_jm_j\frac{t_ix}{t_ix'}$ is lower*-$t_ix$.

Let $j$ be an integer with $1 \leq j \leq s$. Every term vector in $T_{h_jm_j}$ is no bigger than $t_ix'$ since $\mathrm{lt}(h_jm_j) \leq t_ix'$. Let the monomial $m_j$ be $cx_j$, where $c$ is a nonzero integer and $x_j$ is a group element. In the case when $\mathrm{lt}(h_jm_j) = t_ix'$, since $h_jm_j$ is a proper product, $t_ix' = \mathrm{lt}(h_jm_j) = \mathrm{lt}(h_j)x_j \succeq \mathrm{lt}(h_j)$. As a result, $\mathrm{lt}(h_j) \preceq t_ix' \preceq t_ix = \mathrm{lt}(h_j)x_j\frac{t_ix}{t_ix'}$. Therefore $h_j(m_j\frac{t_ix}{t_ix'})$ is a proper product with leading term vector $t_ix$. So $h_j(m_j\frac{t_ix}{t_ix'})$ is lower*-$t_ix$. In the case when $\mathrm{lt}(h_jm_j) < t_ix'$, every term vector in $T_{h_jm_j}$ is smaller than $t_ix'$ in $\leq$. Therefore $\mathrm{lt}(h_jm_j\frac{t_ix}{t_ix'}) < t_ix$. Here we use the minimality of $\mathrm{lt}(fx) = t_ix$ to conclude that $h_j(m_j\frac{t_ix}{t_ix'})$ is not a counterexample. So $h_j(m_j\frac{t_ix}{t_ix'})$ is lower-$t_ix$. In both cases, $h_jm_j\frac{t_ix}{t_ix'}$ is lower*-$t_ix$. Therefore $fx$ is lower*-$\mathrm{lt}(fx)$ and this is a contradiction. □

By Theorem 7.1, to show that $P$ is a symmetrized set, it suffices to show that for each $f$ in $P$, each polynomial array in $\mathrm{Symm}(f)$ can be reduced to 0. We claim that the set $\mathrm{Symm}(f)$ is finite. This is because $\mathcal{V}(s, T_f)$ is finite for every $s$ in $T_f$ and $T_f$ is finite. Moreover,

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

To prove this, it suffices to show that the sets $\mathcal{V}(s, T_f)$ where $s$ is in $T_f$ are disjoint. This is clear because if $x$ is contained in both $\mathcal{V}(s, T_f)$ and $\mathcal{V}(s', T_f)$, 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 $\mathrm{Symm}(f)$ for a given nonzero $f$ in $\mathbb{Z}[G]^k$? Usually, the set $\mathrm{Symm}(f)$ depends on the partial ordering $\preceq$ and the total ordering $\leq$ on $G$. In the remaining part of this section, we will look at $\mathrm{Symm}(f)$ for $f$ in $\mathbb{Z}[G]^1$ when $G$ is cyclic or finite. We will view term vectors in $\mathbb{Z}[G]^1$ as group elements.

Let us first look at the case when $G$ is the infinite cyclic group $\mathbb{Z}$ generated by $a$ with the partial ordering given by $a^0 \prec a^1 \prec a^2 \prec \ldots$ and $a^0 \prec a^{-1} \prec a^{-2} \prec \ldots$, and the total ordering given by $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \ldots$. Any nonzero element $f$ of $\mathbb{Z}[G]$ can be written uniquely as a sum $c_1a^{\lambda_1} + c_2a^{\lambda_2} + \ldots + c_sa^{\lambda_s}$, where $s \geq 1$, $c_1, c_2, \ldots, c_s$ are nonzero integers and $\lambda_1 > \lambda_2 > \ldots > \lambda_s$ are integers in the usual ordering. If $s = 1$, then for each $x$ in $G$, $fx = c_1a^{\lambda}$ for some integer $\lambda$. In this case, $\mathcal{U}(a^{\lambda}, T_f) = \min(\{a^{\lambda} \mid \lambda \in \mathbb{Z}\}) = \{a^0\}$ and $\mathcal{V}(a^{\lambda}, T_f) = \{a^{-\lambda}\}$. Therefore, $\mathrm{Symm}(f)$ has only the monomial vector $c_1a^0$ in it. Now assume that $s > 1$. Let $\lambda$ be an integer. Any polynomial array $fa^{\lambda}$ is of the form $c_1a^{\lambda_1+\lambda} + c_2a^{\lambda_2+\lambda} + \ldots + c_sa^{\lambda_s+\lambda}$. It is not hard to see that for $2 \leq i \leq s-1$, $a^{\lambda_i+\lambda} < a^{\lambda_1+\lambda}$ or $a^{\lambda_i+\lambda} < a^{\lambda_s+\lambda}$ in the total ordering on $G$. Therefore the sets $\mathcal{U}(a^{\lambda_i}, T_f)$ and $\mathcal{V}(a^{\lambda_i}, T_f)$ are empty if $2 \leq i \leq s-1$. We need to find $\mathcal{V}(a^{\lambda_1}, T_f)$ and $\mathcal{V}(a^{\lambda_s}, T_f)$. In $G$, $a^{\lambda_1+\lambda} \geq a^{\lambda_s+\lambda}$ if and only if $\lambda_1 + \lambda + \lambda_s + \lambda \geq 1$, which is true as long as $\lambda \geq \lceil \frac{1-\lambda_1-\lambda_s}{2} \rceil$. Let $\mu_1 = \lceil \frac{1-\lambda_1-\lambda_s}{2} \rceil$. Therefore $\mathcal{U}(a^{\lambda_1}, T_f) = \min(\{a^{\lambda_1+\lambda} \mid \lambda \geq \mu_1\})$. Routine calculation shows that this set is just $\{a^{\lambda_1+\mu_1}\}$. So $\mathcal{V}(a^{\lambda_1}, T_f) = \{a^{\mu_1}\}$. On the other hand, $a^{\lambda_1+\lambda} \leq a^{\lambda_s+\lambda}$ if and only if $\lambda_1 + \lambda + \lambda_s + \lambda \leq 0$, which is true as long as $\lambda \leq \lfloor \frac{-\lambda_1-\lambda_s}{2} \rfloor$. Let $\mu_s = \lfloor \frac{-\lambda_1-\lambda_s}{2} \rfloor$. Therefore $\mathcal{U}(a^{\lambda_s}, T_f) = \min(\{a^{\lambda_s+\lambda} \mid \lambda \leq \mu_s\})$. Routine calculation shows that this set is just $\{a^{\lambda_s+\mu_s}\}$. So $\mathcal{V}(a^{\lambda_s}, T_f) = \{a^{\mu_s}\}$. Summarizing the result for $s \geq 2$, we have $\mathrm{Symm}(f) = \{fa^{\mu_1}, fa^{\mu_s}\}$.

For the second example, 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 $a^0 < a^1 < a^2 < \ldots < a^{-1} < a^{-2} < \ldots$. Let $f$ be a nonzero polynomial array with standard form $c_1a^{\lambda_1} + c_2a^{\lambda_2} + \ldots + c_sa^{\lambda_s}$ as in above. For $s = 1$, $f = c_1a^{\lambda}$ and $\mathrm{Symm}(f) = \{c_1a^0\}$. It is not hard to see that if $2 \leq i \leq s-1$, then $\mathcal{U}(a^{\lambda_i}, T_f)$ is empty. Performing analysis similar to the first example, we get $\mathcal{U}(a^{\lambda_1}, T_f) = \{a^{\lambda_1-\lambda_s}\}$ and $U(a^{\lambda_s}, T_f) = \{a^{-1}\}$. Therefore $\mathrm{Symm}(f) = \{fa^{-\lambda_s}, fa^{-\lambda_s-1}\}$.

Let us take a finite cyclic group as our third example. Suppose that $G$ is generated by $a$ and $G$ has order $m \geq 1$ with partial ordering $a^0 \prec a^1 \prec a^2 \prec \ldots \prec a^{m-1}$ and total ordering $a^0 < a^1 < a^2 < \ldots < a^{m-1}$. Let $f = c_1 a^{\lambda_1} + c_2 a^{\lambda_2} + \ldots + c_s a^{\lambda_s}$, where $s$ is a positive integer, $c_1, c_2, \ldots, c_s$ are nonzero integers and $m > \lambda_1 > \lambda_2 > \ldots > \lambda_s \geq 0$. If $s = 1$, then again $\mathrm{Symm}(f) = \{c_1 a^0\}$. Let us assume that $s \geq 2$. Define $\wedge$ to be addition modulo $m$. In other words, given integers $\lambda, \mu$, we define $\lambda \wedge \mu$ to be the smallest nonnegative number congruent to $\lambda + \mu$ modulo $m$. Since in this example, the partial ordering is a total ordering, for each $i$ with $1 \leq i < m$, $\mathcal{U}(a^{\lambda_i}, T_f)$, which is defined to be a minimal set in this partial ordering, consists of just one element. We claim that $\mathcal{U}(a^{\lambda_1}, T_f) = \{a^{\lambda_1 - \lambda_s}\}$. First, since $a^{\lambda_1 - \lambda_s} = a^{\lambda_1} a^{-\lambda_s}$, we need to show that $a^{\lambda_i} a^{-\lambda_s} \leq a^{\lambda_1 - \lambda_s}$ for $2 \leq i \leq s$. For each $i$ with $2 \leq i \leq s$, $0 \leq \lambda_i - \lambda_s < \lambda_1 - \lambda_s$. So $a^{\lambda_i - \lambda_s} < a^{\lambda_1 - \lambda_s}$. Thus the unique element in $\mathcal{U}(a^{\lambda_1}, T_f)$ is no bigger than $a^{\lambda_1 - \lambda_s}$ in $\leq$. Second, if $\lambda$ is an integer such that $\lambda_1 - \lambda_s > \lambda_1 \wedge \lambda$, we will show that $a^{\lambda_s \wedge \lambda} > a^{\lambda_1 \wedge \lambda}$. This will show that the unique element in $\mathcal{U}(a^{\lambda_1}, T_f)$ has to be at least $a^{\lambda_1 - \lambda_s}$ in $\leq$. Therefore we have $\mathcal{U}(a^{\lambda_1}, T_f) = \{a^{\lambda_1 - \lambda_s}\}$. We may assume that $0 \leq \lambda < m$. If $m - \lambda_s \leq j < m$ or $0 \leq j < m - \lambda_1$, then $\lambda_1 \wedge j \geq \lambda_1 - \lambda_s$. So we may assume that $m - \lambda_1 \leq \lambda < m - \lambda_s$. If this is the case, then $\lambda_1 \wedge \lambda = \lambda_1 + \lambda - m$ and $\lambda_s \wedge \lambda = \lambda_s + \lambda$. So $\lambda_1 \wedge \lambda < \lambda_s \wedge \lambda$. This proves that $a^{\lambda_s \wedge \lambda} > a^{\lambda_1 \wedge \lambda}$. Using a similar argument, it can be proved that for $2 \leq i \leq s$, the unique element in $\mathcal{U}(a^{\lambda_i}, T_f)$ is $a^{\lambda_i \wedge (-\lambda_{i-1})} = a^{\lambda_i - \lambda_{i-1} + m}$. For this example, since $|\mathcal{U}(a^{\lambda_i}, T_f)| = 1$ for $1 \leq i \leq s$, it follows that $1 \leq |\mathrm{Symm}(f)| = s \leq m$.

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

## 4.8 Symmetrization in Polycyclic-By-Finite Groups

In the last half of the previous section, we saw examples of how to find $\mathrm{Symm}(f)$ for $f$ in $\mathbb{Z}[G]$ when $G$ is cyclic or finite. Now we would like to find an algorithm to compute

Symm($f$) for $f$ in $\mathbb{Z}[G]^k$ when $G$ is polycyclic-by-finite. Assume that the standard total ordering defined in Section 3 is used for the total ordering on $G$. We begin with the case $k = 1$.

Let us review some notation first. Assume that $G$ is polycyclic-by-finite and that $G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \ldots \triangleright G_{e+1} = 1$ 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 $\prec_i$ and a group reduction ordering $<_i$ 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 $\leq$ for $G$ with respect to these parameters. By Theorem 4.1, $\leq$ is a group reduction ordering with respect to $\preceq$.

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

For $1 \leq i \leq e$, denote by $\mathcal{T}_i$ the set of all term vectors in $\mathbb{Z}[H_i]$. Denote by $\mathcal{T}'$ the set of all term vectors in $\mathbb{Z}[G_2]$ and by $\mathcal{T}$ 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 $\mathcal{U}_{H_i}(s, \mathcal{S}_i)$ where $S_i$ is a finite subset of $\mathcal{T}_i$ and $s$ is in $\mathcal{S}_i$. We would like to devise a scheme to find $\mathcal{U}_G(s, \mathcal{S})$ for $s$ in a finite subset $\mathcal{S}$ of $\mathcal{T}$. 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 $\mathcal{U}_{G_2}(s', \mathcal{S}')$ for every finite subset $\mathcal{S}'$ of $\mathcal{T}'$ and every element $s'$ in $\mathcal{S}'$. Let $\mathcal{S}$ be the set $\{x_1y_1, x_2y_2, \ldots, x_ry_r\}$ 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 $\mathcal{U}_G(x_1y_1, \mathcal{S}) = \min(\{x_1y_1xy \mid x \in G_2, y \in Y_1 \text{ and } x_1y_1xy > x_jy_jxy \, \forall\, j, 2 \leq j \leq r\})$. By the definition of the standard total ordering, $x_1y_1xy > x_jy_jxy$ if either $\pi(y_1y) >_1 \pi(y_jy)$ in $H$, or $y_1 = y_j$ and $x_1x^{y_1^{-1}} > x_jx^{y_1^{-1}}$ in $G_2$. Let $\mathcal{S}_1$ be the set $\{\pi(y_1), \pi(y_2), \ldots, \pi(y_r)\}$ and $\mathcal{S}'$ be the set $\{x \mid xy_1 \in S\}$. We can find $\mathcal{U}_H(\pi(y_1), \mathcal{S}_1)$ and so we can find $\{y \in Y_1 \mid \pi(y) \in \mathcal{V}_H(\pi(y_1), \mathcal{S}_1)\}$, which we will call $\mathcal{V}_1$. By the induction hypothesis we can find $\mathcal{U}_{G_2}(x_1, \mathcal{S}')$ and so we can find $\mathcal{V}_{G_2}(x_1, \mathcal{S}')$, which we will call $\mathcal{V}'$.

