Cryptanalysis of a Homomorphic Public-Key Cryptosystem

Su-Jeong Choi

Technical Report
RHUL–MA–2006–7
21 September 2006

Royal Holloway
University of London

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

CRYPTANALYSIS OF A HOMOMORPHIC PUBLIC-KEY CRYPTOSYSTEM

Su-Jeong Choi

Royal Holloway and Bedford New College,
University of London

Thesis submitted to
The University of London
for the degree of
Doctor of Philosophy
July 2006.

Abstract

The aims of this research are to give a precise description of a new homomorphic public-key encryption scheme proposed by Grigoriev and Ponomarenko [7] in 2004 and to break Grigoriev and Ponomarenko homomorphic public-key cryptosystem. Firstly, we prove some properties of linear fractional transformations. We analyze the X_n-representation algorithm which is used in the decryption scheme of Grigoriev and Ponomarenko homomorphic public-key cryptosystem and by these properties of the linear fractional transformations, we correct and modify the X_n-representation algorithm. We implement the modified X_n-representation algorithm by programming it and we prove the correctness of the modified X_n-representation algorithm. Secondly, we find an explicit formula to compute the $X(n, S)$-representations of elements of the group Γ_n. The $X(n, S)$-representation algorithm is used in the decryption scheme of Grigoriev and Ponomarenko homomorphic public-key cryptosystem and we modify the $X(n, S)$-representation algorithm. We implement the modified $X(n, S)$-representation algorithm by programming it and we justify the modified $X(n, S)$-representation algorithm. By these two modified X_n-representation algorithm and $X(n, S)$-representation algorithm, we make its decryption scheme more efficient. Thirdly, by using those properties of the linear fractional transformations, we design new X_1-representation algorithms I and II and we mainly use these two X_1-representation algorithms to break Grigoriev and Ponomarenko homomorphic public-key cryptosystem. We implement the algorithms by programming them and we prove the correctness of these two algorithms. Fourthly, we analyze Grigoriev and Ponomarenko homomorphic public-key cryptosystem and we give a clear description of Grigoriev and Ponomarenko scheme with a practical example. We also consider implementation issues for its practical applications. Lastly, we show several attack methods with examples and experiments according as the attack methods and so we break Grigoriev and Ponomarenko homomorphic public-key cryptosystem.

Acknowledgments

My first thanks go to my supervisors, Professor Peter Wild and Professor Simon Blackburn, for their guidance and useful comments during my PhD study at Royal Holloway.

Moreover, I thank to all the members of Information Security Group and Department of Mathematics, and all my friends who have made enjoyable time.

My studies have been supported partially by Royal Holloway College Scholarship.

Most of all, I thank my family. My family have provided full financial support for my study. Without my family's support, this course of study would never have been possible.

I dedicate all my efforts to my father and mother.

Contents

List of Tables

Chapter 1

Introduction

In this thesis, we analyze a new homomorphic public-key encryption scheme introduced by Grigoriev and Ponomarenko [7] in 2004 and we give a precise description of Grigoriev and Ponomarenko homomorphic public-key cryptosystem. Next, we break Grigiriev and Ponomarenko homomorphic public-key cryptosystem. The main stream of thesis consists of four parts. The first part is a background required to comprehend Grigoriev and Ponomarenko homomorphic public-key cryptosystem. The second part is about three representation algorithms. Two X_n-representation algorithm and $X(n, S)$-representation algorithm are used in the decryption scheme of Grigoriev and Ponomarenko homomorphic public-key cryptosystem and new X_1-representation algorithms are used for cryptanalysis of Grigoriev and Ponomarenko homomorphic public-key cryptosystem. The third part is a description of Grigoriev and Ponomarenko homomorphic public-key cryptosystem. The last part is cryptanalysis of Grigoriev and Ponomarenko homomorphic public-key cryptosystem.

In Chapter 1, we give the structure of thesis and we mention the main parts which are handled in each chapter.

In Chapter 2, we survey symmetric-key cryptography and public-key cryptography. In particular, we focus on public-key cryptography. Moreover, because this new homomorphic public-key encryption is a probabilistic encryption scheme, we study deterministic encryption and probabilistic encryption. As the security of Grigoriev and Ponomarenk homomorphic public-key cryptosystem relies on the difficulty of the membership problem for a group, we survey computational problems and decision problems.

In Chapter 3, we study combinatorial group theory because the message space of Grigoriev and Ponomarenko homomorphic public-key encryption scheme is a finitely presented group and the word problem is implicitly related to the decryption scheme of Grigoriev and Ponomarenko homomorphic public-key cryptosystem. We also survey normal forms in connection with the word problem. In addition, we study matrix group theory such as general linear groups, special linear groups and modular groups because the ciphertext space of Grigoriev and Ponomarenko homomorphic public-key cryptosystem is a subgroup of a modular group.

In Chapter 4, we prove some properties of linear fractional transformations. By these properties, we correct and modify the X_n-representation algorithm used in the decryption scheme of Girgoriev and Ponomarenko homomorphic public-key cryptosystem. We implement the modified X_n-representation algorithm by programming it and we prove the correctness of the modified X_n-representation algorithm.

In Chapter 5, we analyze the $X(n, S)$-representation algorithm which is used in the decryption scheme of Grigoriev and Ponomarenko homomorphic public-key cryptosystem and we modify the $X(n, S)$-representation algorithm. We implement the modified $X(n, S)$-representation algorithm by programming it and we prove the correctness of the modified $X(n, S)$-representation algorithm.

In Chapter 6, we design new X_1-representation algorithms I and II to represent a subgroup of the modular group. We implement these two X_1-representation algorithms by programming them and we prove the correctness of the algorithms respectively.

In Chapter 7, we give a precise description of Girgoriev and Ponomarenko homomorphic public-key cryptosystem. Moreover, we demonstrate its practical implementation with an example and we compare Grigoriev and Ponomarenko' description with our description.

In Chapter 8, we show several attack methods to break Grigoriev and Ponomarenko homomorphic public-key cryptosystem with examples and experiments according as the attack methods.

Chapter 2

Public-Key Cryptography

In this chapter, we survey some subjects related to public-key cryptography, based on [7], [14] and [16].

In Section 2.1, we survey both symmetric-key cryptography and public-key cryptography as general cryptographic techniques. In Section 2.2, we give two examples which are the most well-known two public-key cryptosystems, RSA encryption and ElGamal encryption. In Section 2.3, we study computational problems and decision problems which are in connection with the security issues of public-key cryptosystems. In Section 2.4, we study deterministic encryption and probabilistic encryption and then we make a comparison between them.

2.1 Background

Cryptography is a study of encryption and decryption technologies. In other words, it is a science of securing information by coding so that it can be read only by those with authentication or permission. Cryptography makes extensive use of mathematics, particularly discrete mathematics including topics from combinatorics, statistics, information theory, computational complexity and number theory. So cryptography is a study of mathematical techniques related to aspects of information security.

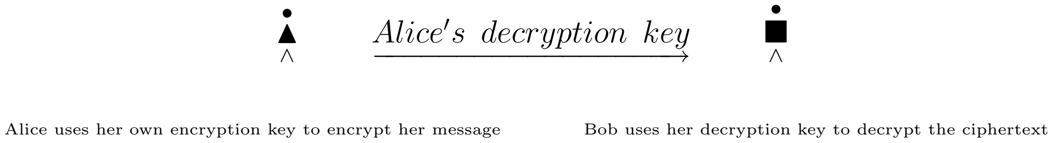
As its applications, in general, it is used to protect national secrets and it is associated with the military, the diplomatic service and government. In addition, it can be used in a computer or computer network to secure information in a website and also to protect financial information such as credit card information, during financial transactions. So it covers a wide range of security issues in the transmission and protection of information such as massive file storage, electronic commerce through public networks.

There are two kinds of cryptosystems, symmetric-key cryptosystems and public-key cryptosystems. If the encryption key is equal to the decryption key, then the cryptosystem is called a symmetric-key cryptosystem, whereas a public-key cryptosystem uses two distinct encryption key and decryption key and then the computation of the decryption key from the encryption key is infeasible. In this case, the encryption key is public and decipher keeps the decryption key secret.