**Proposition 4.8.1** $\mathcal{U}_G(x_1y_1, \mathcal{S}) = \{x_1y_1xy \mid y \in \mathcal{V}_1$ *and* $x^{y_1^{-1}} \in \mathcal{V}'\}$ *and* $\mathcal{V}_G(x_1y_1, \mathcal{S}) =$

$\{xy \mid y \in \mathcal{V}_1 \text{ } and \text{ } x^{y_1^{-1}} \in \mathcal{V}'\}$.

**Proof:** If $x_1y_1xy$ is in $\mathcal{U}_G(x_1y_1, \mathcal{S})$, we want to show that $y$ is in $\mathcal{V}_1$ and $x^{y_1^{-1}}$ is in $\mathcal{V}'$. Since $x_1y_1xy$ is in $\mathcal{U}_G(x_1y_1, \mathcal{S})$, $x_1y_1xy \geq x_iy_ixy$ and $\pi(y_1y) \geq \pi(y_jy)$ for every $i$ with $1 \leq i \leq r$. So there exists $y'$ in $\mathcal{V}_1$ such that $\pi(y_1y') \preceq_1 \pi(y_1y)$. It can be seen that $x_1y_1xy' \geq x_iy_ixy'$ for every $i$ such that $1 \leq i \leq r$. In addition, $x_1y_1xy' \preceq x_1y_1xy$. Therefore, using the definition of $\mathcal{U}_G(x_1y_1, \mathcal{S})$ and the fact that $x_1y_1xy$ is in $\mathcal{U}_G(x_1y_1, \mathcal{S})$, we can conclude that $x_1y_1xy' = x_1y_1xy$ and $y = y'$. Also, since $x_1y_1xy$ is in $\mathcal{U}_G(x_1y_1, \mathcal{S})$, $x_1x^{y_1^{-1}} \geq x_ix^{y_i^{-1}}$ whenever $1 \leq i \leq r$ and $y_1 = y_i$. In other words, $x_1x^{y_1^{-1}} \geq x_ix^{y_1^{-1}}$ whenever $x_i$ is in $\mathcal{S}'$. Therefore there exists $(x')^{y_1^{-1}}$ in $\mathcal{V}'$ such that $x_1(x')^{y_1^{-1}} \preceq x_1x^{y_1^{-1}}$. It can be seen that $x_1y_1x'y \geq x_iy_ix'y$ for $1 \leq i \leq r$. Since $x_1y_1x'y \preceq x_1y_1xy$, $x_1y_1x'y = x_1y_1xy$ and $x = x'$. Therefore $x^{y_1^{-1}}$ is in $\mathcal{V}'$.

Conversely, if $y$ is in $\mathcal{V}_1$ and $x^{y_1^{-1}}$ is in $\mathcal{V}'$, we want to show that $x_1y_1xy$ is in $\mathcal{U}_G(x_1y_1, \mathcal{S})$. It can be seen that $x_1y_1xy = x_1x^{y_1^{-1}}y_1y \geq x_iy_ixy$ for each $i$ with $1 \leq i \leq r$. So there exists $x'y'$ in $\mathcal{V}_G(x_1y_1, \mathcal{S})$ with $x_1y_1x'y' \preceq x_1y_1xy$. Therefore $x_1(x')^{y_1^{-1}}y_1y' \preceq x_1x^{y_1^{-1}}y_1y$, $\pi(y_1y') \preceq_1 \pi(y_1y)$ and $x_1(x')^{y_1^{-1}} \preceq x_1x^{y_1^{-1}}$. Using the first part of this proof, we can see that $y'$ is in $\mathcal{V}_1$ and $(x')^{y^{-1}}$ is in $\mathcal{V}'$. Since $y$ is in $\mathcal{V}_1$ and $\pi(y_1y') \preceq_1 \pi(y_1y)$, $\pi(y_1y') = \pi(y_1y)$ and $y_1 = y$. Since $x^{y^{-1}}$ is in $\mathcal{V}'$ and $x_1(x')^{y^{-1}} \preceq x_1x^{y^{-1}}$, $x_1(x')^{y^{-1}} = x_1x^{y^{-1}}$ and $x = x'$. □

As a result, if we know how to find $\mathcal{U}_{H_i}(s_i, \mathcal{S}_i)$ for each finite subset $S_i$ of $H_i$ and each $s_i$ in $S_i$, then we know how to find $\mathrm{Symm}_G(f)$. A recursive algorithm is given below to find $\mathrm{Symm}_G(f)$.

Procedure Symmetrize($G$,$f$)

Input: $G$ : A group together with a subnormal series

$G = G_1 \rhd G_2 \rhd G_3 \rhd \ldots \rhd G_{e+1} = 1$,

quotients $H_i$ and coset representatives $Y_i$ as described above;

$f$ : A nonzero polynomial.

Output: $\mathrm{Symm}_G(f)$.

Begin

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

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

Let $f = f_1y_1 + f_2y_2 + \ldots + f_ry_r$, 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 $\mathbb{Z}[G_2]$;

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

$SP := \emptyset$;

For $i := 1$ to $r$ do begin

Let $\mathcal{V} := \mathcal{V}_H(\pi(y_i), \mathcal{S})$;

Let $SP' :=$ Symmetrize$(G_2, f_i)$;

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

end;

Return $SP$;

end.

Even though this procedure is given as a recursive procedure, it can actually be written as a nonrecursive procedure. There may be a lot of recursive calls to the procedure Symmetrize but the depth of recursion never exceeds $e$.

Proposition 8.1 gives us a bound on the size of $\mathrm{Symm}_G(f)$ for $f$ in $\mathbb{Z}[G]$. Suppose that $1 \leq i \leq e$. Assume that for every $f$ in $\mathbb{Z}[H_i]$, $|\mathrm{Symm}_{H_i}(f)|$ is bounded by $M_i$, then for $f$ in $\mathbb{Z}[G]$, $|\mathrm{Symm}_G(f)|$ is bounded by $M_1 \times M_2 \times \ldots \times M_e$. 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 $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \ldots$ or by $a^0 < a^1 < a^2 < \ldots < a^{-1} < a^{-2} < \ldots$, then $|\mathrm{Symm}_H(f)| \leq 2$ for all $f$ in $\mathbb{Z}[H]$. Therefore if $G$ has a subnormal series in which each quotient is infinite cyclic, then $|\mathrm{Symm}_G(f)| \leq 2^e$ for every $f$ in $\mathbb{Z}[G]$, where $e$ is the length of the subnormal series.

Since we know how to find $\mathrm{Symm}(f)$ for $f$ in $\mathbb{Z}[G]$, we will move on to the case $\mathbb{Z}[G]^k$. Suppose in extending the partial ordering $\preceq$ and the total ordering $\leq$ in $G$ to $\mathcal{T}$ and $\mathcal{M}$, we use the lexicographic extensions. Any nonzero $f$ in $\mathbb{Z}[G]^k$ can be written as $e_if_i + e_{i+1}f_{i+1} + \ldots + e_kf_k$ with $1 \leq i \leq k$, $f_i \neq 0$ and $f_i, f_{i+1}, \ldots, f_k$ in

$\mathbb{Z}[G]$. For any element $x$ in $G$, $\mathrm{lt}(fx)$ is of the form $e_i x'$ for some $x'$ in $G$. In fact, if $\mathrm{Symm}(f_i) = \{f_i x_1, f_i x_2, \ldots, f_i x_r\}$, then $\mathrm{Symm}(f) = \{fx_1, fx_2, \ldots, fx_r\}$. Since we know how to symmetrize polynomials in $\mathbb{Z}[G]$, we know how to symmetrize polynomial arrays in $\mathbb{Z}[G]^k$.

## 4.9 The Full Algorithm and Examples

At this point, we are ready to look at the full Gröbner basis algorithm in $\mathbb{Z}[G]^k$ when $G$ is polycyclic-by-finite. Let $G = G_1 \rhd G_2 \rhd G_3 \rhd \ldots \rhd G_{c+1} = 1$ be a subnormal series of $G$ such that each quotient is cyclic or finite. Assume that we have a reduction ordering set for $\mathbb{Z}[G]^k$.

Procedure Min_Gröbner_Basis($P$,$\mathcal{B}$)

Input: $P$ : A set of nonzero polynomial arrays in $\mathbb{Z}[G]^k$,

$P = \{f_1, f_2, \ldots, f_{|P|}\}$.

Output: $\mathcal{B}$ : A minimal Gröbner basis of $\langle P \rangle$.

Begin

$P' := P$;

$S := \mathrm{Symm}(f_1) \cup \mathrm{Symm}(f_2) \cup \ldots \cup \mathrm{Symm}(f_{|P|}) \cup$

$\{\alpha_{ij}, \beta_{ij} \mid 1 \le i < j \le |P|, f_i \text{ and } f_j \text{ are aligned}\}$;

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 $\mathrm{lm}(f_i) \preceq \mathrm{lm}(g)$ do begin

Let $x$ be such that $\mathrm{lt}(g) = \mathrm{lt}(f_i)x$;

$g := g - \lfloor \frac{\mathrm{lc}(g)}{\mathrm{lc}(f_i)} \rfloor f_i x$;

end;

If $g \neq 0$ then begin

$f_{p+1} := g$;

$S := S \cup \mathrm{Symm}(f_{p+1}) \cup$

$\{\alpha_{i,p+1}, \beta_{i,p+1} \mid 1 \le i \le p, f_i \text{ and } f_{p+1} \text{ are aligned}\}$;

$P' := P' \cup \{f_{p+1}\}$;

end

end;

Let $M'$ be $\min(\mathrm{lt}(P'))$ with respect to $\preceq$;

For each $m$ in $M'$, let $f_m$ be a polynomial in $P'$ such that $\mathrm{lt}(f_m) = m$;

Let $\mathcal{B}$ be the set of all such $f_m$'s;

end.

In the algorithm, we reduce the polynomial arrays in $\mathrm{Symm}(f_i)$ 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 6.3, $P'$ contains a minimal Gröbner basis, which can be found by the last three statements of the procedure. The procedure terminates because if $m_1 = \mathrm{lm}(f_{|P|+1}), m_2 = \mathrm{lm}(f_{|P|+2}), \ldots$, then the sequence of monomial vectors $m_1, m_2, \ldots$ satisfies the property that $m_i \not\preceq m_j$ whenever $i < j$. Therefore this sequence is finite. In finite time, the procedure will stop adding new polynomial arrays to $P'$. After the procedure finishes reducing the remaining polynomial arrays in $S$, it executes the last three statements and return a minimal Gröbner basis.

Now we will see two examples of Gröbner basis calculation in integral group rings of polycyclic groups. In the first example, 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 $a_2^{\lambda_2} a_1^{\lambda_1}$ 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 $f_1 = a_1^2 - 3a_1 + 2$ and $f_2 = a_2^2 - 3a_2 + 2$. Let $P$ be $\{f_1, f_2\}$. It can be deduced that $\mathrm{Symm}(f_1) = \{f_1, f_1 a_1^{-1}\}$ and $\mathrm{Symm}(f_2) = \{f_2, f_2 a_2, f_2 a_2^2\}$. Now $f_1 a_1^{-1} = 2a_1^{-1} + a_1 - 3$ 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 $f_2 a_2 = -3a_2^2 + 2a_2 + 1 \xrightarrow{*}_{P'} -7a_2 + 7$ 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 $f_2 a_2^2 = 2a_2^2 + a_2 - 3 \xrightarrow{*}_{P'} 0$. 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 $P' = \{a_1^2 - 3a_1 + 2, 2a_1^{-1} + a_1 - 3, a_2^2 - 3a_2 + 2, 7a_2 - 7\}$.