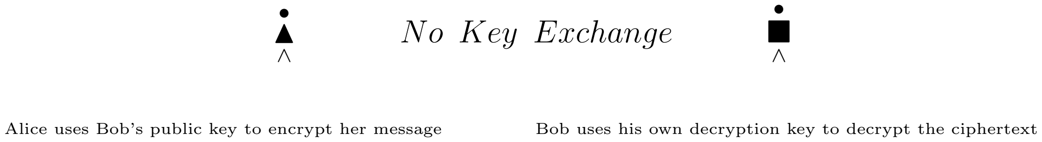
Until the late 1970's, all cryptographic message transmission was by symmetric key and it is used for the military and diplomatic purposes. The advantages of symmetric-key cryptography are keys for encryption are relatively short and thus symmetric-key cryptography is efficient for encryption. Symmetric-key cryptography can also be used for various cryptographic mechanisms including pseudorandom number generators, hash functions, and computationally efficient digital signature schemes as primitives.

The disadvantage of a symmetric cryptosystem, Alice and Bob should exchange the secret key before they start the communication. Someone that knows the encryption key can obtain the corresponding decryption key and so secure key exchange is a main problem. Therefore one of the major issues with symmetric-key cryptosystems is to find an efficient method to agree on and exchange keys securely. This problem is referred to as the key distribution problem. In a large network, there are many key pairs to be managed and thus the trust third party needs efficient key management.

Symmetric-Key Cryptosystem


Alice's decryption key
Alice uses her own encryption key to encrypt her message
Bob uses her decryption key to decrypt the ciphertext

Public-Key Cryptosystem


No Key Exchange
Alice uses Bob's public key to encrypt her message
Bob uses his own decryption key to decrypt the ciphertext

The advantage of public-key cryptosystems is that they do not need key exchange. When Alice sends a message to Bob, Alice uses his public key for encryption and Bob can decrypt the message by using his private key. So in case of a public-key cryptosystem, since Alice and Bob have no common shared key and only the decryption key is secret, the key management is simple. Therefore in a large network, the number of keys necessary may be considerably smaller than in a symmetric-key cryptosystem and public-key encryption scheme may be used to establish a key for a symmetric-key cryptosystem. Another advantage of public-key cryptosystems is that the public-key cryptosystem provides a digital signature scheme which can not be repudiated. The digital signature keeps the original entity from denying their data.

The disadvantage of public-key cryptosystems is the computational performance for encryption and decryption. So public-key encryption schemes are much slower than symmetric-key encryption schemes and thus public-key cryptosystems are much less efficient than symmetric-key cryptosystems.

In practice, we combine a symmetric-key cryptosystem and a public-key cryptosystem. For instance, Alice encrypts the message by using the session key which Alice generates and she also encrypts the session key by using Bob's public key. Then Bob decrypts the session key by using his private key and decrypts the ciphertext by the session key. Public-key encryption schemes are most commonly used in practice for the transport of keys used for data encryption by symmetric-key encryption schemes and other applications such as data integrity, authentication, credit card numbers and PIN numbers. Furthermore, the most application of a public-key cryptosystem is confidentiality without key exchange, that is, a message which Alice encrypts by using Bob's public key can only be decrypted by Bob's private key. Public-key digital signature algorithms can be used for sender's authentication. Those properties of public-key cryptosystems are useful for many applications such as digital cash, password-authenticated key agreement, multi-party key agreement and so on.

2.2 Encryption Schemes

We describe the most well-known two public-key cryptosystems, called RSA cryptosystem and ElGamal Cryptosystem. The RSA public-key encryption was invented in 1978 by Rivest, Shamir and Adleman and it provides both privacy and authentication. Moreover, the RSA encryption can be found in Microsoft Window, Netscape Navigator, Apple and Sun and it is also used for electronic cash. Its security is based on the intractability of the integer factorization problem and there is no efficient algorithms known for this problem. The ElGamal cryptosystem is related to the Diffie-Hellman key exchange in 1976. Diffie-Hellman key exchange has the fact that it is easy to calculate powers in modular arithmetic, but difficult to compute logarithms. It means that it takes considerable running time and cost to compute discrete logarithms relative to the calculation of powers. So the security of ElGamal cryptosystem is based on the intractability of the discrete logarithm problem and the Diffie -Hellman problem in ${Z_p}^*$ and there is no known polynomial-time algorithm to solve discrete logarithm problem.

RSA Encryption

We first explain how the RSA encryption scheme works as follows : let the n-bit integer $N = pq$ be the product of two large primes p and q of the same size. Let e and d be two integers satisfying $ed \equiv 1 \bmod \varphi(N)$ where $\phi(N) = (p-1)(q-1) = N + 1 - (p+q)$ is the Euler ϕ function of N. These integers N, e, d are called, respectively, the RSA modulus, the encryption exponent, and the decryption exponent where N and e are the public key and d is the secret key. To encrypt a message m, the sender Alice computes the ciphertext c, which is the least positive residue of m^e modulo N. To decrypt c, the receiver Bob computes the least positive residue of c^d modulo N and then $c^d \equiv m^{ed} \equiv m \pmod{N}$.

Example

Bob chooses $p = 97$ and $q = 83$. Then $n = 8051$ and $\phi(n) = 96 \times 82 = 7872$. Bob chooses $e = 3221$ and he computes the inverse d of 3221 mod 7872. Then Bob has $d = 2813$. Hence, his public key is $n = 8051$ and $e = 3221$ and the secret key is $d = 2813$. Now, Alice encrypts the plaintext 7326 and she computes

$$7326^{3221} \bmod 8051 = 4816$$

and send the ciphertext 4816 to Bob and he decrypts the ciphertext 4816. Bob uses his secret key $d = 2813$ to compute

$$4816^{2813} \bmod 8051 = 7326$$

and then Bob obtains the plaintext 7326. □

Although we know $n = pq$, we can not compute p and q. Hence it is computationally infeasible to find $\varphi(n) = (p-1)(q-1)$ and thus we can not determine d satisfying $ed \equiv 1 \ (\text{mod}\varphi(n))$. Therefore the security of RSA is based on the intractability of the integer factorization problem. The RSA encryption scheme is slow relative to other cryptosystems, roughly 100 to 1000 times slower than DES.

ElGamal Encryption

Now we give another example of a public-key cryptosystem, called ElGamal encryption. We describe the encryption scheme as follows : Bob generates a large random prime p and a generator g. Bob chooses a random x from $\mathbb{Z}_p{}^*$ and computes $h = g^x$. Then Bob publishes h with p and g as his public key. Bob retains x as his secret key. Alice chooses a random y from $Z_p{}^*$ and calculates $c_1 = g^y$ and $c_2 = mh^y$. Alice sends the ciphertext (c_1, c_2) to Bob. Then Bob decrypts a ciphertext (c_1, c_2) with his secret key x by computing

$$c_2(c_1^x)^{-1} = \tfrac{mh^y}{g^{xy}} = \tfrac{mg^{xy}}{g^{xy}} = m$$

as the plaintext message.

Example

Let $p = 2111$ and $g = 2$ is a primitive element modulo p. Let $x = 321$. Then

$$h = 2^{321} \bmod 2111 = 1233.$$

Alice sends a message $m = 1382$ to Bob and she chooses a random integer $y = 423$. Then Alice computes

$$c_1 = 2^{423} \bmod 2111 = 695$$

and

$$c_2 = 1382 \times 1233^{423} \bmod 2111 = 252.$$

Thus the ciphertext is $(c_1, c_2) = (695, 252)$ and Bob decrypts

$$m = 252 \times \left(695^{321}\right)^{-1} \bmod 2111 = 1382.$$

This is the plaintext. □

2.3 Hard Problems

We survey hard problems which most of public-key cryptosystems rely on in connection with security issues. We first introduce integer factorization problem and discrete logarithm problem as the most popular computational problems.

Definition 2.3.1 Integer Factorization Problem
Given a positive integer n, find its prime factorization $n = {p_1}^{e_1}{p_2}^{e_2}\cdots{p_k}^{e_k}$ where the p_i are pairwise distinct primes and each $e_i \geq 1$.

The security of most of public-key cryptosystems depends on the intractability of the integer factorization problem and it has been studied intensively for the past 20 years. So far, the most efficient algorithm to factorize an integer is the general number field sieve method.

Definition 2.3.2 Discrete Logarithm Problem
Let G be a group and $\alpha, \beta \in G$. Find an integer x if it exists such that $\beta = \alpha^x$ in G.

Another computational number theoretic problem that is widely believed to be intractable is that of extracting discrete logarithms in a finite field. The discrete logarithm problem is defined in $\langle \alpha \rangle \subset G$ so it is a problem about cyclic groups. Finding discrete logarithms is difficult, but the inverse operation of exponentiation can be computed efficiently by using the square and multiply method. If the discrete logarithm problem is hard, then we have a one-way function and it is fast computation $\beta = \alpha^x$ but in general, it is difficult to compute x. Exponentiation in other groups is also a reasonable candidate for a one-way function supposing that the discrete logarithm problem for the group is hard. For instance, the discrete logarithm problem is hard in the group of points on an elliptic curve.

Form now, we survey some decision problems as hard problems used in cryptographic settings. A decision problem is a problem with a yes or no answer, that is, a function whose range is two values, such as $0, 1$.

Definition 2 3.3 Membership Problem
Let G be a group and $X \subset G$ be a finite set. For a given $g \in G$, test whether $g \in \langle X \rangle$.

In other words, the problem of deciding whether a given element g of the group G belongs to a fixed subgroup $\langle X \rangle$ is called the membership problem for $\langle X \rangle$ in G. Note that group membership problems tend to be undecidable.

Definition 2.3.4 Let G be a group and let $X \subset G$ be a finite set. An X-representation of $g \in \langle X \rangle$ is a product of elements from X and their inverses that is equal to g, that is, $g = {x_1}^{a_1}{x_2}^{a_2} \cdots {x_k}^{a_k}$ where $\langle X \rangle$ is a subgroup generated by X, $x_i \in X \cup X^{-1}$ and $a_i \in \mathbb{Z}$.

Generally any element of $\langle X \rangle$ has at least one X-representation with respect to G but not necessary the unique one. However, if G is a free group on X, then each element of G has the unique X-representation as an irreducible word.

Definition 2.3.5 Representation Problem

Let G be a group and $X \subset G$ be a finite set. For $g \in \langle X \rangle$, find an X-representation of g where $\langle X \rangle$ is a subgroup generated by X.

The representation problem consists in finding a certificate for the membership problem. For instance, if $G = \mathbb{F}^*$ is the multiplicative group of a finite field $\mathbb{F}$ and $X = \{g\}$ where g is a generator of the group $\mathbb{F}^*$, then the representation problem coincides with the discrete logarithm problem. It is remarked that the representation problem is NP-hard in average in general even if G is a free group with a finite rank.

Definition 2.3.6 Word Problem

Given two words over the alphabet, decide whether they represent the same element of the group.

The word problem for groups is the problem of deciding whether two given words of a presentation of a group represent the same element. In fact, there exists no general algorithm for this problem in a general group. It is an important fact that the decidability and the complexity of the word problem of a finitely generated group depend on the group and not on the generators or the presentation chosen. In other words, if G has decidable word problem for some finite generating set X, then G has decidable word problem for every finite generating set.

The word problem is only concerned with finitely presented groups in Section 3.2. A word is a product of generators, and two such words may denote the same element of the group even if they appear to be different because by using the group axioms and the given relations it may be possible to transform one word into the other. The problem is to find an algorithm which for any two given words decides whether they denote the same group element. The effect of the relations in G is to make various strings that represent the same element of G. In fact the relations provide a list of strings that can be either introduced where we want, or canceled out whenever we see them, without changing the group element that is the result of the multiplication. In the worst case, the relation between strings says they are equal in G is not decidable.

2.4 Probabilistic Encryption

We introduce two kinds of algorithms, deterministic algorithm and probabilistic algorithm which are used in cryptographic encryption schemes and then we discuss deterministic encryption and probabilistic encryption.

Definition 2.4.1 Deterministic Algorithm
A deterministic algorithm depends on its input data alone.

If a deterministic algorithm runs repeatedly with the same input data, it will always proceed in the same way and so its complexity provides an accurate and consistent estimate of its time and space requirements.

Most of cryptosystems based on the number theory to transmit a message are deterministic. For a given plaintext, anybody can take the same ciphertext, that is, under a fixed public key, a probabilistic plaintext m is always encrypted to the same ciphertext c. For example, RSA, Rabin and Knapsack encryption schemes are deterministic. So its disadvantage is if an attacker knows that the plaintext belongs to a small set, then the attacker can encrypt all possibilities in order to determine which is the supposedly secret message. The RSA, Rabin and knapsack encryption schemes are deterministic because a plaintext is always encrypted to the same ciphertext. Another disadvantage is that it is easy to detect when the same message is sent twice. Hence, deterministic encryption can leak information to an attacker. Especially, because in a public-key cryptosystem, anyone can encrypt chosen messages using a public key, the attacker can build a large dictionary of useful plaintxet and ciphertext pairs and then observe the encrypted channel for matching ciphertexts.

Because of these drawbacks of the deterministic encryption, cryptographers proposed probabilistic encryption. Probabilistic encryption was introduced by Goldwasser and Micali in 1982 and it uses randomness to attain a provable and very strong level of security. Hence, in order that we do not leak even partial information about the plaintext, the encryption must be probabilistic. In particular, when we use the public-key cryptosystem, the probabilistic encryption is important because the plaintext corresponds to many different ciphertexts. For example, ElGamal encryption scheme is one of many encryption schemes which use randomization in the encryption process. ElGamal cryptosystem is efficient probabilistic encryption scheme.

Definition 2.4.2 Randomized Algorithm

A randomized algorithm makes use of a random number generator during its execution.

In general, many algorithms in computational group theory depend on making some random choices such as choosing random elements of groups. A randomized algorithm does not depend on its input data alone and its performance may vary from one run to another run with the same input. Thus, the complexity is the average running time and space requirements of the algorithm under the assumption the random number generator being used is working properly and is capable of choosing genuinely random integers within a given range $[a, b]$. Moreover, deterministic algorithms follow the same execution path (sequence of operations) each time they execute with the same input. By contrast, a randomized algorithm makes random decisions at certain points in the execution. Hence its execution paths may differ each time and the random decisions are based on the outcome of random number generator.

Chapter 3

Combinatorial Group Theory

This chapter is based on [9] and [14].

In many cases groups arise by means of presentations. A presentation of a group G consists of a set of generators of G with a collection of relations among these generators such that any other relation among the generators is derived from the given relations. Combinatorial group theory is the study of groups given by presentations. Since authors used a presentation of a finite group as its message space of Grigoriev and Ponomarenko homomorphic public-key cryptosystem, we study combinatorial group theory which covers these topics.

In Section 3.1, we study free groups and finitely generated free groups because all free groups used in Grigoriev and Ponomarenko homomorphic public-key cryptosystem are finitely generated. In Section 3.2, we study finitely presented groups and survey normal forms of a finitely presented group in connection with the word problem. In Section 3.3, we survey general linear groups, special linear groups and modular groups related to the ciphertext space of Grigoriev and Ponomarenko homomorphic public-key cryptosystem.

3.1 Free Groups

We begin with some definitions in connection with a concept of a free group. $X^{\pm}$ denotes $X \cup X^{-1}$ and elements of $X^{\pm}$ are called letters.

Definition 3.1.1 A word in X is a finite sequence of letters, $w = a_1 \cdots a_n$, $n \geq 0$ where $a_i \in X^{\pm}$. If $n = 0$, then $w = 1$ called the empty word.

Definition 3.1.2 The length $|w|$ of $w = a_1 \cdots a_n$ where $a_i \in X^{\pm}$ is $|w| = n$.

Theorem 3.1.3 The set W_X of all words in X is a semigroup under juxtaposition.

Definition 3.1.4 A word is said to be reduced or irreducible if it does not contain a subword of the form xx^{-1} or of the form $x^{-1}x$.