In the second example, suppose that $G$ 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 almost a consistent polycyclic presentation for $G$. The collected form of an element in $G$ is again $a_2^{\lambda_2}a_1^{\lambda_1}$ 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 $P = \{a_1^2 - 3a_1 + 2, a_2^2 - 3a_2 + 2\}$. Now $\mathrm{Symm}(f_1)$ is $\{f_1, f_1a_1^{-1}\}$ and $\mathrm{Symm}(f_2)$ is $\{f_2, f_2a_2^{-1}\}$. As in the last example, $f_1a_1^{-1} = 2a_1^{-1} + a_1 - 3$ cannot be reduced by $P$. So we define $f_3$ to be $2a_1^{-1} + a_1 - 3$ and set $P'$ to be $P \cup \{f_3\}$. Also $f_2a_2^{-1} = 2a_2^{-1} + a_2 - 3$ 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 $\beta_{12} = f_1a_2^2 - f_2a_1^2 \xrightarrow{*}_{P'} a_2^{-2}a_1 + 3a_2^{-1}a_1 - 4a_1 + 2a_2^2 - 6a_2 + 4$, which we call $f_5$ and add to $P'$. The symmetrized set of $f_5$ is $\{f_5a_2^{-1}, -f_5a_2^{-2}, f_5a_1^{-1}, f_5a_2^{-1}a_1^{-1}\}$. Now $f_5a_1^{-1} \xrightarrow{*}_{P'} a_2^{-2} + 3a_2^{-1} - 4$. The symmetrized set of $a_2^{-2}+3a_2^{-1}-4$ is $\{a_2^{-1}-4a_2+3, 4a_2^2-3a_2-1\}$. Define $f_6$ to be $a_2^{-1}-4a_2+3$ and reset $P'$ to include $f_6$. Reducing $4a_2^2 - 3a_2 - 1$ we get $9a_2 - 9$. Let $f_7$ be $9a_2 - 9$ and set $P'$ to be $P' \cup \{f_7\}$. Reducing $\alpha_{37} = 5f_3a_2^{-1} - f_7a_1^{-1}$ we get $a_2a_1^{-1} + 9a_1^{-1} + 5a_2a_1 - 15a_2^{-1}$, which we call $f_8$ 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 $\{f_1, f_2\}$. A minimal Gröbner basis of this ideal is $\{f_1, f_2, f_3, f_6, f_7, f_8\}$.

## 4.10 Gröbner Basis Strategies

Let $P = \{f_1, f_2, \ldots, f_r\}$ be a symmetrized set of polynomial arrays in $\mathbb{Z}[G]^k$. In Section 6 we saw that if $\alpha_{ij}$ and $\beta_{ij}$ can be reduced to 0 for every $i, j$ such that $1 \le i < j \le r$ and $f_i$ and $f_j$ are aligned, then $P$ contains a minimal Gröbner basis of the submodule $\langle P\rangle$ generated by $P$. The algorithm given in Section 9 shows how to find a Gröbner basis of $\langle P\rangle$. In this algorithm, we reduce the polynomial arrays $\alpha_{ij}$ and $\beta_{ij}$ for every $i, j$ such that $1 \le i < j \le r$ and $f_i$ and $f_j$ are aligned. It is possible, however, to avoid reducing some of them and thus improve the overall efficiency of the computation. In this section, we will discuss some strategies to cut down the number of polynomial arrays to reduce. Here, we assume that the reduction total ordering is a linear extension of the reduction partial ordering. Let us call this assumption the *linear extension assumption.* The reduction strategy discussed in Proposition 10.2 is an adaptation of the one proposed in [GM88].

To simplify the exposition, we make the following definition.

**Definition:** Let $P$ be a set of nonzero polynomial arrays. Let $g_1$ and $g_2$ be polynomial arrays and $t$ be a term vector. Define $g_1 \equiv_t g_2$ if $g_1 - g_2$ is $\text{lower}_P - t$.

**Lemma 4.10.1** *Let $P$ be a set of nonzero polynomial arrays. Let $g_1, g_2$ and $g$ be polynomial arrays. Let $t$ be a term vector.*

1. *As a relation on the set of polynomial arrays, $\equiv_t$ is an equivalence relation.*
2. *If $g_1 \equiv_t g_2$, then $g_1 + g \equiv_t g_2 + g$.*
3. *If $P$ is symmetrized and $x$ is an element in $G$ such that $x$ is aligned with $t$, then $g_1 x \equiv_{tx} g_2 x$.*

**Proof:** Parts 1 and 2 are clear. Part 3 follows from part 1 of Lemma 6.2. $\square$

Let $P = \{f_1, f_2, \ldots, f_r\}$ be a symmetrized set of polynomial arrays. If $1 \leq i < j \leq r$ and $f_i$ and $f_j$ are aligned, then we say that the pair $(i, j)$ is a *critical pair.* If $(i, j)$ is a critical pair, then we define $c_{ij} = c_{ji}$ to be $\text{lcm}(\text{lc}(f_i), \text{lc}(f_j))$ and $t_{ij} = t_{ji}$ to be $\text{lcm}(\text{lt}(f_i), \text{lt}(f_j))$. Define a total ordering $\leq$ on $P$ by $f_i < f_j$ if either $\text{lm}(f_i) < \text{lm}(f_j)$, or $\text{lm}(f_i) = \text{lm}(f_j)$ and $i < j$. Define $\min(f_i, f_j)$ to be the minimum of $f_i$ and $f_j$ in the ordering $\leq$, and $\max(f_i, f_j)$ to be the maximum of $f_i$ and $f_j$ in $\leq$. We would also like to define a total ordering on the set of critical pairs. Again we denote this total ordering by $\leq$. Given two critical pairs $(i, j)$ and $(i', j')$, to define the relation $\leq$, first we compare $c_{ij} t_{ij}$ and $c_{i'j'} t_{i'j'}$ using the ordering $\leq$ on monomials. If $c_{ij} t_{ij} = c_{i'j'} t_{i'j'}$, then we compare $\max(f_i, f_j)$ and $\max(f_{i'}, f_{j'})$ using the ordering $\leq$ on $P$ as described above. If we still have a tie, then we compare $\min(f_i, f_j)$ and $\min(f_{i'}, f_{j'})$, again using the ordering $\leq$ on $P$. Since the number of critical pairs is finite, $\leq$ is a well-ordering. Call a critical pair $(i, j)$ *redundant* if there exists $l \neq i, j$ such that $t_{il} \preceq t_{ij}$, $t_{jl} \preceq t_{ij}$, both $c_{il}$ and $c_{jl}$ divide $c_{ij}$ properly as integers, and the critical pair formed by $i$ and $l$

and the critical pair formed by $j$ and $l$ are both smaller than the critical pair formed by $i$ and $j$ in the ordering $\leq$. Call $(i,j)$ *nonredundant* if it is not redundant. For $1 \leq i,j \leq r$, we say that $f_i$ *divides* $f_j$ if $\mathrm{lt}(f_i) \preceq \mathrm{lt}(f_j)$ and $\mathrm{lc}(f_i)$ divides $\mathrm{lc}(f_j)$ properly as integers. We say that $f_i$ is *essential* if $1 \leq j \leq r$ and $f_j$ divides $f_i$ implies that $i \leq j$.

**Proposition 4.10.2** *Let $P$ be a finite symmetrized set of polynomial arrays. Assume that for each $f_i$ which is not essential, there exists $f_l$ in $P$ such that $f_l$ divides $f_i$, $f_l < f_i$ and $\beta_{il}$ is lower-$t_{il}$. Let $\mathcal{K}$ be the set of all nonredundant critical pairs $(i,j)$ such that both $f_i$ and $f_j$ are essential. If $\beta_{ij}$ is lower-$t_{ij}$ for each $(i,j)$ in $\mathcal{K}$, then $\beta_{ij}$ is lower-$t_{ij}$ for each critical pair $(i,j)$.*

**Proof:** Let us argue by contradiction. Let $(i,j)$ be the smallest critical pair in $\leq$ such that $\beta_{ij}$ is not lower-$t_{ij}$. Suppose that $(i,j)$ is redundant. Let $l$ be as described in the definition of a redundant critical pair. By definition $\beta_{ij} = \frac{c_{ij}}{lc(f_i)} f_i x_i - \frac{c_{ij}}{lc(f_j)} f_j x_j$, $\beta_{il} = \pm(\frac{c_{il}}{lc(f_i)} f_i x_i' - \frac{c_{il}}{lc(f_l)} f_l x_l')$ where $x_i, x_j, x_i', x_l'$ are in $G$, $\mathrm{lt}(f_i x_i) = \mathrm{lt}(f_j x_j) = t_{ij}$ and $\mathrm{lt}(f_i x_i') = \mathrm{lt}(f_l x_l') = t_{il}$. Since $c_{il}$ divides $c_{ij}$ properly and $t_{il} \preceq t_{ij}$, $\frac{c_{ij}}{lc(f_i)} f_i x_i = \frac{c_{il}}{lc(f_i)} f_i x_i' m$, where $m$ is a monomial. By the minimality of the pair $(i,j)$ and the linear extension assumption, $\beta_{il}$ is lower-$t_{ij}$. So $\frac{c_{ij}}{lc(f_i)} f_i x_i = \frac{c_{il}}{lc(f_l)} f_l x_l' m \pm \beta_{il} m \equiv_{t_{ij}} f_l m'$ where $m' = \frac{c_{il}}{lc(f_l)} x_l' m$. Similarly, $\frac{c_{ij}}{lc(f_j)} f_j g_j \equiv_{t_{ij}} f_l m''$ for some monomial $m''$ such that $\mathrm{lt}(f_l m'')$ is $t_{ij}$. Therefore $\beta_{ij} \equiv_{t_{ij}} f_l(m' - m'')$. Since $\beta_{ij}$ has leading term vector smaller than $t_{ij}$ in $\leq$, and both $f_l m'$ and $f_l m''$ have leading term vector $t_{ij}$, $m' = m''$ and $\beta_{ij}$ is lower-$t_{ij}$.

Suppose that $f_i$ is not essential. Assume that $l$ is such that $f_l$ divides $\mathrm{lt}(f_i)$, $f_l < f_i$ and $\beta_{il}$ is lower-$t_{il}$. So $f_i \equiv_{t_{il}} f_l m$ for some monomial $m$ such that $m$ is aligned with $f_l$. Notice that $t_{il} = \mathrm{lt}(f_i)$. Now $\beta_{ij} = f_i m' + f_j m''$ for some monomials $m', m''$. Therefore $\beta_{ij} \equiv_{t_{ij}} f_l m m' + f_j m''$. It can be seen that $f_l m m' + f_j m'' = \beta_{jl}\bar{m}$ or $\beta_{lj}\bar{m}$, depending on whether $j < l$ or $l < j$, where $\bar{m}$ is some monomial and $\mathrm{lt}(\beta_{jl}\bar{m}) < t_{ij}$. It is routine to check that the critical pair formed by $j$ and $l$ is smaller than the critical pair $(i,j)$ in $\leq$. By the minimality of the pair $(i,j)$, $\beta_{jl}$ or $\beta_{lj}$ is lower-$t_{jl}$. We use the linear extension assumption here. Therefore $\beta_{ij}$ is lower-$t_{ij}$.

The case when $f_j$ is not essential is similar. $\square$

We say that the polynomial array $\alpha_{ij}$ can be *loosely reduced* relative to $P$ if $\alpha_{ij} \equiv_{t_{ij}} g_1m_1 + g_2m_2 + \ldots + g_sm_s$, where $s$ is a positive integer and for all $l$ with $1 \leq l \leq s$, $g_lm_l$ is a proper $P$-product, $\mathrm{lt}(g_lm_l) = t_{ij}$ and $\mathrm{lc}(g_l) < \max(\mathrm{lc}(f_i), \mathrm{lc}(f_j))$.

**Proposition 4.10.3** *Let $P$ be a finite symmetrized set of polynomial arrays. Suppose that $\beta_{ij}$ is lower-$t_{ij}$ for each critical pair $(i,j)$. If $\alpha_{ij}$ can be weakly reduced or loosely reduced for each critical pair $(i,j)$ such that $f_i$ and $f_j$ are essential, then $\alpha_{ij}$ can be weakly reduced for each critical pair $(i,j)$.*

**Proof:** Assume the hypothesis. Let $S$ be the set of critical pairs $(i,j)$ such that $\alpha_{ij}$ cannot be weakly reduced. If $S$ is nonempty, then let $(i,j)$ in $S$ be such that $\max(\mathrm{lc}(f_i), \mathrm{lc}(f_j))$ is minimum. If there is more than one such pair $(i,j)$, pick the one which is smallest with respect to $\leq$. There are three possibilities. First, $f_i$ may not be essential. Second, $f_j$ may not be essential. Third, $\alpha_{ij}$ can be loosely reduced but cannot be weakly reduced. We will consider the case when $f_i$ is not essential. The case when $f_j$ is not essential is similar.

Let $f_l$ be such that $f_l$ divides $f_i$, $l < i$. Therefore $f_i = f_lm + \beta_{il}$ for some monomial $m$. Clearly $f_lm$ is a proper product . So $f_i \equiv_{\mathrm{lt}(f_i)} f_lm$. Now $\alpha_{ij} = f_im' + f_jm''$ for some monomials $m'$ and $m''$. Therefore $\alpha_{ij} \equiv_{t_{ij}} f_lmm' + f_jm'' = q_1\alpha_{jl}x + q_2\beta_{jl}x$ where $q_1, q_2$ are integers and $x$ is the group element such that $t_{ij} = t_{jl}x$. Since $\beta_{jl}x$ is lower-$t_{ij}$, $\alpha_{ij} \equiv_{t_{ij}} q_1\alpha_{jl}x$. By the choice of the pair $(i,j)$ and using the linear extension assumption, $\alpha_{jl}$ can be weakly reduced, therefore $\alpha_{ij}$ can be weakly reduced.