Let $X = \{x_1, x_2, \cdots, x_t\}$ be a set of symbols where X need not be countable or ordered. A word on $X^{\pm}$ means an expression of the form ${x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2} \cdots {x_{a_m}}^{\epsilon_m}$ where $a_i \in \{1, 2, \cdots, t\}, \epsilon_i = \pm 1$, $x_{a_i} \in X$ and x_{a_i}s are not necessarily distinct symbols. That is, a word is a string of elements of X with exponents either $+1$ or -1. So a free group F on X is identified with a subset of W_X consisting of all irreducible words and the identity of F is the empty word $1_X \in W_X$. The words of the free group are like the names of the elements of the free group. Successive deletion of parts xx^{-1} or $x^{-1}x$ from any word w must lead to a reduced word. This determines an equivalence relation on W_X and each equivalence class has a unique representative which is a reduced word. Define $\bar{w}$ to be the reduced word corresponding to w. Multiplication in F is defined by concatenation followed by a reduction to a reduced word. Now we give another definition of a free group in terms of a free basis.

Definition 3.1.5 A subset X of a group F is said to be a free basis for F if, for every function $\varphi : X \to G$ can be extended uniquely to a homomorphism $\bar{\varphi} : F \to G$ such that for every $x \in X$, $\bar{\varphi}(x) = \varphi(x)$.

A group F is said to be a free group if there is some subset which is a free basis for F. For example, the additive group of integers $\mathbb{Z}$ is a free group with either of the singleton sets $\{1\}$ or $\{-1\}$ as a free basis. The following results are well known.

Theorem 3.1.6 Let X be a set. Then there exists a free group F with a basis X.

Theorem 3.1.7 Every group is a quotient group of a free group.

Theorem 3.1.8 Every subgroup of a free group is free.

Theorem 3.1.9 Free groups on X_1 and X_2 are isomorphic if and only if $|X_1| = |X_2|$.

Theorem 3.1.10 The matrices $A = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ and $B = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$ over $\mathbb{Z}$ are a basis for a free group.

3.2 Finitely Presented Groups

We start with the definition of a presentation of the group.

Definition 3.2.1 A presentation $G = \langle X | \Re \rangle$ is a pair consisting of a set X called generators and a set $\Re$ of words on X called relations.

Let G be a group and $X \subseteq G$. Then the smallest normal subgroup of G containing X is defined as the normal closure of X in G and denote it by N. The group presented by $G = \langle X | \Re \rangle$ is the group F/N where F is the free group with free basis X and N is the normal closure of $\Re$ in F, that is, the smallest normal subgroup containing $\Re$.

Definition 3.2.2 If both X and $\Re$ are finite, then $G = \langle X | \Re \rangle$ is said to be a finite presentation.

Every element g of $G = \langle X | \Re \rangle$ can be described by a word in $X^{\pm}$ and there are many ways to describe an element. In other words, the finitely presented group G consists of equivalence classes of words. The fact that w represents the identity means that repeated application of the equations in $\Re$ with the rules of free cancelation transform w into the empty word 1_X. Applying an equation $u = v$ from $\Re$ means replacing a subword equal to either u or v by the other, and applying the rules of free cancelation to w means either deleting or inserting a subword of the form xx^{-1} or $x^{-1}x$, for $x \in X$, that is, two words w and v are equivalent in G if we can transform w to v by a finite sequence of replacement as follows :

(1) deleting $x_i {x_i}^{-1}$ or ${x_i}^{-1} x_i$
(2) inserting $x_i {x_i}^{-1}$ or ${x_i}^{-1} x_i$
(3) deleting r_j or ${r_j}^{-1}$
(4) inserting r_j or ${r_j}^{-1}$.

Example 1

The group $\langle\ x,\ y \mid x^2,\ y^3,\ (xy)^2\ \rangle$, which is a dihedral group of order 6, can be written as $\langle\ x, y \mid x^2 = 1,\ y^2 = y^{-1},\ xyx = y^{-1}\ \rangle$. □

Example 2

A presentation of the symmetric group S_3 with generators $x = (1, 2)$ and $y = (1, 2, 3)$ is $\langle\ x,\ y \mid x^2 = 1,\ y^2 = y^{-1},\ xyx = y^{-1}\ \rangle$. □

Definition 3.2.3 A normal form which is within an equivalence class specifies a representative element, which is in a simplest form.

If two distinct terms t and v have the same normal form, then $t = v$ is an identity. Every object under consideration must have exactly one normal form, and two objects that have the same normal form must be essentially the same. In general, it is not true that one can get a normal form for the elements, by stepwise cancelation. Usually we would like the normal form for $u \in G$ to be the simplest word defining u. If we can compute normal forms, then we can solve the word problem as two words represent the same element of the group if and only if they have the same normal form. If we find normal forms for group elements with an algorithm which put words in the group generators into normal forms, they enable us to determine the finiteness or infiniteness of the number of elements of the group because we can generally count the number of distinct normal forms.

3.3 Modular Groups

This section is based on [12] and [17].

There is a connection between matrix theory and number theory because matrix groups can be defined over $\mathbb{Z}$ as one of the basic rings of number theory.

Matrix groups play an important role in many different branches of mathematics. In particular, the most important group is the modular group, $\mathrm{SL}_2(\mathbb{Z})$ as it is the most famous example of Fuchsian groups of which the study led to the introduction of combinatorial group theory. So we study general linear groups, special linear groups and modular groups used for the construction of Grigoriev and Ponomarenko homomorphic public-key cryptosystem in Chapter 7.

Definition 3.3.1 Given a ring R with identity, the general linear group $\mathrm{GL}_n(R)$ is the group of $n \times n$ invertible matrices with elements in R.

Definition 3.3.2 Given a ring R with identity, the special linear group $\mathrm{SL}_n(R)$ is the group of $n \times n$ matrices with elements in R and determinant 1.

We denote the special linear group $\mathrm{SL}_n(q)$, where q is a prime power, the set of $n \times n$ matrices with determinant 1 and entries in the finite field $\mathbb{F}_q$. The special linear group $\mathrm{SL}_n(R)$ is a subgroup of the general linear group $\mathrm{GL}_n(R)$.

Definition 3.3.3 The projective special linear group $\mathrm{PSL}_n(q)$ is the group obtained from the special linear group $\mathrm{SL}_n(q)$ on factoring by the scalar matrices contained in that group.

Theorem 3.3.4 The following are equivalent.

(1) $\mathrm{SL}_2(\mathbb{Z})/\pm I$, the quotient of the group $\mathrm{SL}_2(\mathbb{Z})$ of 2×2 integer matrices with determinant 1 modulo its central subgroups $\{\pm I\}$.

(ii) The group of complex fractional linear transformations $z \mapsto \frac{az+b}{cz+d}$ with integer coefficients satisfying $ad - bc = 1$.

Definition 3.3.5 The modular group is the group of all linear fractional transformations of the upper half of the complex plane which have the form

$$z \mapsto \tfrac{az+b}{cz+d}$$

where a, b, c and d are integers with $ad - bc = 1$, and the group operation is composition of functions.

The modular group is the group of all linear fractional transformations with determinant 1 and the modular group is a specific case of the special linear group. Moreover, the modular group is defined as $\mathrm{PSL}_2(\mathbb{Z})$, but instead of the notation $\mathrm{PSL}_2(\mathbb{Z})$, we use the notation $\mathrm{SL}_2(\mathbb{Z})$. We also give another description to define the modular group. The modular group is generated by two transformations S and T

$$S: \ z \mapsto z+1 \quad \text{and} \quad T: \ z \mapsto \tfrac{-1}{z}$$

or

$$S = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad T = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Every transformation $\frac{az+b}{cz+d}$ with $a, b, c, d \in \mathbb{Z}$ and $ad - bc = 1$ can be expressed in the form

$$S^{u_1}\ T\ S^{u_2}\ T\ \cdots\ S^{u_m}\ T.$$

In addition, a presentation of the modular group is

$$\langle\ S,\ T \mid S^2 = I,\ (ST)^3 = I\ \rangle$$

and thus the modular group is isomorphic to the free product of the cyclic groups C_2 and C_3.

Chapter 4

X_n-Representation Algorithm

Let n be a natural number with $n \geq 2$ and

$$A_n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix} \text{ and } B_n = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix}.$$

Then Γ_n denotes a group generated by linear fractional transformations corresponding to two matrices A_n and B_n. Also, X_n denotes a set consisting of two matrices A_n and B_n. We consider the group Γ_n acting on $\mathcal{C} \cup \{\infty\}$ as linear fractional transformations given by

$${A_n}^u(z) = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix}^u (z) = \begin{pmatrix} 1 & nu \\ 0 & 1 \end{pmatrix} (z) = z + nu$$

and

$${B_n}^u(z) = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix}^u (z) = \begin{pmatrix} 1 & 0 \\ nu & 1 \end{pmatrix} (z) = \tfrac{z}{nuz+1}$$

for $z \in \mathcal{C} \cup \{\infty\}$. Since Γ_n is a free group, freely generated by the set X_n by Theorem 3.1.10 and [pp.168, 14], an element M of Γ_n has the unique representation as a reduced word by Theorem 3.1.11 and we call it the X_n-representation of M. There are four types of the X_n-representations and the representation of $M \in \Gamma_n$ takes one of four forms as follows :

$${A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \text{ (odd } m\text{)}$$

$${A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \text{ (even } m\text{)}$$

$$
\begin{gathered}
{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}\ \text{(even } m\text{)}\\
{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}\ \text{(odd } m\text{)}
\end{gathered}
$$

where for each $i = 1, \cdots, m$, u_i is a nonzero integer. For each $M(\neq I) \in \Gamma_n$, M has the unique X_n-representation. The X_n-representation algorithm was introduced by Grigoriev-Ponomarenko [7] and it is one of the algorithms used in the decryption scheme of their homomorphic public-key cryptosystem. The X_n-representation algorithm is used to compute the X_n-representation of $M \in \Gamma_n$. If we input a matrix $M \in \Gamma_n$ to the X_n-representation algorithm, then the X_n-representation algorithm outputs the corresponding reduced word over ${X_n}^{\pm}$.

There are inadequateness in the description and justification of the algorithm given by Grigoriev and Ponomarenko. The aim of the chapter is describe and justify a correct version of the X_n-representation algorithm. The chapter is arranges as follows. In Section 4.1, we prove some properties of two linear fractional transformations ${A_n}^u$ and ${B_n}^u$ with an arbitrary nonzero integer u. In Section 4.2, we analyze the X_n-representation algorithm and correct some parts of the X_n-representation algorithm. In Section 4.3, we modify the X_n-representation algorithm to make it efficient. In Section 4.4, we implement the modified X_n-representation algorithm by programming it with Maple 6. In Section 4.5, we justify the correctness of the modified X_n-representation algorithm.

4.1 Linear Fractional Transformations

In this section, we prove several important properties of linear fractional transformations ${A_n}^u$ and ${B_n}^u$ where u is a nonzero integer and we find some explicit formulae to make the X_n-representation algorithm more efficient. Further, we will utilize these properties to design new X_1-representation algorithms in

Chapter 6 which is used extensively for cryptanalysis of Grigoriev and Ponomarenko homomorphic public-key cryptosystem in Chapter 8. Let D be the unit open disk in the complex plane with center 0, that is, $D = \{z \in \mathcal{C} || z| < 1\}$ and let $D^c = \mathcal{C} - \bar{D} = \{z \in \mathcal{C} || z| > 1\}$ be the complement of the closure of D. At first, we show that ${A_n}^u$ maps D into D^c and ${B_n}^u$ maps D^c into D.

Lemma 4.1.1 Let u be a nonzero integer and $z \in D$. Then ${A_n}^u(z) \in D^c$.

Proof Let $z = a + bi \in D$. Then $-1 < a < 1$ and ${A_n}^u(z) = \begin{pmatrix} 1 & nu \\ 0 & 1 \end{pmatrix}(z) = z + nu = (a + nu) + bi$. Since $n \geq 2$, if $u \geq 1$, then $a + nu > -1 + nu \geq 1$ and so, $a + nu \in D^c$. If $u \leq -1$, then $a + nu < 1 + nu \leq -1$ and so, $a + nu \in D^c$. Therefore, in either case, ${A_n}^u(z) = (a + nu) + bi \in D^c$. □

Lemma 4.1.2 Let u be a nonzero integer and $z \in D^c$. Then ${B_n}^u(z) \in D$.

Proof Let $z = a + bi \in D^c$ and consider ${B_n}^u(z) = \begin{pmatrix} 1 & 0 \\ nu & 1 \end{pmatrix}(z) = \frac{z}{nuz+1} = \frac{1}{nu+\frac{1}{z}}$. As $z \in D^c$, $\frac{1}{z} \in D$ and by Lemma 4.1.1, ${A_n}^u(\frac{1}{z}) = \frac{1}{z} + nu \in D^c$. Hence, ${B_n}^u(z) = \frac{1}{nu+\frac{1}{z}} \in D$. □

The following Theorems immediately are obtained by Lemma 4.1.1 and Lemma 4.1.2.

Theorem 4.1.3 If $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd m, $i = 1, \cdots, m \in \mathbb{N}$ and nonzero $u_i \in \mathbb{Z}$, then for $z \in D$, $M(z) \in D^c$.

Theorem 4.1.4 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with even m, $i = 1, \cdots, m \in \mathbb{N}$ and nonzero $u_i \in \mathbb{Z}$, then for $z \in D$, $M(z) \in D$.

Theorem 4.1.5 If $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ with even m, $i = 1, \cdots, m \in$

$\mathbb{N}$ and nonzero $u_i \in \mathbb{Z}$, then for $z \in D^c$, $M(z) \in D^c$.

Theorem 4.1.6 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ with odd m, $i = 1, \cdots, m \in \mathbb{N}$ and nonzero $u_i \in \mathbb{Z}$, then for $z \in D^c$, $M(z) \in D$.

4.2 Analysis of X_n-Representation Algorithm

In this section, we analyze the X_n-representation algorithm given by Grigoriev and Ponomarenko in [7] and we correct errors which appear in its description. A matrix $M \in \Gamma_n$ is input to the X_n-representation algorithm and the algorithm outputs the X_n-representation of M. (z, z') denotes a pair of complex numbers with $|z| < 1$ and $|z'| > 1$. Grigoriev and Ponomarenko suggest $(z, z') = (\frac{1}{2}, 2)$. Grigoriev and Ponomarenko note that given $z \in D \cup D^c$, there is at most one integer u such that

$$(z \in D^c \wedge {A_n}^u(z) \in D) \vee (z \in D \wedge {B_n}^u(z) \in D^c).$$

If such an integer exists, then put $C = {A_n}^u$ if $z \in D^c$ and $C = {B_n}^u$ if $z \in D$.

X_n-Representation Algorithm

Step 1 $(L, L') \leftarrow (M, M)$ and $(w, w') \leftarrow (1_{X_n}, 1_{X_n})$.

Step 2 $L = I \Rightarrow$ output w. $L' = I \Rightarrow$ output w'.

Step 3 $(w, w') \leftarrow (C^{-1}w, {C'}^{-1}w')$ and $(L, L') \leftarrow (CL, C'L')$ where $C = C(Lz)$ and $C' = C(L'(z'))$. Go to Step 2.

First of all, we correct an error in their description of the X_n-representation algorithm. In Step 3 of the X_n-representation algorithm, their setting is $w = C^{-1}w$ and $w' = {C'}^{-1}w'$, but they should be $w = wC^{-1}$ and $w' = w'{C'}^{-1}$. In order to demonstrate why this correction is needed,

In Step 1 of the first iteration, input

$$M = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix} = \begin{pmatrix} 1+n^2 & n \\ n & 1 \end{pmatrix}$$

to the X_n-representation algorithm where the X_n-representation of M is A_nB_n. In Step 3 of the first iteration, $L(\frac{1}{2}) = M(\frac{1}{2}) = \frac{n(n+2)+1}{n+2}$ and $L'(2) = M(2) = \frac{2n^2+n+2}{2n+1} = \frac{n(2n+1)+2}{2n+1}$. For $n \geq 2$, $L(\frac{1}{2}) = M(\frac{1}{2}) > 1$ and $L(2) = M(2) > 1$. Therefore, set $C = {A_n}^u$ and $C' = {A_n}^u$. Since $M(\frac{1}{2}) > 1$ and $M(2) > 1$, to find the nonzero integer u, let

$$C(M(\tfrac{1}{2})) = {A_n}^u(M(\tfrac{1}{2})) = nu + \tfrac{n(n+2)+1}{n+2} = nu + n + \tfrac{1}{n+2} = n(u+1) + \tfrac{1}{n+2} \in D$$

$$C(M(2)) = {A_n}^u(M(2)) = nu + \tfrac{n(2n+1)+2}{2n+1} = nu + n + \tfrac{2}{2n+1} = n(u+1) + \tfrac{2}{2n+1} \in D.$$

In both cases, we find $u = -1$ and thus, $C = {A_n}^{-1}$ and $C' = {A_n}^{-1}$. The algorithm sets $L = CL = {A_n}^{-1}M = B_n$, $L' = C'L = {A_n}^{-1}M = B_n$. Thus, $w = C^{-1}w = A_n$ and $w' = C'^{-1}w = A_n$.