Now assume that $\alpha_{ij}$ can be loosely reduced, we want to show that $\alpha_{ij}$ can be weakly reduced. By assumption, $\alpha_{ij} \equiv_{t_{ij}} g_1m_1 + g_2m_2 + \ldots + g_sm_s$ where $g_1m_1, g_2m_2, \ldots, g_sm_s$ are proper products as in the definition of a loosely reduced polynomial array. Let us call such a way of writing $\alpha_{ij}$ a *representation* of $\alpha_{ij}$. Define the *integer parameter* of a representation to be the integer $|\mathrm{lc}(g_1m_1)| + |\mathrm{lc}(g_2m_2)| + \ldots + |\mathrm{lc}(g_sm_s)|$. If $g_1m_1, g_2m_2, \ldots, g_sm_s$ are of the same sign, then we are done. Now assume that $g_1m_1$

and $g_2m_2$ have opposite signs. Let $g_1$ and $g_2$ be the polynomial arrays $f_{i'}$ and $f_{j'}$, respectively. Without loss of generality, assume that $j' > i'$. So $g_1m_1 + g_2m_2 \equiv_{t_{ij}} \alpha_{i'j'}m$ for some monomial $m$. By the choice of the pair $(i, j)$, $\alpha_{i'j'}$ can be weakly reduced. Therefore $\alpha_{i'j'} \equiv_{t_{i'j'}} g'_1m'_1 + g'_2m'_2 + \ldots + g'_{s'}m'_{s'}$ as in the definition of weakly reduced. As a result, $\alpha_{ij} \equiv_{t_{ij}} g'_1m'_1x + g'_2m'_2x + \ldots + g'_{s'}m'_{s'}x + g_3m_3 + g_4m_4 + \ldots + g_sm_s$ where $x$ is the group element in $G$ such that $t_{i'j'}x = t_{ij}$. Note that for $1 \leq l \leq s'$, $\mathrm{lc}(g'_l) \leq \mathrm{lc}(\alpha_{i'j'}) \leq \mathrm{lc}(g_1) < \max(\mathrm{lc}(f_i), \mathrm{lc}(f_j))$. In this representation of $\alpha_{ij}$, the integer parameter $|\mathrm{lc}(g'_1m'_1)| + |\mathrm{lc}(g'_2m'_2)| + \ldots + |\mathrm{lc}(g'_{s'}m'_{s'})| + |\mathrm{lc}(g_3m_3)| + |\mathrm{lc}(g_4m_4)| + \ldots + |\mathrm{lc}(g_sm_s)|$ is smaller than $|\mathrm{lc}(g_1m_1)| + |\mathrm{lc}(g_2m_2)| + \ldots + |\mathrm{lc}(g_sm_s)|$, the integer parameter of the previous representation. We can repeat this rewriting process as long as there are two proper products in the representation of $\alpha_{ij}$ having different signs on their leading coefficients. Using an induction argument, we can rewrite $\alpha_{ij}$ such that all the proper products have the same sign, namely the positive sign. Therefore $\alpha_{ij}$ can be weakly reduced. $\square$

# Chapter 5

# Computations with Gröbner Bases

In the previous chapter we saw how to obtain a Gröbner basis of a submodule of a free $\mathbb{Z}[G]$-module of finite rank where $G$ is a reduction group. Here we will see how to use it.

## 5.1 Finding Group Generators

In this section, we will answer the questions posed in Chapter 3. 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 $\mathrm{lc}(f) > 0$ for all $f$ in $P$. Suppose that $g$ is a nonzero polynomial array with standard form $c_1t_1 + c_2t_2 + \ldots + c_rt_r$. We say that $g$ is *fully reduced* relative to $P$ if for every $f$ in $P$ and for every $i$ with $1 \leq i \leq r$, $\mathrm{lm}(f) \not\preceq c_it_i$. We also say that the zero polynomial array is fully reduced. Given a polynomial array $g_1$, it is possible to find a polynomial array $g$ such that $g$ is fully reduced relative to $P$ and $g_1 - g$ is in the right ideal generated by $P$. In fact, such $g$ can be found in a finite number of steps. We proceed by first reducing $g_1$ until it cannot be reduced any further by $P$. More precisely, in a finite number of reductions, we can find a polynomial array $g_1'$ such that $g_1 \xrightarrow{*}_P g_1'$ and either $g_1' = 0$ or $\mathrm{lm}(f) \not\preceq \mathrm{lm}(g_1')$ for each $f$ in $P$. If $g_1' = 0$ or if the standard form of $g_1'$ consists of only one monomial, then take $g$ to be $g_1'$. Otherwise let $c_1t_1$ be $\mathrm{lm}(g_1')$ and let $g_2$ be $g_1 - c_1t_1$. Repeat the same computation with $g_2$. In a finite number of reductions we can find a polynomial array $g_2'$ such that $g_2 \xrightarrow{*}_P g_2'$ and $g_2'$ cannot be reduced any more by $P$. If $g_2' = 0$ or if the standard form of $g_2'$ consists of only one monomial, then take $g = c_1t_1 + g_2'$. Otherwise let $c_2t_2$ be $\mathrm{lm}(g_2')$ and let $g_3$ be $g_2 - c_2t_2$. We repeat the same procedure with $g_3$. In this way we will get a sequence of monomial vectors $c_1t_1, c_2t_2, \ldots$. Since $t_1 > t_2 > \ldots$ and $\leq$ is a well-ordering, this process will terminate eventually and we will obtain a fully reduced polynomial array $g$.

Let $\mathcal{B}$ be a Gröbner basis of a right submodule $M_0$ of $\mathbb{Z}[G]^k$. For any polynomial array $g$, we can find a unique polynomial array $g'$, fully reduced relative to $\mathcal{B}$, such that $g - g'$ is in $M_0$. The procedure described in the previous paragraph will find such a $g'$ and it is not hard to see that $g'$ is unique. We denote $g'$ by $\mathrm{nf}(g)$ and call $g'$ the *normal form* of $g$ relative to $\mathcal{B}$.

**Proposition 5.1.1** *Let $\mathcal{B}$ 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 $\mathcal{B}$. Then $\mathbb{Z}[G]^k/M_0$ is generated as an abelian group by the set $\{t + M_0 \mid t \in T\}$.*

**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 $c_1t_1 + c_2t_2 + \ldots + c_rt_r$ be the standard form of $g'$. Then $g + M_0 = c_1(t_1 + M_0) + c_2(t_2 + M_0) + \ldots + c_r(t_r + M_0)$. □

Suppose that $G$ is a polycyclic group. Let $G = G_1 \triangleright G_2 \triangleright \ldots \triangleright G_{n+1} = 1$ be a polycyclic series of $G$ with $1_1, a_2, \ldots, a_n$ as the polycyclic generating sequence. Define $Y_i$ to be $\{a_i^j \mid j \in \mathbb{Z}\}$ if $G_i/G_{i+1}$ is infinite and $\{a_i^0, a_i^1, \ldots, a_i^{m_i-1}\}$ if $G_i/G_{i+1}$ is finite of order $m_i$. Assume that for each $i$ with $1 \leq i \leq n$, the partial ordering $\preceq_i$ is as described in the examples in Section 4.4 and the partial ordering $\preceq$ on $G$ is the standard partial ordering defined using the parameters above.

**Proposition 5.1.2** *Let $\mathcal{B}$ be a Gröbner basis of a right submodule $M_0$ in $\mathbb{Z}[G]^k$. Let $\mathcal{K}$ 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 if and only if for each $(i,j)$ in $\mathcal{K}$, there exists integers $\alpha^{(ij)} \geq 0$, $\beta^{(ij)} \leq 0$ such that $e_ia_j^{\alpha^{(ij)}}$ and $e_ia_j^{\beta^{(ij)}}$ are leading monomial vectors of some polynomial arrays in $\mathcal{B}$.*

**Proof:** Suppose that $\alpha^{(ij)}$ and $\beta^{(ij)}$ exist for every pair $(i,j)$ in $\mathcal{K}$. Then the set $T$ defined in Proposition 1.1 is a subset of the set $T' = \{e_i a_n^{\lambda_n} \dots a_2^{\lambda_2} a_1^{\lambda_1} \in \mathcal{T} \mid \forall (i,j) \in \mathcal{K}, \beta^{(ij)} < \lambda_j < \alpha^{(ij)}\}$. It is not hard to see that $T'$ is finite. Therefore $T$ is finite and $\mathbb{Z}[G]^k/M_0$ is finitely generated as an abelian group.

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

## 5.2 Constructing Polycyclic Presentations

In this section we will finish up with the extension algorithm. We will use the notation established in Section 3.3. By Proposition 3.3.5, $M/M_0$ and $N/N'$ are isomorphic as $\mathbb{Z}[G/N]$-modules. Since $M$ is the free $\mathbb{Z}[G/N]$-module $\mathbb{Z}[G/N]^k$ and we can find finitely many submodule generators for $M_0$, it is possible to find a Gröbner basis $\mathcal{B}$ for $M_0$. By Proposition 1.2, it is possible to decide whether $M/M_0$, and thus whether $N/N'$ is finitely generated as an abelian group. Using Theorem 2.3.1, we see that $G/N'$ is polycyclic if and only if $N/N'$ is finitely generated as an abelian group. Therefore we can decide whether $G/N'$ is polycyclic. Suppose that $G/N'$ is in fact polycyclic. Then the set $T$ in Proposition 1.1 is a finite set of group generators for $M/M_0$. It is not hard to find $T$. We begin by finding the set $T'$ defined in the proof of Proposition 1.2. Since $T'$ is a finite set containing $T$, $T$ can be found by eliminating elements in $T'$ but not in $T$. 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 $T$ both as 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 an element in $T$ to be considered as a term vector. As term vectors, $[a_{n+1}] > [a_{n+2}] > \ldots > [a_l]$. What we will do basically is to transform, similar in principle to Tietze transformation, the presentation given in Theorem 3.3.3 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 $a_i$ if and only if there exists a positive integer $c$ such that the monomial vector $[ca_i]$ can be reduced by $\mathcal{B}$. Since the monomial vector $[a_i]$ cannot be reduced by $\mathcal{B}$, we see that $c > 1$. Let $c_i$ be the smallest such $c$. If $\mathrm{nf}([c_i a_i]) = [c'_{i+1}a_{i+1} + c'_{i+2}a_{i+2} + \ldots + c'_l a_l]$. then the new power relation involving $a_i$ is $a_i^{c_i} = a_l^{c'_l} \ldots a_{i+2}^{c'_{i+2}} a_{i+1}^{c'_{i+1}}$. We also obtain the new power relation with left side $a_i^{-1}$ in a similar way. Let $I'$ be the set of all $i$, $1 \leq i \leq l$, 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 $i \notin I'$, $a_i a_j^{-1} = a_j^{-1} a_i$ for $j \notin I'$ and $a_i^{-1} a_j^{-1} = a_j^{-1} a_i^{-1}$ for $i, j \notin I'$. 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 $a_i^{\pm 1} a_j^{\pm 1}$. We will illustrate here how the rule with left side $a_i a_j$ can be obtained. The rest are similar. As an element in $M$, $[a_j a_i^{-1}]$ is the module element $[a_j]$ acted on by $a_i^{-1}$. If the normal form of $[a_j a_i^{-1}]$ is $[c'_{n+1}a_{n+1} + c'_{n+2}a_{n+2} + \ldots + c'_l a_l]$, then there is a new conjugacy relation $a_i a_j = a_l^{c'_l} \ldots a_{n+2}^{c'_{n+2}} a_{n+1}^{c'_{n+1}} a_i$. Now we have to tackle the relations inherited from $S$. Recall that $|S| = s$. Let $U_i = V_i$ be a relation in $S$ where $1 \leq i \leq s$. In Section 3.3, we modified this relation to $U_i = [z'_i]V_i$ where $z'_i$ is in $Z'$. Let the normal form of $[z'_i]$ be $[c'_{n+1}a_{n+1} + c'_{n+2}a_{n+2} + \ldots + c'_l a_l]$. We obtain a new relation $U_i = a_l^{c'_l} \ldots a_{n+2}^{c'_{n+2}} a_{n+1}^{c'_{n+1}} V_i$. Let $S'$ be the set consisting of all the new relations described above. Then a consistent polycyclic presentation for $G/N'$ is $\langle Y' \mid S' \rangle$.

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 Chapter 3, we defined $\mu(x_i)$ to be $[z_i]^{-1}\sigma(x_i)$ for each $x_i$ in $X$. If the normal form of $[-z_i]$ is $[c'_{n+1}a_{n+1} + c'_{n+2}a_{n+2} +$

$\ldots + c_l' a_l]$, then we define $\sigma'(x_i)$ to be $a_l^{c_l'} \ldots a_{n+2}^{c'_{n+2}} a_{n+1}^{c'_{n+1}} \sigma(x_i)$. The homomorphism $\nu$ defined in Section 3.3 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 $G/N'$.

### 5.3 Computation in Modules

In this section we will consider methods to compute in right modules of the integral group ring of a reduction group $G$. From here on in this chapter, all modules are right modules unless otherwise stated.