In Step 1 of the second iteration, input $L = B_n$ and $L' = B_n$ to the X_n-representation algorithm.

In Step 3 of the second iteration, for $n \geq 2$, $L(\frac{1}{2}) = B_n(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}n+1} = \frac{1}{n+2} < 1$ and $L'(2) = B_n(2) = \frac{2}{2n+1} < 1$. Set $C = {B_n}^u$ and $C' = {B_n}^u$. To find the nonzero integer u, let

$$C(L(\tfrac{1}{2})) = {B_n}^u(L(\tfrac{1}{2})) = {B_n}^u(\tfrac{1}{n+2}) = \frac{\frac{1}{n+2}}{\frac{1}{n+2}nu+1} = \frac{1}{nu+n+2} \in D^c.$$

Then $-1 < n(u+1)+2 < 1$ and $-3 < n(u+1) < -1$. So $n(u+1) = -2$. Since $n \geq 2$, we find $u+1 = -1$ and $u = -2$. So $C = {B_n}^{-2}$. Let

$$C'(L(2)) = {B_n}^u(L(2)) = {B_n}^u(\tfrac{2}{2n+1}) = \frac{\frac{2}{2n+1}}{\frac{2}{2n+1}nu+1} = \frac{2}{2nu+2n+1} = \frac{1}{nu+n+\frac{1}{2}} \in D^c.$$

Then $-1 < nu+n+\frac{1}{2} < 1$ and $\frac{-3}{2} < n(u+1) < \frac{1}{2}$. So $n(u+1) = -1$ or $n(u+1) = 0$. Therefore, $C' = {B_n}^{-1}$. Since $n \geq 2$, $u = -1$.

The algorithm sets $L = CL = {B_n}^{-2}B_n = {B_n}^{-1} \neq I$, $L' = C'L' = {B_n}^{-1}B_n = I$. $w = C^{-1}w = {B_n}^2A_n$, $w' = C'^{-1}w' = B_nA_n$. Hence, in the second iteration, for $z = 2$, the X_n-representation algorithm outputs B_nA_n as the X_n-representation of M. Therefore, in Step 3, its description should be corrected to setting

$$w \leftarrow wC^{-1} \text{ and } w' \leftarrow w'C'^{-1}.$$

Next, it should be noted that the X_n-representation algorithm sometimes throws up errors as it operates. As concrete cases, by definition of the linear fractional transformation, ${B_n}^u(z) = \infty$ in case that the denominator of ${B_n}^u(z)$ is zero. For example, ${B_2}^{-1}(\frac{1}{2}) = \infty$. In addition, we may have the case $|{B_n}^u(z)| = 1$, for example, $|{B_3}^{-1}(\frac{1}{2})| = 1$ and we may also have the case ${A_n}^u(z) = 0$, for example, ${A_2}^{-1}(2) = 0$. These concrete examples show that the X_n-representation algorithm does not work for those cases. Hence, we have to consider all cases so that the X_n-representation algorithm works for any case.

Grigoriev and Ponomarenko do not show clearly how to compute the integer u of ${A_n}^u$ and ${B_n}^u$. In order to obtain the exponent u of ${A_n}^u$ and ${B_n}^u$, Grigoriev and Ponomarenko require the determination of the nonzero integer exponent u such that $(z \in D^c \wedge {A_n}^u(z) \in D) \vee (z \in D \wedge {B_n}^u(z) \in D^c)$, but the algorithm does not provide a direct way to compute it. We give explicit formulae to compute the nonzero exponent u of A_n and B_n in the X_n-representation algorithm. This allows our modified X_n-representation algorithm in Section 5.3 to run very efficiently. The following two theorems provide explicit formulae to compute the exponent u of ${A_n}^u$ and ${B_n}^u$.

Theorem 4.2.1 Let $z \in \mathbb{R}$ with $|z| > 1$. If there exists a nonzero integer u such that $|{A_n}^u(z)| < 1$, then $u = \lceil \frac{-1-z}{n} \rceil = \lfloor \frac{1-z}{n} \rfloor$.

Proof Let $z \in D^c \cap \mathbb{R}$. Suppose $|{A_n}^u(z)| = |nu + z| < 1$ for a nonzero $u \in \mathbb{Z}$. Then $-1 < nu + z < 1$, so that $-1 - z < nu < 1 - z$. Hence $\frac{-1-z}{n} < u < \frac{1-z}{n}$. Since for $n \geq 2$, the distance between $\frac{-1-z}{n}$ and $\frac{1-z}{n}$ is

$\frac{2}{n} \leq 1$, there is at most one integer between them. If one of $\frac{-1-z}{n}$ and $\frac{1-z}{n}$ is an integer, then there is no integer between $\frac{-1-z}{n}$ and $\frac{1-z}{n}$ and this is in contradiction with our assumption. Thus neither $\frac{-1-z}{n}$ nor $\frac{1-z}{n}$ is an integer, and $u = \lceil \frac{-1-z}{n} \rceil = \lfloor \frac{1-z}{n} \rfloor$. □

Theorem 4.2.2 Let $z \in \mathbb{R}$ with $|z| < 1$. If there exists a nonzero integer u such that $|{B_n}^u(z)| > 1$, then $u = \lceil \frac{-1}{nz} + \frac{-1}{n} \rceil = \lfloor \frac{-1}{nz} + \frac{1}{n} \rfloor$.

Proof Let $z \in D \cap \mathbb{R}$. Assume $|{B_n}^u(z)| = |\frac{z}{nuz+1}| > 1$ for a nonzero $u \in \mathbb{Z}$. Then $|z| > |nuz + 1|$. If $z > 0$, then $|nuz + 1| < z$, so that $-z < nuz + 1 < z$, $-z - 1 < nuz < z - 1$ and $\frac{-z-1}{nz} < u < \frac{z-1}{nz}$. Hence, $\frac{-1}{nz} + \frac{-1}{n} < u < \frac{-1}{nz} + \frac{1}{n}$. If $z < 0$, then $|nuz+1| < -z$, so that $z - 1 < nuz < -z - 1$. So $\frac{-z-1}{nz} < u < \frac{z-1}{nz}$ and $\frac{-1}{nz} + \frac{-1}{n} < u < \frac{-1}{nz} + \frac{1}{n}$. Because for $n \geq 2$, the distance between $\frac{-1}{nz} + \frac{-1}{n}$ and $\frac{-1}{nz} + \frac{1}{n}$ is $\frac{2}{n} \leq 1$, there exists at most one integer between $\frac{-1}{nz} + \frac{-1}{n}$ and $\frac{-1}{nz} + \frac{1}{n}$. If one of $\frac{-1}{nz} + \frac{-1}{n}$ and $\frac{-1}{nz} + \frac{1}{n}$ is an integer, then there is no integer between $\frac{-1}{nz} + \frac{-1}{n}$ and $\frac{-1}{nz} + \frac{1}{n}$ and it contradicts our assumption. Thus, neither $\frac{-1}{nz} + \frac{-1}{n}$ nor $\frac{-1}{nz} + \frac{1}{n}$ is an integer and $u = \lceil \frac{-1}{nz} + \frac{-1}{n} \rceil = \lfloor \frac{-1}{nz} + \frac{1}{n} \rfloor$. □

In addition, the X_n-representation algorithm has no step to define the functions C and C' though Grigoriev and Ponomarenko mentioned setting C. So we add a step to define the functions C and C' to the X_n-representation algorithm for its modification.

In the next section, we give our modified version of the X_n-representation algorithm. We correct the error of Griogoriev and Ponomarenko's X_n-representation algorithm, adding steps to deal with special cases and we also add steps that provide a direct computation of the exponent u of ${A_n}^u$ and ${B_n}^u$ in a efficient way and definition of the function C.