Suppose that $P$ is a subset of $\mathbb{Z}[G]^k$. In Chapter 4 we saw that it is possible to find a Gröbner basis of the right $\mathbb{Z}[G]$-module $M_0$ generated by $P$. It is also possible to find a Gröbner basis of the two-sided $\mathbb{Z}[G]$-module $M'$ generated by $P$. Suppose the set $\{b_1, b_2, \ldots, b_l\}$ generates $G$. First we find a Gröbner basis $\mathcal{B}_1$ for $M_0$. Observe that $\mathcal{B}_1$ is a Gröbner basis of $M'$ if and only if for each $m$ in $\mathcal{B}_1$ and each integer $i$ with $1 \leq i \leq l$, $b_i m$ and $b_i^{-1} m$ are in $M_0$. By performing reductions, we can test membership in $M_0$. If there exists a module element of the form $b_i m$ or $b_i^{-1} m$ which is not in $M_0$, then we find a Gröbner basis $\mathcal{B}_2$ for the right submodule generated by $\mathcal{B}_1 \cup \{b_i m \mid 1 \leq i \leq n, m \in \mathcal{B}_1\}$. We then proceed to check whether $b_i m, b_i^{-1} m \xrightarrow{*}_{\mathcal{B}_2} 0$ for every $i$ with $1 \leq i \leq l$ and every $m$ in $\mathcal{B}_2$. If this is true, then $\mathcal{B}_2$ is a Gröbner basis of $M'$. If not, then we proceed to find $\mathcal{B}_3$, a Gröbner basis of the right submodule generated by $\mathcal{B}_2 \cup \{b_i m, b_i^{-1} m \mid 1 \leq i \leq l, m \in \mathcal{B}_2\}$. Continue this process until we have obtained a Gröbner basis of $M'$. We need to show that this process will stop eventually. Consider the sequence of Gröbner bases $\mathcal{B}_1, \mathcal{B}_2, \ldots$. If $\langle \mathcal{B}_j \rangle$ denotes the right $\mathbb{Z}[G]$-submodule generated by $\mathcal{B}_j$ for $j \geq 1$, then $\langle \mathcal{B}_1 \rangle \subset \langle \mathcal{B}_2 \rangle \subset \ldots$. Since $\mathbb{Z}[G]$ is right Noetherian by Corollary 4.5.3, this sequence must stop and we will obtain a Gröbner basis $\mathcal{B}$ of $M'$ after we have checked that $b_i m$ and $b_i^{-1} m$ are in $\langle \mathcal{B} \rangle$ for every $i$ with $1 \leq i \leq l$ and every $m$ in $\mathcal{B}$.

Recall that $e_1, e_2, \ldots, e_k$ are standard basis elements in $\mathbb{Z}[G]^k$. Let $l$ be an integer with $1 \leq l \leq k$. We define $\mathbb{Z}[G]_l^k$ to be the submodule of $\mathbb{Z}[G]^k$ generated by $e_l, e_{l+1}, \ldots, e_k$. Suppose $\mathcal{B}$ is a Gröbner basis of a submodule $M_0$ of $\mathbb{Z}[G]^k$. Define $\mathcal{B}_l$

to be $\mathcal{B} \cap \mathbb{Z}[G]_l^k$.

**Lemma 5.3.1** *If $g$ is in $M_0 \cap \mathbb{Z}[G]_l^k$, then $g$ can be written as $f_1h_1 + f_2h_2 + \ldots + f_sh_s$, where for $1 \leq i \leq s$, $f_ih_i$ is a proper $\mathcal{B}_l$-product.*

**Proof:** Suppose $g$ is in $M_0 \cap \mathbb{Z}[G]_l^k$. Since lexicographic extensions are used in defining the partial and total orderings on $\mathcal{T}$ and $\mathcal{M}$, when we reduce $g$, we only use polynomial arrays in $\mathcal{B}_l$. Now $g$ can be reduced to 0, therefore $g$ can be written in the form stated in the lemma. $\square$

The following lemma follows from Theorem 4.6.3 and Lemma 3.1.

**Lemma 5.3.2** *Let $M_0$ be a submodule of $\mathbb{Z}[G]^k$. Let $\mathcal{B} = \{f_1, f_2, \ldots, f_r\}$ be a Gröbner basis of $M_0$. If $\mathcal{B}_l = \mathcal{B} \cap \mathbb{Z}[G]_l^k$, then $\mathcal{B}_l$ is a Gröbner basis for the submodule generated by $\mathcal{B}_l$.*

Let $\overline{M}$ be a finitely generated right $\mathbb{Z}[G]$-module. Then $\overline{M}$ is isomorphic to $M/M_0$ for a free right $\mathbb{Z}[G]$-module $M$ of finite rank and a submodule $M_0$ of $M$. Since $G$ is a reduction group, $\mathbb{Z}[G]$ is right Noetherian and $M_0$ is finitely generated as a right $\mathbb{Z}[G]$-module. Given a finite set of generators for $M_0$, we can find a Gröbner basis $\mathcal{B}$ for $M_0$. The word problem in $M/M_0$ can be solved as follows. Given two elements in $M$, we can find and compare their normal forms relative to $\mathcal{B}$. The two elements are equal modulo $M_0$ if and only if their normal forms are equal.

Let $Z$ be a finite set and $M$ be the free $\mathbb{Z}[G]$-module on $Z$. Let $T$ be a finite subset of $M$ and $\langle T \rangle$ be the submodule of $M$ generated by $T$. We say that a $\mathbb{Z}[G]$-module $M'$ has a *module presentation* $\langle Z \,||\, T \rangle$ if there exists a right module epimorphism $\psi : M \to M'$ such that the kernel of the epimorphism is $\langle T \rangle$. It is clear that every finitely generated $\mathbb{Z}[G]$-module has a module presentation $\langle Z \,||\, T \rangle$. Moreover, we can find one such that $T$ is a Gröbner basis for the submodule $\langle T \rangle$.

Suppose $\langle Z \,||\, T \rangle$ is a module presentation for a $\mathbb{Z}[G]$-module $M'$. Let $M$ be the free module on $Z$ and $M_0$ be the submodule of $M$ generated by $T$. Then $M'$ is isomorphic as a right $\mathbb{Z}[G]$-module to to $M/M_0$. Every quotient of $M'$ is isomorphic to $M/M_1$ for some submodule $M_1$ containing $M_0$ of $M$. If $T'$ is a set of polynomial arrays in $M$ such that $T'$ generates $M_1$ modulo $M_0$, then a module presentation for the quotient $M'$ is $\langle Z \,||\, T \cup T' \rangle$.

Let $Z = \{z_1, z_2, \ldots, z_r\}$ and $Z' = \{z'_1, z'_2, \ldots, z'_{r'}\}$ be disjoint sets. Suppose that we have module presentations $\langle Z \,||\, T \rangle$ and $\langle Z' \,||\, T' \rangle$ for two $\mathbb{Z}[G]$-modules. Let $M$ and $M'$ be free $\mathbb{Z}[G]$-modules on $Z$ and $Z'$, respectively. Let $M_0$ and $M'_0$ be the submodules generated by $T$ and $T'$, respectively. A module homomorphism $\varphi$ from $M/M_0$ to $M'/M'_0$ is determined by its image $\varphi(z + M_0)$ for each $z$ in $Z$. Let $\overline{\varphi}(z)$ be a polynomial array in $M'$ such that $\varphi(z + M_0) = \overline{\varphi}(z) + M'_0$ for each $z$ in $Z$. It is clear that the polynomial arrays $\overline{\varphi}(z_1), \overline{\varphi}(z_2), \ldots, \overline{\varphi}(z_r)$ determine the homomorphism $\varphi$. Moreover, the range of $\varphi$, $\varphi(M/M_0)$, is generated by $\overline{\varphi}(z_1), \overline{\varphi}(z_2), \ldots, \overline{\varphi}(z_r)$ modulo $M'_0$. It is possible to find the kernel $M_1/M_0$ of $\varphi$ by finding a Gröbner basis of the submodule $M_1$ as follows. Consider the free $\mathbb{Z}[G]$-module $N$ on the set $Z' \cup Z$. As a $\mathbb{Z}[G]$-module $N$ is isomorphic to $\mathbb{Z}[G]^{r+r'}$. The set $Z' \cup Z$ can be thought of as a standard module basis of $N$. We will write an element in $N$ as an ordered pair $(m', m)$ where $m'$ is in $M'$ and $m$ is in $M$. Consider the submodule $N_0$ of $N$ generated by $U = \{(\overline{\varphi}(z), z) \mid z \in Z\} \cup \{(0, t) \mid t \in T\} \cup \{(t', 0) \mid t' \in T'\}$. It is possible to find a Gröbner basis $\mathcal{B}_N$ for $N_0$ using lexicographic extensions with $z'_1 > z'_2 > \ldots > z'_{r'} > z_1 > z_2 > \ldots > z_r$.

**Proposition 5.3.3** *$\mathcal{B} = \{m \mid (0, m) \in \mathcal{B}_N\}$ is a Gröbner basis of $M_1$.*

**Proof:** Let $m$ be in $\mathcal{B}$. Then $(0, m)$ is contained in $N_0$. Therefore $(0, m) = \sum_{u \in U} u h_u$ where for each $u$ in $U$, $h_u$ is in $\mathbb{Z}[G]$. Observe that for each $(m', m)$ in $U$, $\varphi(m + M_0) = m' + M'_0$. It follows that $\varphi(m + M_0) = M'_0$ and so $m$ is in $M_1$.

Now assume that $m$ is in $M_1$. Let $m = z_1 h_1 + z_2 h_2 + \ldots + z_r h_r$ where $h_1, h_2, \ldots, h_r$ are in $\mathbb{Z}[G]$. Then $M'_0 = \varphi(m + M_0) = \varphi(z_1)h_1 + \varphi(z_2)h_2 + \ldots + \varphi(z_r)h_r = \overline{\varphi}(z_1)h_1 + \overline{\varphi}(z_2)h_2 + \ldots + \overline{\varphi}(z_r)h_r + M'_0$. In other words, $\overline{\varphi}(z_1)h_1 + \overline{\varphi}(z_2)h_2 + \ldots + \overline{\varphi}(z_r)h_r = \sum_{t' \in T'} t' h'_{t'}$ where $h'_{t'}$ is in $\mathbb{Z}[G]$ for each $t'$ in $T'$. Therefore $(0, m) = \sum_{i=1}^{r} (\overline{\varphi}(z_i), z_i) h_i - \sum_{t' \in T'} (t', 0) h'_{t'}$ which is in $N_0$ and can be reduced to 0 by $\mathcal{B}_N$. By Lemma 3.1, if we reduce $(0, m)$, we only need to use elements of the form $(0, \overline{m})$ in $\mathcal{B}_N$. This implies that $m$ can be reduced to 0 by $\mathcal{B}$. The fact that $\mathcal{B}$ is a Gröbner basis follows from Lemma 3.2. □

Suppose we have a module presentation $\langle Z \,||\, T \rangle$. Let $M$ be the free module on $Z$ and $M_0$ be the submodule generated by $T$. Every submodule of $M/M_0$ is $M_1/M_0$ for some submodule $M_1$ containing $M_0$ of $M$. If we have a finite set of generators for $M_1/M_0$, it is possible to find a module presentation for it. Suppose that $M_1/M_0$ is generated by $s$ elements. Then let $Z'$ be a set of $s$ elements. Let $M'$ be the free module on $Z'$. We can construct a $\mathbb{Z}[G]$-module homomorphism from $M'$ to $M/M_0$ by mapping each element in $Z'$ to a generator for $M_1/M_0$. It is possible to find a Gröbner basis $\mathcal{B}$ of the kernel of this epimorphism. A module presentation for $M_1/M_0$ would then be $\langle Z' \,||\, \mathcal{B} \rangle$.

Given a module presentation $\langle Z \,||\, T \rangle$. Let $M$ be the free module on $Z$, $M_0$ be the submodule generated by $T$ and $m$ be a polynomial array in $M$. It is possible to find the right ideal $\mathrm{Ann}_{\mathbb{Z}[G]}(m + M_0) = \{f \in \mathbb{Z}[G] \mid (m + M_0)f = 0\}$. Let $\varphi : \mathbb{Z}[G] \to M/M_0$ be the map taking 1 in $\mathbb{Z}[G]$ to $m + M_0$. Then $\mathrm{Ann}_{\mathbb{Z}[G]}(m)$ is exactly the kernel of $\varphi$.

Let $P = \{f_1, f_2, \ldots, f_s\}$ be a set polynomial arrays in $\mathbb{Z}[G]^k$. Let $M$ be the set $\{(g_1, g_2, \ldots, g_s) \mid \sum_{i=1}^{s} f_i g_i = 0\}$. Then $M$ is a right $\mathbb{Z}[G]$-submodule of $\mathbb{Z}[G]^s$ and is called the *module of syzygies* of the $s$-tuple $(f_1, f_2, \ldots, f_s)$. It is possible to describe $M$ by finding a Gröbner basis of $M$. Let $\varphi : \mathbb{Z}[G]^s \to \mathbb{Z}[G]^k$ be the map defined by $\varphi(g_1, g_2, \ldots, g_s) = f_1 g_1 + f_2 g_2 + \ldots + f_s g_s$. Then $\varphi$ is a $\mathbb{Z}[G]$-module homomorphism with kernel $M$. We know how to find a Gröbner basis of the kernel of $\varphi$ and this gives a Gröbner basis of $M$.

Suppose that $M_1$ and $M_2$ are submodules of $\mathbb{Z}[G]^k$ and $T_1$ and $T_2$ are finite sets of module generators for $M_1$ and $M_2$, respectively. In other words, we have module presentations for $\mathbb{Z}[G]^k/M_1$ and $\mathbb{Z}[G]^k/M_2$. It is possible to find a Gröbner basis of the submodule $M_1 \cap M_2$. Let $\varphi : \mathbb{Z}[G]^k \to \mathbb{Z}[G]^k/M_1 \times \mathbb{Z}[G]^k/M_2$ be the right $\mathbb{Z}[G]$-module homomorphism defined by $\varphi(h) = (h + M_1, h + M_2)$. Clearly the kernel of $\varphi$ is $M_1 \cap M_2$. Since we know how to find the kernel of $\varphi$, we can find a Gröbner basis of $M_1 \cap M_2$.