4.3 Modified X_n-Representation Algorithm

We describe our modified X_n-representation algorithm. The modified X_n-representation algorithm takes a matrix $M \in \Gamma_n$ and a real number $z = \frac{1}{2}$ or $z = 2$ as inputs, and the modified X_n-representation algorithm outputs a word in ${X_n}^{\pm}$. Unlike Grigoriev and Ponomarenko's X_n-representation algorithm operates a pair $(z, z') = (\frac{1}{2}, 2)$ at the same time, the modified X_n-representation algorithm first runs for $z = \frac{1}{2}$. If the modified X_n-representation algorithm outputs the X_n-representation, then the modified X_n-representation algorithm stops and then we do not need to run the X_n-representation algorithm for $z = 2$. If the modified X_n-representation algorithm outputs ϵ for $z = \frac{1}{2}$, then the modified X_n-representation algorithm runs for $z = 2$ to compute the X_n-representation of M. Hence, the modified X_n-representation algorithm outputs either ϵ or the X_n-representation of an input $M \in \Gamma_n$. Note that the algorithm terminates when the algorithm has an output.

Modified X_n-Representation Algorithm

Step 0

$\mathbf{L} \leftarrow \mathbf{M}$

$\mathbf{w} \leftarrow \mathbf{1_{X_n}}.$

Step 1

(1) $\mathbf{L(z) = 0, |L(z)| = 1, L(z) = \infty \Rightarrow}$ **output** ϵ.

(2) $\mathbf{|L(z)| > 1 \Rightarrow v \leftarrow \lfloor \frac{1-L(z)}{n} \rfloor}$ **and** $\mathbf{C \leftarrow {A_n}^v}$.

(3) $\mathbf{|L(z)| < 1 \Rightarrow v \leftarrow \lfloor \frac{-1}{nL(z)} + \frac{1}{n} \rfloor}$ **and** $\mathbf{C \leftarrow {B_n}^v}$.

Step 2

$\mathbf{C = I \Rightarrow}$ **output** ϵ.

Otherwise, L $\leftarrow$ CL and w $\leftarrow$ wC^{-1}

Step 3

L = I $\Rightarrow$ output w.

Otherwise, return Step 1.

4.4 Programming Implementation

In this section, we implement the modified X_n-representation algorithm by programming it to show how the modified X_n-representation algorithm works correctly. We use Maple version 6 to make a program source code. It turns out that the modified X_n-representation algorithm works very efficiently in practice to compute the X_n-representation of $M \in \Gamma_n$.

The following X_n-representation program is equivalent to the modified X_n-representation algorithm because the X_n-representation program source code includes all steps of the modified X_n-representation algorithm. Only the difference between the algorithm and the program depends on the skill to make a program and the programming language. The modified X_n-representation program first runs for $z = \frac{1}{2}$ and next, it runs for $z = 2$ according as the X_n-representation types. The X_n-representation program outputs the X_n-representation of M for either $z = \frac{1}{2}$ or $z = 2$. Now we show the X_n-representation program source code in the following.

X_n-Representation Program Source Code

```
with(GaussInt):
with(linalg):
su:=proc(z::float, n::integer, M11::integer, M12::integer, M21::integer, M22::integer)
local M, v, C, L;
z;
M:=matrix(2,2,[M11, M12, M21, M22]);
```

```
L(z) :=(M11*z+M12)/(M21*z+M22);
if abs(L(z)) = 1 then
print(epsilon);
fi;
if abs(L(z))>1 then
v:=floor((1-L(z))/n);
C:=matrix(2, 2, [1, n * v, 0, 1]) ;
print(matrix(2, 2, [1, n, 0, 1])^{-v});
fi;
if abs(L(z))<1 then
v:=floor((-1)/(n * L(z)) + 1/n);
C:=matrix(2, 2, [1, 0, n*v, 1]);
print (matrix(2, 2, [1, 0, n, 1])^{-v}) ;
fi;
L:=multiply(C, M);
print(L);
end proc:
```

Note that this program source code implements a single pass through the loop of our algorithm. In order to compute the X_n-representation of $M \in \Gamma_n$, we input either $z = 0.5$ instead of $z = \frac{1}{2}$ or $z = 2.0$ instead of $z = 2$, the natural number n, the entities $M11, M12, M21, M22$ of the matrix M to the X_n-representation program. For every execution of the X_n-representation program, the program outputs two matrices. The first matrix means $C^{-1} = {A_n}^{-v}$ or $C^{-1} = {B_n}^{-v}$ in Step 2 of the modified X_n-representation algorithm and the second matrix means $L = CL$ in Step 2 of the modified X_n-representation algorithm. If the identity matrix turns up in the second matrix, then execution of the program terminates. Next collect the first matrix of every execution sequentially and concatenate the first matrices which are collected from all executions in order and then we can obtain a word in ${X_n}^{\pm}$ as the X_n-representation of an input M. When infinite entries appears in the second matrix, then execution of the program terminates. It means that the X_n-representation algorithm outputs ϵ and the algorithm terminates. From now, we show concrete examples to show how the modified X_n-representation algorithm works correctly by implementing the X_n-representation program.

Example 1

For $M = A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \in \Gamma_2$, we check whether the X_n-representation program works correctly. Input the six values $z = 0.5$, $n = 2$, $M11 = 1$, $M12 = 2$, $M21 = 0$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

$> \mathrm{su}(0.5, 2, 1, 2, 0, 1);$

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}^{1}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm and so execution of the program terminates. Collect the first matrix

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}^{1}$$

and this is the X_2-representation of M.

Input $z = 2.0$, $n = 2$, $M11 = 1$, $M12 = 2$, $M21 = 0$ and $M22 = 1$ to the X_n-representation program.

For $z = 2$,

```
> su(2.0, 2, 1, 2, 0, 1) ;
```

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0, 2, 1, -2, 0, 1) ;
```

$$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}^\infty$$

$$\begin{pmatrix} 1 & -2 \\ -\infty & \infty \end{pmatrix}$$

The second matrix of the second execution of the program is an unusual matrix $\begin{pmatrix} 1 & -2 \\ -\infty & \infty \end{pmatrix}$ and thus execution of the program terminates. This case is the same as the situation that the modified X_n-representation algorithm outputs ϵ when $L(2) = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} (2) = 0$ and the algorithm terminates. Hence the X_n-representation program does not output the X_2-representation of M for $z = 2$. $\square$

Example 2

We show implementation for another even number $n = 4$ as an input. Given $M = A_4 = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix} \in \Gamma_4$, input $z = 0.5$, $n = 4$, $M11 = 1, M12 = 4$, $M21 = 0$, $M22 = 1$ to the X_n-representation program.

```
> su(0.5, 4, 1, 4, 0, 1);
```

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{1}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. Thus execution of the program terminates and collect the first matrix

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{1}.$$

and this is the X_4-representation of M obtained from the program for $z = \frac{1}{2}$. Hence the program works correctly to compute the X_4-representation of M.

For $z = 2$,

```
> su(2.0, 4, 1, 4, 0, 1);
```

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{2}$$

$$\begin{pmatrix} 1 & -4 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0, 4, 1, -4, 0, 1) ;
```

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{0}$$

$$\begin{pmatrix} 1 & -4 \\ 0 & 1 \end{pmatrix}$$

The first matrix of the second execution of the program is $\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{0} = I$ which is $C^{-1} = {A_n}^{-v} = I$ and so $C = I$ in Step 2 of the modified X_n-representation algorithm and so execution of the program terminates. This case is the same as the X_n-representation algorithm outputs ϵ and the algorithm terminates.

So the program does not output the X_4-representation of M for $z = 2$. □

Example 3

So far we have seen how the program works for two even numbers $n = 2$ and $n = 4$. Now we see how the program works for odd natural numbers. Given $M = A_3 = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} \in \Gamma_3$, input $z = 0.5$, $n = 3$, $M11 = 1$, $M12 = 3$, $M21 = 0$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

```
> su(0.5, 3, 1, 3, 0, 1) ;
```

$$\begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}^1$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and collect the first matrix

$$\begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}^1$$

and this is the X_3-representation of M.