# Chapter 6

# Examples

In this chapter we look at a few results obtained from using the polycyclic quotient algorithm on some finite presentations. A C program called **pqa** is developed and the files of the program are stored on the computer disk at the back of this thesis. Some specifics about this implementation are described in the Appendix.

Due to the highly interactive nature of the program, an accurate timing and memory information cannot be obtained easily. To give the readers some ideas 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 *easy* implies that the problem needs about two megabytes of memory and less than five minutes of interaction time to finish. A rating of *normal* implies that the problem can be finished in thirty minutes using two megabytes of memory. A rating of *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 *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 *extremely hard*. In Appendix B we will see some factors affecting the performance of this program. It is possible that an extremely hard problem can be finished in about five minutes with a certain approach. It is also possible that an easy problem can become hard or very hard if unwise choices are made in certain steps of the program. Examples of such choices include choosing a wrong order in processing critical pairs, performing too much or too little full reduction and also choosing the wrong group presentation. Refer to Appendix B for more details. So the rating here can be somewhat misleading. It is the author's intention here to merely give a general feeling of the difficulty of each problem.

In this chapter, 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:**

1. For the first example we will look at the finite group $S_4$, the symmetric group 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 [Hal59]. The fact that $G$ is the free nilpotent group of rank two, class three is proved by Groves and the result appears in [HN76].

   We use **pqa** to compute the derived series of $G$. However, direct computation is not successful. The program **pqa** 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 **pqa**. 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 $G/N$. 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 found to be $\mathbb{Z}^2, \mathbb{Z}^2, 1$ in almost no time. This shows that $G'' \leq G^{(3)}N$. By inspecting the consistent polycyclic presentation for $G/G''$ we can see that $[b, a, a]$ is not in $G''$. Since $G'/G''$ is free abelian, $G'' \cap N = 1$. So $G'' = G^{(3)}$.

3. For the third example we look at the group $G$ presented by $\langle\, a, b, c \mid abcab = c, bcabc = a, cabca = b \,\rangle$ given to me by W. Nickel. The first quotient $G/G'$ is $\mathbb{Z}_2^2 \times \mathbb{Z}_4$ and the second quotient $G'/G''$ is $\mathbb{Z}^2 \times \mathbb{Z}_4$. The computation takes almost no CPU time. However, computing the third quotient $G''/G^{(3)}$ turns out to be rather difficult for **pqa**. By Tietze transformation, the above presentation can be simplified to $\langle\, a, b, c \mid a^2 = b^2 = c^2, abcab = c \,\rangle$. 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 **pqa**. The quotient $G''/G^{(3)}$ is $\mathbb{Z}$ and the quotient $G^{(3)}/G^{(4)}$ is 1. The rating for this problem is normal.

We will illustrate a few more results obtained from running **pqa** on the following presentations. The results are shown in a table form on the next page.

$$
\begin{aligned}
G_1 &= \langle\, a, b \mid [a^2, b, a], [a, b^2, a], [a, b, a^2], [b, a, b] \,\rangle \\
G_2 &= \langle\, a, b \mid [a, b, a], [b^2, a, b^2]\, [\, [b, a]^{b^3}, [b, a]^{b^{-4}}], [b, a]^2 \,\rangle \\
G_3 &= \langle\, a, b \mid [a, b, a], [b^2, a, b]\, [\, [b^5, a]^{b^7}, [b^{-1}, a^{-3}]^{b^{-7}}], [b, a]^2 \,\rangle \\
G_4 &= \langle\, a, b \mid [a, b, a], [b, a, b]\, [\, [b^2, a], [bab^2, a]\, ], [b, a]^2 \,\rangle \\
G_5 &= \langle\, a, b \mid [a, b, a], [b^2, a, b^2] \,\rangle \\
G_6 &= \langle\, a, b \mid [b, a, a]\, [b^2, a, [b, ab^2a]\, ], [b, ab^-1a] \,\rangle \\
G_7 &= \langle\, a, b \mid [a, b]^2, [a, b, b]\, [b, a, a], [b^{-1}, a^2, ab] \,\rangle \\
G_8 &= \langle\, a, b \mid [b, a, b]\, [b, a, a^{-1}, b], [b, a, a^2] \,\rangle \\
G_9 &= \langle\, a, b \mid [b, a^3], [b, a^2, b^2], [b, a, b^3], [b, a, b^2, a^2] \,\rangle \\
G_{10} &= \langle\, a, b \mid [b, a, b], [b, a^5] \,\rangle \\
G_{11} &= \langle\, a, b \mid [b, a, b], [b, a, a, a, a] \,\rangle \\
G_{12} &= \langle\, a, b \mid [b, a^3], [b, a^2, b^2], [b, a, b^3], [b, a, b^2, a, b, a], [b, a^2, b, a, b, a] \,\rangle
\end{aligned}
$$

In the following table, 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 **pqa** 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 $G_{11}$ cannot currently be computed. An entry in the column named *Rating* describes the level of difficulty in computing the most difficult quotient which is not given as 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.

| Group $G$ | $G/G'$ | $G'/G''$ | $G''/G^{(3)}$ | $G^{(3)}/G^{(4)}$ | Rating |
|---|---|---|---|---|---|
| $G_1$ | $\mathbb{Z}^2$ | $\mathbb{Z} \times \mathbb{Z}_2$ | 1 | 1 | Easy |
| $G_2$ | $\mathbb{Z}^2$ | $\mathbb{Z}_2^3$ | not polycyclic | ? | Easy |
| $G_3$ | $\mathbb{Z}^2$ | $\mathbb{Z}_2^2$ | not polycyclic | ? | Easy |
| $G_4$ | $\mathbb{Z}^2$ | $\mathbb{Z}_2$ | not polycyclic | ? | Easy |
| $G_5$ | $\mathbb{Z}^2$ | $\mathbb{Z}^3$ | not polycyclic | ? | Easy |
| $G_6$ | $\mathbb{Z}^2$ | $\mathbb{Z}$ | not polycyclic | ? | Easy |
| $G_7$ | $\mathbb{Z}^2$ | $\mathbb{Z}_2^3$ | $\mathbb{Z}^2$ | 1 | Normal |
| $G_8$ | $\mathbb{Z}^2$ | $\mathbb{Z}^2$ | 1 | 1 | Normal |
| $G_9$ | $\mathbb{Z}^2$ | $\mathbb{Z}^2$ | 1 | 1 | Normal |
| $G_{10}$ | $\mathbb{Z}^2$ | $\mathbb{Z}^4$ | not polycyclic | ? | Normal |
| $G_{11}$ | $\mathbb{Z}^2$ | $\mathbb{Z}^3$ | $\mathbb{Z} \times \mathbb{Z}_8 \times \mathbb{Z}_{32}$ | ? | Normal |
| $G_{12}$ | $\mathbb{Z}^2$ | $\mathbb{Z}^2$ | 1 | 1 | Very hard |

# Appendix A

# Implementation

In this appendix we outline some features of the **pqa** program. The program is written using the **gnu C** compiler and makes heavy use of the multiple precision integer package **gmp**. The **pqa** program takes as input a finite presentation for a group $H$ and attempts to compute the quotients $H/H^{(e)}$ for $e \geq 1$. The program will first compute the quotient $H/H'$, then $H/H''$. If $H/H''$ is polycyclic, then it will compute $H/H^{(3)}$. In general, if it has computed the quotient $H/H^{(e)}$ and the quotient is polycyclic, then it will attempt to find the quotient $H/H^{(e+1)}$. The program will exit if it finds out that the quotient $H/H^{(e+1)}$ is not polycyclic or if the quotient is equal to the quotient $H/H^{(e)}$. The version described in this thesis requires a lot of interaction between the user and the program.

## A.1 A Data Structure for Group Elements

In this chapter, $G$ is always a polycyclic group and $\mathbb{Z}[G]$ is the integral group ring of $G$. Let $a_1, a_2, \ldots, a_n$ be a polycyclic generating sequence of $G$. Every element $x$ in $G$ can be expressed uniquely as a collected element $a_n^{x_n} \ldots a_2^{x_2} a_1^{x_1}$. This representation gives a convenient data structure for group elements. We represent the element $x$ as an array of $n$ integers $x_n, \ldots, x_2, x_1$. In the program **pqa**, the data type **element** is defined to be such an array. Internally, as a pointer, **element** does not point to the head of the array but rather to the end of the array. In other words, if **u** is of **element** type, then **u[-1]** $= x_1$, **u[-2]** $= x_2, \ldots,$ **u[-n]** $= x_n$.

Recall from Section 2.4 that a consistent polycyclic presentation consists of relations with left sides $a_i a_j, a_i a_j^{-1}, a_i^{-1} a_j, (a_i^{-1})^{a_j^{-1}}, a_i^{m_i}$ and right sides their respective collected forms. To describe a consistent polycyclic presentation for a group $G$ with $n$ polycyclic generators, it is sufficient to store the right sides of these relations in some order. Each of these right sides can be viewed as an array of $n$ long integers. There are at most $4\binom{n}{2}$ conjugacy relations and at most $2n$ power relations. Therefore we need at most $4\binom{n}{2} + 2n$ arrays of size $n$ to store a consistent polycyclic presentation. Therefore we need $O(n^3)$ bytes to store such a presentation.

We have previously discussed two different group reduction orderings for infinite cyclic groups. In this implementation, for an infinite cyclic group generated by $a$, we use the group reduction ordering $a^0 < a^1 < a^{-1} < a^2 < a^{-2} < \ldots$. For a finite cyclic group with order $m$ and generator $a$, we use the group reduction ordering $a^0 < a^1 < a^2 < \ldots < a^{m-1}$. Let $\ll$ be the total ordering of the integers defined by $0 \ll 1 \ll -1 \ll 2 \ll -2 \ldots$.

For a polycyclic group $G$ with a polycyclic generators sequence $a_1, a_2, \ldots, a_n$, we define the group reduction ordering as follows. For $1 \leq i \leq n$, let $G_i$ be the subgroup generated by $a_i, a_{i+1}, \ldots, a_n$. Define $G_{n+1}$ to be the trivial subgroup. Then $G = G_1 \triangleright G_2 \triangleright G_3 \triangleright \ldots \triangleright G_{n+1} = 1$ is a subnormal series of $G$ with every quotient cyclic. Let $i$ be an integer, $1 \leq i \leq n$. Let $H_i$ be the quotient $G_i/G_{i+1}$. For each $i$, the set $Y_i$ of coset representatives of $G_{i+1}$ in $G_i$ is taken to be the set $\{a_i^j \mid j \in \mathbb{Z}\}$ if the quotient $H_i$ is cyclic, and the set $\{a_i^0, a_i^1, a_i^2, \ldots, a_i^{m-1}\}$ if $H_i$ is finite of order $m$. Each quotient $H_i$ is cyclic and we use the group reduction ordering for cyclic groups discussed in the paragraph above for each $H_i$. We can define the standard partial ordering $\preceq$ and the standard total ordering $\leq$ on $G$ relative to the parameters described. By Theorem 4.4.1, the standard total ordering $\leq$ is a group reduction ordering on $G$. For each element $x$ in $G$, the $Y$-form of $x$ is precisely the collected form of $x$, $a_n^{x_n} \ldots a_2^{x_2} a_1^{x_1}$. Given two elements $x$ and $x'$ of $G$. Let their respective collected forms be $a_n^{x_n} \ldots a_2^{x_2} a_1^{x_1}$ and $a_n^{x'_n} \ldots a_2^{x'_2} a_1^{x'_1}$. It is not hard to decide whether $x \preceq x'$ and whether $x \leq x'$. In fact, $x \preceq x'$ if for each $i$ with $1 \leq i \leq n$, $x_i$ and $x'_i$ has the same sign and $|x_i| \leq |x'_i|$. Also, $x < x'$ 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 $x_i \ll x'_i$.

## A.2 A Data Structure for Polynomials

The main algorithm used in **pqa** is a Gröbner basis algorithm for integral group rings of polycyclic groups described in Chapter 4. To use this algorithm, we need to store the set $P$ of polynomial arrays and the set $S$ of polynomial arrays to be reduced. Polynomial arrays can be described as a list of polynomials in $\mathbb{Z}[G]$. In this section, we will describe how polynomials are stored in the program **pqa**. We will use the same notation here as in the previous section.

Let $f$ be a nonzero polynomial in $\mathbb{Z}[G]$. Then $f$ can be written in the form $f_1 a_1^{p_1} + f_2 a_1^{p_2} + \ldots + f_r a_1^{p_r}$ where $f_1, f_2, \ldots, f_r$ are polynomials in $\mathbb{Z}[G_2]$ and $p_1 \gg p_2 \gg \ldots \gg p_r$ as integers. This form of writing $f$ suggests an inductive way to store $f$. Let us assume that 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, however, 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 $a_i^p$, in which case the field **gen** is $i$ and the field **power** is $p$. The fields **hpt** and **vpt** are pointers to other **cell**s 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. If a pointer points to a structure $c$, then the value of the pointer is given as $\&c$. 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 as follows.

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 ($n+1, 1, 0$,**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 ($1, 1, p_1$,**NULL**,&(**MP_INT** $c$));

else if $p_r = 0$ then begin

    let $c_1, c_2, \ldots, c_{r-1}$ be a list of $r-1$ **cell**s;

    for $i := 1$ to $r-2$ do begin

        if $f_i$ is a constant $c$, then the fields of $c_i$ are ($1, 1, p_i, \&c_{i+1}$,&(**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 ($1, 1, p_{r-1}, pf_r$,&(**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 $r$ **cell**s;

    for $i := 1$ to $r-1$ do begin

        if $f_i$ is a constant $c$, then the fields of $c_i$ are ($1, 1, p_i, \&c_{i+1}$,&(**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 ($1, 1, p_r$,**NULL**,&(**MP_INT** $c$));

    else the fields of $c_r$ are ($1, 0, p_r$,**NULL**,$pf_r$);

    let $pf$ point to $c_1$;

    end;

end.

Figure A.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 $a_i^{p_i}$ 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 imples 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 $a_1, a_2, a_3$.

To conclude this section, we remark that module elements in $\mathbb{Z}[G]^k$ is implemented as an array of $k$ polynomials. Again, let $G$ be the rank 3 free abelian group on the generators $a_1, a_2, a_3$. The module element $e_1 a_3 a_2^{-2} a_1 + 2e_1 a_2^2 - 3e_3$ in $\mathbb{Z}[G]^3$ is represented internally as an array of three polynomials. The first polynomial is $a_3 a_2^{-2} a_1 + 2a_2^2$. The second polynomial is the zero polynomial represented by a null pointer. The third polynomial is $-3$.

## A.3 Implementing Symmetrized Sets

In section 4.9, we saw an algorithm to compute a minimal Gröbner basis of a submodule. The notation we will use here will be the same as that section. 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.

What about the polynomial arrays in Symm($g$)? Let $g_i$ be in Symm($g$). If $g'$ is in $P'$ which is the set we are building up to be the Gröbner basis, then clearly $g'$ can be reduced to zero by $P'$. Therefore whenever a nonzero polynomial array $g$ comes up, in addition to adding $g$ to $P'$, we also add the polynomial arrays $g'$ to $P'$ for every $g'$ in Symm($g$). Notice that $g' = gx$ for some group element $x$ in $G$. Assume that $g$ has been stored using the method described in Section 2. 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 $x$.

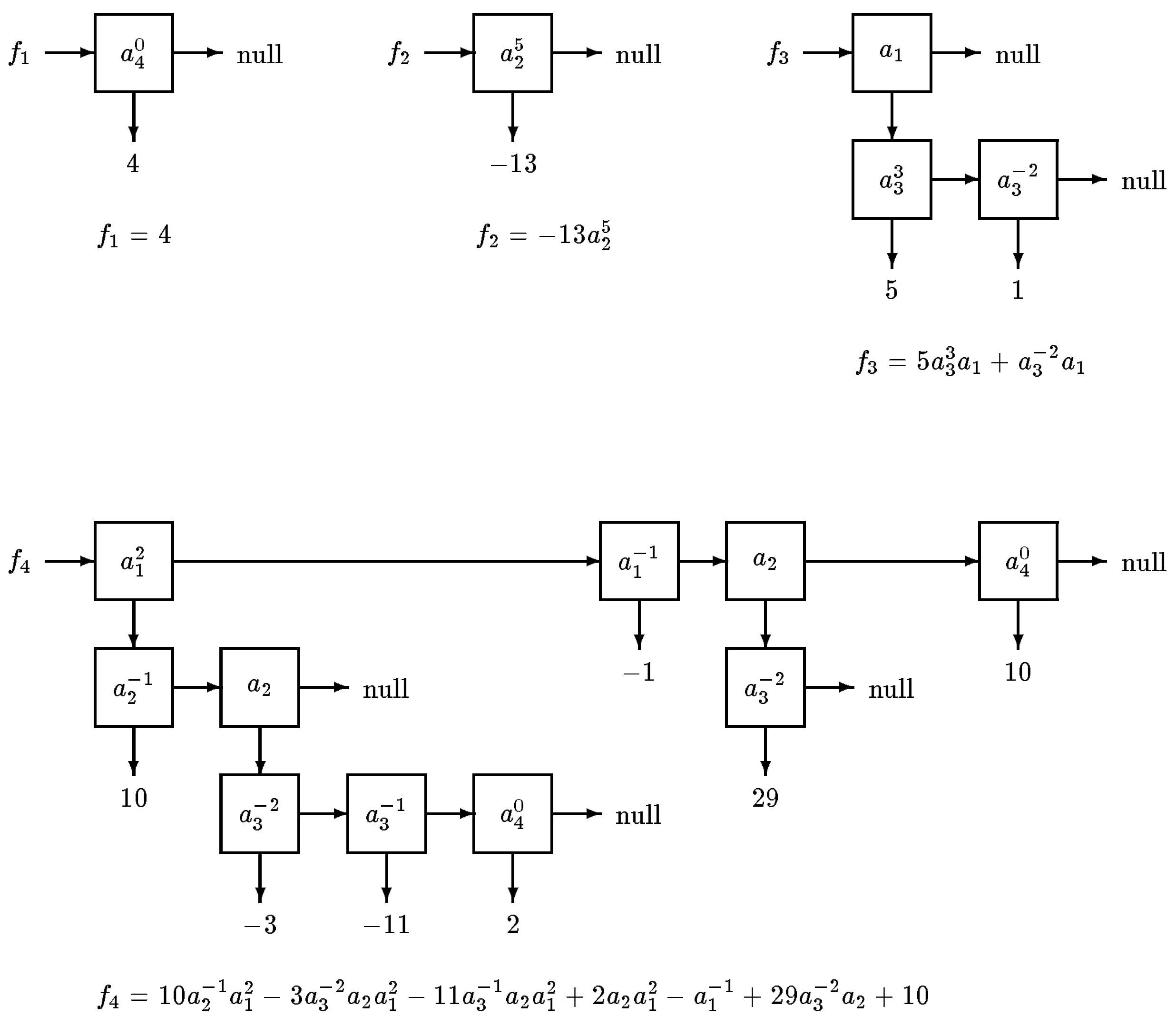


Figure A.1: Internal representation of some polynomials

This gives rise to the data structures **ModStorage** and **HTStorage** in the program.

The data structure **ModStorage** is designed to store a polynomial array $g$ together with some accounting information about $g$. The polynomial arrays in $P'$ are internally represented by the data structure **HTStorage**. A **HTStorage** has a pointer to a **ModStorage** and contains information which describes an element $x$ in $G$. If $g'$ is the polynomial array described by the **HTStorage** and $g$ is the polynomial array described by the **ModStorage**, then $g' = gx$. In the program, a **HTStorage** also stores the leading term of $g'$. In other words, we do not actually store $g'$ but simply a way to find $g'$.

An advantage of using such data structures is a saving in memory. A **HTStorage** typically needs much fewer bytes to store than a polynomial array. Instead of storing several copies of essentially the same polynomial array which may have thousands of terms, with these data structures we only store just one copy of the polynomial array and several **HTStorage**s instead. Since **HTStorage** only gives us a way to find $g'$ from $g$, we have to worry about the extra time needed to compute $g'$. It turns out that in most of the time when we need to use $g'$, the extra operation it will take is only one more multiplication of group elements.

# Appendix B

# Using the polycyclic quotient program

In this appendix, let us use the notation established in Chapter 3. The program **pqa** consists of 4 parts. Suppose that we are given a finite presentation for a group $G$ and an integer $k$ and that we have computed a consistent polycyclic presentation for $G/G^{(k)}$. The first part of the program finds and describes the left $\mathbb{Z}[G/G^{(k)}]$-module $M$. The second part finds a finite set of module generators for the submodule $M_0$ of $M$. The third part computes a Gröbner basis of $M_0$. The fourth part uses this Gröbner basis to decide whether or not $G/G^{(k+1)}$ is polycyclic and if the answer is yes, it constructs a consistent polycyclic presentation and a defining pair for $G/G^{(k+1)}$. Almost all of the time in this program is used on the third part. Complexity issues in Gröbner basis computations in polynomial rings over fields were discussed in the appendix of [BW91]. The authors cite a result from [MM84] that Gröbner basis computations have doubly exponential behavior in the number of variables. To complicate things, the Gröbner basis approach outlined in Chapter 4 requires the use of collections in polycyclic groups. Many computational group theorists believe that such collections have exponential run-time behavior. Therefore it is not a surprise that part three will take up most of the time.

From here on in this appendix, we will consider some basic strategies in computing a Gröbner basis from a set of polynomial arrays in a finitely generated free module of the integral group ring of a polycyclic group. Before we describe the strategies, let us look at a few parameters which affect the performance of **pqa**.

Define the *length* of a polynomial array $f$ to be the number of terms in $f$. Define the *weight* of $f$ to be the biggest coefficient of the terms in $f$ in absolute value. We have seen the leading coefficient of $f$ before. In this context, however, when we talk about the leading coefficient of $f$, we mean the absolute value of the leading coefficient. The phrase leading term will be used as defined before.

If we inspect the algorithm given in Section 4.9, we see that there are many places in which we have made choices. These choices will affect the performance of the Gröbner basis algorithm. In the while loop for reducing a polynomial array $g$ from $S$, it is very likely that there may be more than one polynomial array from $P'$ we can use to reduce $g$. With different choices of this polynomial array, we may arrive at different polynomial arrays $g$ at the end of the while loop. In the program **pqa**, to reduce $g$, we look at the polynomial arrays in $P'$ sequentially and if there is one which we can use to reduce $g$, we use it immediately, without looking ahead to see whether another such polynomial array exists. Suppose that both $f_i$ and $f_j$ can be used to reduce $g$. Usually we cannot tell which one will give a more useful polynomial array after the reduction. To make a very general statement, if $f_i$ has shorter length and lighter weight than $f_j$, then $f_i$ is more preferable to be used to reduce $g$. Usually, the polynomial array resulted from reducing $g$ by $f_i$ has shorter length and lighter weight than the one resulted from reducing by $f_j$. Even so, suppose that $g_1$ and $g_2$ are polynomial arrays resulted from reducing $g$ by two different ways and $g_1$ has shorter length and lighter weight than $g_2$. It is not entirely true that using $g_1$ as the result of reducing $g$ will speed up the algorithm any more than using $g_2$.

Here let us look at a slightly different method of reduction than the while loop in the algorithm. Suppose that $g$ is a polynomial array which is either equal to $\beta_{ij}$ for some $i, j$ or in $\mathrm{Symm}(f_i)$ for some $i$. To reduce $g$, we can find from $P'$ a list of polynomial arrays which have leading terms preceding the leading term of $g$. Let $c$ be the greatest common divisor of the leading coefficients of these polynomial arrays. Using this list of polynomial arrays, it is possible to find a polynomial array which is a sum of proper products, has leading coefficient $c$ and has leading term the same as the leading term of $g$. Using this polynomial array to reduce $g$ we can make sure that the result of the reduction $g_1$ either has the leading term vector smaller than the leading term vector of $g$ or has a positive leading coefficient smaller than $c$. Now if $g$ is equal to $\alpha_{ij}$ for some $i, j$, then Proposition 4.10.3 gives a similar method in reducing $g$. We need to take a more restrictive set of polynomial arrays but otherwise things work out very much the same. This approach of reducing polynomial arrays was not tested in this version of **pqa**, but it is a method worth implementing and being compared to the while loop in the algorithm in Section 4.9.

In the previous two paragraphs we are mostly concerned with ways to reduce the leading monomial of a polynomial array $g$. However, we can reduce other monomials in the standard form of $g$ also. Let $P''$ be a subset of $P'$. We may try to fully reduce $g$ with respect to $P''$. It is possible that after the full reduction, we will obtain a polynomial array $g'$ which has shorter length than $g$. One advantage of obtaining a short $g$ is that usually the symmetrized set of a short polynomial array has few polynomial arrays in it. However, in choosing $P''$, we have to be careful. In particular, if $f$ is a polynomial array in $P''$ with a huge length, then performing full reduction on $g$ using $f$ may yield a long polynomial array. In **pqa**, there is a parameter which can be changed by the user called **short-poly**. For every polynomial array $f$ in $P'$ with length at most **short-poly**, the program fully reduces $g$ by $f$. With the experimentation done so far, changing **short-poly** may have a rather drastic effect on the performance of **pqa**. If **short-poly** is set to be too big, say 2000, then the time to perform full reduction may be too long. In general, **pqa** works well when **short-poly** is smaller than 100.

Another important factor affecting the performance of the polycyclic quotient algorithm is the order to process the polynomial arrays in $S$. Let us use the phrase *to process a polynomial* $g$ to describe the process of reducing $g$ and forming new pairs as in the second while loop and the if statement in the algorithm given in Section 4.9. In Appendix A, we saw that we do not have to perform reductions for elements in $\mathrm{Symm}(f)$ for $f$ in $P'$ by adding extra polynomial arrays to the list of polynomial arrays $P'$. So we will just concentrate on reducing the polynomial arrays of the form $\alpha_{ij}$ and $\beta_{ij}$. Instead of actually computing these polynomial arrays and wasting memory to store them, we store the pair of indices $(i, j)$ where $i < j$ and compute them when they are needed. In the following paragraph, the phrase *processing the pair* $(i, j)$ is analogous to processing the polynomial arrays $\alpha_{ij}$ and $\beta_{ij}$.

In Section 4.10, we saw that it is not necessary to reduce $\beta_{ij}$ for some pairs of $(i, j)$.

In fact, it is possible to separate the pairs of indices into two different sets. The first set consists of all those pairs $(i,j)$ with $\mathrm{lm}(f_i) \preceq \mathrm{lm}(f_j)$ or $\mathrm{lm}(f_j) \preceq \mathrm{lm}(f_i)$. The second set consists of the remaining pairs. Internally, the data structures for the two sets are two stacks which store the pairs in the order that they are generated. Some experiments have been performed to compare the performance of a stack and a queue for storing the pairs and in most of the cases the stack wins out against the queue. Suppose that $(i,j)$ is a pair from the first stack and that $\mathrm{lm}(f_i) \preceq \mathrm{lm}(f_j)$. We may take $\alpha_{ij}$ to be $f_i$, which can clearly be weakly reduced. Therefore we only have to compute $\beta_{ij}$ for this pair. Moreover, after reducing $\beta_{ij}$, by Proposition 4.10.2, we have the option of deleting the polynomial $f_j$ from $P'$ to save memory because $f_j$ is not essential. Therefore we usually process the pairs from the first stack before processing the pairs from the second stack.

Let us define the *length* of a pair $(i,j)$ to be sum of the lengths of $f_i$ and $f_j$. Processing a pair with shorter length is usually faster than processing one with longer length. Also, the length of the polynomial array after reduction is in general shorter when we process a pair with shorter length. Therefore pairs with short length are placed at a higher priority than the longer ones. This implementation of the polycyclic quotient algorithm has a search routine to look for short pairs. This routine uses the parameters **pair_length2, pair_length3, incr2, incr3** to determine which pair to process.

Other attempts have been made to speed up the Gröbner basis algorithm in **pqa**. They include randomizing the polynomial arrays in $S$, establishing a third stack for those pairs $(i,j)$ with $\mathrm{lt}(f_i) = \mathrm{lt}(f_j)$ and setting up a **build** mode to store partial results from reduction. More study and experimentation are needed at this stage.

# References


[AL89] J. Apel and W. Lassner. An extension of Buchberger's algorithm and calculations in enveloping fields of Lie algebras. In Lorenzo Robbiano, editor, *Computational Aspects of Commutative Algebra*, pages 227–236. Academic Press, 1989.

[BCM81a] G. Baumslag, F.B. Cannonito, and C.F. III Miller. Computable algebra and group embeddings. *J. Algebra*, 69:186–212, 1981.

[BCM81b] G. Baumslag, F.B. Cannonito, and C.F. III Miller. Some recognizable properties of solvable groups. *Math. Z.*, 178:289–295, 1981.

[BCRS91] G. Baumslag, F.B. Cannonito, D.J.S. Robinson, and D. Segal. The algorithmic theory of polycyclic-by-finite groups. *J. Algebra*, 142:118–149, 1991.

[BKW74] A.J. Bayes, J. Kautsky, and J.W. Wamsley. Computation in nilpotent groups (application). In M.F. Newman, editor, *Proc. Second Internat. Conf. Theory Groups*, Lecture Notes in Mathematics 372, pages 82–89. Springer-Verlag, Berlin, 1974.

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Math. Inst. University of Innsbruck, Austria, 1965.

[BW91] T. Becker and V. Weispfenning. *Gröbner Bases: a Computational Approach to Commutative Algebra.* Berlin: Springer-Verlag, 1991.

[Can91] F.B. Cannonito. Some algorithms for polycyclic groups. In C.M. Cambell and E.F. Robertson, editors, *Groups St Andrews 1989*, volume 1 of *LMS Lecture Notes 159*, pages 76–83. Cambridge University Press, Cambridge, 1991.

[CFL58] K.T. Chen, R.H. Fox, and R.C. Lyndon. Free differential calculus IV. *Ann. Math.*, 68:81–95, 1958.

[Fel76] V. Felsch. A machine independent implementation of a collection algorithm for the multiplication of group elements. In R.D. Jenks, editor, *SYMSAC 76, Proc. 1976 ACM Symp. Symbolic and Algebraic Computation*, pages 159–166, New York, 1976. Assoc. Comput. Mach.

[FFG93] D. R. Farkas, C. D. Feustel, and E. L. Green. Synergy in the theories of Gröbner bases and path algebras. *Canadian Journal of Mathematics*, 45(4):727–739, 1993.

[GM88] R. Gebauer and H.M. Möller. On an installation of Buchberger's algorithm. *J. Symbolic Comp.*, 6:275–286, 1988.

[Hal54] P. Hall. Finiteness conditions for soluble groups. *Proc. London Math. Soc.*, 4(3):419–436, 1954.

[Hal59] M. Hall Jr. *The Theory of Groups.* Macmillan, New York, 1959.

[Hir38a] K.A. Hirsch. On infinite soluble groups, I. *Proc. London Math. Soc.*, 44(2):53–60, 1938.

[Hir38b] K.A. Hirsch. On infinite soluble groups, II. *Proc. London Math. Soc.*, 44(2):336–414, 1938.

[Hir46] K.A. Hirsch. On infinite soluble groups, III. *Proc. London Math. Soc.*, 49(2):184–194, 1946.

[Hir52] K.A. Hirsch. On infinite soluble groups, IV. *J. London Math. Soc.*, 27:81–85, 1952.

[Hir54] K.A. Hirsch. On infinite soluble groups, V. *J. London Math. Soc.*, 29:250–251, 1954.

[HJ82] J.R. Howie and D.L. Johnson. An algorithm for the second derived factor group. In C.M. Cambell and E.F. Robertson, editors, *Groups St Andrews 1981*, LMS Lecture Notes 71, pages 237–241. Cambridge University Press, Cambridge, 1982.

[HN76] G. Havas and T. Nicholson. Collection. In R.D. Jenks, editor, *SYMSAC '76, Proc. 1976 ACM Symp. Symbolic and Algebraic Computation*, pages 9–14, New York, 1976. Assoc. Comput. Mach.

[HN80] G. Havas and M.F. Newman. Application of computers to questions like those of Burnside. In J.L. Mennicke, editor, *Burnside Groups*, Lecture Notes in Mathematics 806, pages 211–230. Springer-Verlag, Berlin, 1980.

[HT95] D. Hartley and P. Tuckey. Gröbner bases in Clifford and Grassmann algebras. *J. Symbolic Comp.*, 20:197–205, 1995.

[Joh90] D.L. Johnson. *Presentations of Groups.* Cambridge University Press, 1990.

[Kha81] O.G. Kharlampovič. A finitely presented soluble group with insoluble word problem. *Izv. Akad. Nauk. Ser. Mat.*, 45:852–873, 1981.

[KR88] B.W. Kernighan and D.M. Ritchie. *The C Programming Language.* Prentice Hall, 1988.

[KRW90] A. Kandri-Rody and V. Weispfenning. Non-commutative Gröbner bases in algebras of solvable type. *J. of Symbolic Comp.*, 9:1–26, 1990.

[LG84] C.R. Leedham-Green. A soluble group algorithm. In M.D. Atkinson, editor, *Computational Group Theory*, pages 85–102. Academic Press, New York, 1984.

[LGS90] C.R. Leedham-Green and L.H. Soicher. Collection from the left and other strategies. *J. Symbolic Comp.*, 9:665–675, 1990.

[Mac73] I.D. Macdonald. Computer results on Burnside groups. *Bull. Australian Math. Soc.*, 9:433–438, 1973.

[Mac74] I.D. Macdonald. A computer application to finite p-groups. *J. Australian Math. Soc.*, 17:102–112, 1974.

[Mal58] A.J. Mal'cev. Homomorphisms into finite groups. *Ivanov. Gos. Ped. Inst. Učen. Zap.*, 29:49–60, 1958.

[MKD77] W. Magnus, A. Karrass, and Solitar D. *Combinatorial Group Theory.* Berlin: Springer-Verlag, 1977.

[MM84] H.M. Möller and T. Mora. Upper and lower bounds for the degree of Gröbner bases. In *EUROSAM '84, International Symposium on Symbolic and Algebraic Computation*, pages 172–183. Springer-Verlag, Berlin, 1984.

[Möl85] H.M. Möller. A reduction strategy for the Taylor resolution. In *Proc. EUROCAL 85*, pages 526–534. Springer-Verlag, Berlin, 1985.

[Mor86] T. Mora. Gröbner bases for noncommutative polynomial rings. In Jacques Calmet, editor, *Algebraic Algorithms and Error-Correcting Codes*, LNCS# 229, pages 353–362. Springer-Verlag, Berlin, 1986.

[Mor88] T. Mora. Standard bases and non-Noetherianity: non-commutative polynomial rings. In Thomas Beth and Michael Clausen, editors, *AAECC-4*, LNCS# 307, pages 98–109. Springer-Verlag, Berlin, 1988.

[Mor89] T. Mora. Gröbner bases in noncommutative algebras. In P. Gianni, editor, *ISSAC '88*, LNCS# 358, pages 150–161. Springer-Verlag, Berlin, 1989.

[Mor94] T. Mora. An introduction to commutative and non-commutative Gröbner bases. *Theoretical Computer Science*, 134(1):131–174, 1994.

[New76a] M.F. Newman. Calculating presentations for certain kinds of quotient groups. In R.D. Jenks, editor, *SYMSAC 76, Proc. 1976 ACM Symp. Symbolic and Algebraic Computation*, pages 2–8, New York, 1976. Assoc. Comput. Mach.

[New76b] M.F. Newman. A computer aided study of a group defined by fourth powers. *Bull. Australian Math. Soc.*, 14:293–297, 1976.

[Nic93] W. Nickel. *Central Extensions of Polycyclic Groups.* PhD thesis, Australian National University, 1993.

[Pas77] D. Passman. *The Algebraic Structure of Group Rings.* John Wiley and Sons, Inc., 1977.

[Ple87] W. Plesken. Towards a solvable quotient algorithm. *J. Symbolic Comp.*, 4:111–122, 1987.

[Seg83] D. Segal. *Polycyclic Groups.* Cambridge University Press, 1983.

[Seg90] D. Segal. Decidable properties of polycyclic groups. *Proc. London Math. Soc.*, 61(3):497–528, 1990.

[Sim87] C.C. Sims. Verifying nilpotence. *J. Symbolic Comp.*, 3:231–247, 1987.

[Sim90] C.C. Sims. Implementing the Baumslag-Cannonito-Miller polycyclic quotient algorithm. *J. Symbolic Comp.*, 9:707–723, 1990.

[Sim93] C.C. Sims. Complexity issues in infinite group theory. *DIMACS Series in Discrete Mathematics*, 11:277–286, 1993.

[Sim94] C.C. Sims. *Computation with Finitely Presented Groups.* Cambridge University Press, 1994.

[SW73a] T.W. Sag and J.M. Wamsley. Minimal presentations for groups of order $2^n$, $n \leq 6$. *J. Australian Math. Soc.*, 15:461–469, 1973.

[SW73b] T.W. Sag and J.M. Wamsley. On computing the minimal number of defining relations for finite groups. *Math. Comp.*, 27:361–368, 1973.

[Tie08] H. Tietze. Ueber die topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten. *Monatsh. für Math. und Phys.*, 19:1–118, 1908.

[VL84] M.R. Vaughan-Lee. An aspect of the nilpotent quotient algorithm. In Michael D. Atkinson, editor, *Proceedings of the London Math. Soc. Symp. on Computational Group Theory*, pages 75–84. Academic Press, London, 1984.

[VL90] M.R. Vaughan-Lee. Collection from the left. *J. Symbolic Comp.*, 9:725–733, 1990.

[Wam74] J.M. Wamsley. Computation in nilpotent groups (theory). In M.F. Newman, editor, *Proc. Second Internat. Conf. Theory Groups*, pages 691–700. Berlin: Springer-Verlag, 1974.

[Wam77] J.W. Wamsley. Computing soluble groups. In *Conference on the theory of groups (Canberra 1975)*, Lecture Notes in Mathematics 573. Springer-Verlag, 1977.

[Wei92] V. Weispfenning. Finite Gröbner bases in non-Noetherian skew polynomial rings. In *ISSAC '92*, pages 329–333. ACM, 1992.

# Vita

## Eddie H. Lo

**1968** Born in Hong Kong on April 9.

**1979** Graduated from St. James' Primary School, Hong Kong.

**1986** Graduated from St. Paul's Co-educational College, Hong Kong.

**1986-87** Attended University of Hong Kong, Hong Kong. Majored in Mechanical Engineering.

**1987-89** Attended University of Hawaii, Manoa, Hawaii. Majored in Mathematics and Computer Science.

**1989** B.Sc., University of Hawaii at Manoa.

**1990** Graduate work in Mathematics, University of Hawaii, Manoa, Hawaii.

**1990-96** Graduate work in Mathematics, Rutgers, The State University of New Jersey, New Brunswick, New Jersey.

**1990-92** Receipient of National Needs Fellowship, Department of Mathematics, Rutgers, The State University of New Jersey.

**1992-93** Teaching Assistant, Department of Mathematics, Rutgers, The State University of New Jersey.

**1993-94** Research Assistant, Department of Mathematics, Rutgers, The State University of New Jersey.

**1994-96** Teaching Assistant, Department of Mathematics, Rutgers, The State University of New Jersey.

**1996** Ph.D. in Mathematics, Rutgers, The State University of New Jersey.