Input $z = 2.0$, $n = 3$, $M11 = 1$, $M12 = 3$, $M21 = 0$ and $M22 = 1$ to the X_n-representation program.

For $z = 2$,

```
> su(2.0, 3, 1, 3, 0, 1) ;
```

$$\begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & -3 \\ 0 & 1 \end{pmatrix}$$

> su(2.0, 3, 1, -3, 0, 1) ;

$$\epsilon$$

Since $\begin{pmatrix} 1 & -3 \\ 0 & 1 \end{pmatrix}(2.0) = -1$, this means that $|L(2)| = 1$ in Step 1 of the modified X_n-representation algorithm. Both the program and the algorithm output the same ϵ and so both of them terminate. Hence the X_n-representation program does not output the X_3-representation of M for $z = 2$. □

Example 4

We implement the modified X_n-representation algorithm for another odd natural number to show how the algorithm works. Given $M = A_5 = \begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix} \in \Gamma_5$, input $z = 0.5$, $n = 5$, $M11 = 1$, $M12 = 5$, $M21 = 0$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

>su(0.5, 5, 1, 5, 0, 1) ;

$$\begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix}^1$$

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

The second matrix of the first execution of the program is the identity matrix which is that $L = CL = \begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix}^{-1}\begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_5-representation algorithm. So execution of the program terminates and collect the first matrix

$$\begin{pmatrix}1 & 5\\0 & 1\end{pmatrix}^{1}.$$

This is the X_n-representation of M.

Input $z = 2.0$, $n = 5$, $M11 = 1$, $M12 = 5$, $M21 = 0$ and $M22 = 1$ to the X_n-representation program.

For $z = 2$,

```
> su(2.0, 5, 1, 5, 0, 1) ;
```

$$\begin{pmatrix}1 & 5\\0 & 1\end{pmatrix}^{2}$$

$$\begin{pmatrix}1 & -5\\0 & 1\end{pmatrix}$$

```
> su(2.0, 5, 1, -5, 0, 1) ;
```

$$\begin{pmatrix}1 & 5\\0 & 1\end{pmatrix}^{0}$$

$$\begin{pmatrix}1 & -5\\0 & 1\end{pmatrix}$$

The first matrix of the second execution of the program is the identity matrix which is $C^{-1} = {A_n}^{-v} = \begin{pmatrix}1 & 5\\0 & 1\end{pmatrix}^{0} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and the modified X_n-representation algorithm outputs ϵ in step 2 of the algorithm and the algorithm terminates. □

Example 5

Given $M = B_2 = \begin{pmatrix}1 & 0\\2 & 1\end{pmatrix} \in \Gamma_2$, input $z = 0.5$, $n = 2$, $M11 = 1$, $M12 = 0$, $M21 = 2$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

```
> su(0.5, 2, 1, 0, 2, 1) ;
```

$$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$$

```
> su(0.5, 2, 1, 0, -2, 1) ;
```

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}^\infty$$

$$\begin{pmatrix} \infty & -\infty \\ -2 & 1 \end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}^\infty$ which is the same situation that the modified X_n-representation algorithm outputs ϵ in Step 1 of the modified X_n-representation algorithm because $L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} (\frac{1}{2}) = \infty$ in Step 1 of the modified X_n-representation algorithm. In the case, the algorithm and the program terminate. Hence we can not obtain the X_2-representation of M for $z = \frac{1}{2}$.

For $z = 2$,

```
> su(2.0, 2, 1, 0, 2, 1);
```

$$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}^1$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. Collect the first matrix

$\begin{pmatrix}1 & 0\\ 2 & 1\end{pmatrix}^{1}$ and it is the X_2-representation of M. $\square$

Example 6

$M = \begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix} \in \Gamma_4$ and the X_4-representation of M is B_4. Input $z = 0.5$, $n = 4$, $M11 = 1$, $M12 = 0$, $M21 = 4$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

```
> su(0.5, 4, 1, 0, 4, 1) ;
```

$$\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{2}$$

$$\begin{pmatrix}1 & 0\\ -4 & 1\end{pmatrix}$$

```
> su(0.5, 4, 1, 0, -4, 1) ;
```

$$\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{0}$$

$$\begin{pmatrix}1 & 0\\ -4 & 1\end{pmatrix}$$

The first matrix of the second execution of the program is the identity matrix $\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{0} = I$ which is $C^{-1} = {B_n}^{-v} = I$ and so $C = I$ in Step 2 of the modified X_n-representation algorithm. Thus execution of the program terminates and this case is the same as the situation that the modified X_n-representation algorithm outputs ϵ in Step 2 of the modified X_n-representation algorithm and the algorithm terminates. Therefore the X_n-representation program does not output the X_4-representation of M for $z = \frac{1}{2}$.

Input $z = 2$, $n = 4$, $M11 = 1$, $M12 = 0$, $M21 = 4$ and $M22 = 1$ to the

X_n-representation program.

For $z = 2$,

```
> su(2.0, 4, 1, 0, 4, 1) ;
```

$$\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{1}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{-1}\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix} = I$. So execution of the program terminates. Collect the first matrix

$$\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{1}$$

and this is the X_4-representation of M. $\square$

Example 7

Given $M = B_3 = \begin{pmatrix}1 & 0\\ 3 & 1\end{pmatrix} \in \Gamma_3$, input $z = 0.5$, $n = 3$, $M11 = 1$, $M12 = 0$, $M21 = 3$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

```
> su(0.5, 3, 1, 0, 3, 1) ;
```

$$\begin{pmatrix}1 & 0\\ 3 & 1\end{pmatrix}^{2}$$

$$\begin{pmatrix}1 & 0\\ -3 & 1\end{pmatrix}$$

```
> su(0.5, 3, 1, 0, -3, 1) ;
```

$$\epsilon$$

The program outputs ϵ in the second execution of the program and so execution of the program terminates. Since $|L(\frac{1}{2})| = |\begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix}(\frac{1}{2})| = 1$, both the progran and the algorithm output the same ϵ. Hence the X_n-representation program does not output the X_3-representation of an input M for $z = \frac{1}{2}$.

Input $z = 2.0$, $n = 3$, $M11 = 1$, $M12 = 0$, $M21 = 3$ and $M22 = 1$ to the X_n-representation program.

For $z = 2$,

```
> su(2.0, 3, 1, 0, 3, 1) ;
```

$$\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}^1$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates. Collect the first matrix

$$\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}^1$$

and this is the X_n-representation of M. □

Example 8

Given $M = B_5 = \begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix} \in \Gamma_5$, input $z = 0.5$, $n = 5$, $M11 = 1$, $M12 = 0$, $M21 = 5$ and $M22 = 1$ to the X_n-representation program.

For $z = \frac{1}{2}$,

```
> su(0.5, 5, 1, 0, 5, 1) ;
```

$$\begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & 0 \\ -5 & 1 \end{pmatrix}$$

```
> su(0.5, 5, 1, 0, -5, 1);
```

$$\begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix}^0$$

$$\begin{pmatrix} 1 & 0 \\ -5 & 1 \end{pmatrix}$$

Since the first matrix of the second execution of the program is the identity matrix $\begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix}^0 = I$ which is $C^{-1} = {B_n}^{-v} = I$ and so, $C = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and in this case, the modified X_n-representation algorithm outputs ϵ in Step 2 of the algorithm. Hence the X_n-representation program does not output the X_5-representation of M for $z = \frac{1}{2}$.

Input $z = 2.0$, $n = 5$, $M11 = 1$, $M12 = 0$, $M21 = 5$ and $M22 = 1$ to the X_n-representation program.

For $z = 2$,

```
> su(2.0, 5, 1, 0, 5, 1) ;
```

$$\begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix}^1$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and collect the first matrix

$$\begin{pmatrix} 1 & 0 \\ 5 & 1 \end{pmatrix}^{1}.$$

This is the X_5-representation of M. $\square$

Example 9

Given $M = {A_4}^{-2}{B_4}^{3}{A_4}^{2} = \begin{pmatrix} -95 & -768 \\ 12 & 97 \end{pmatrix} \in \Gamma_4$, we check whether the program outputs the correct X_4-representation of M. Input the six values $z = 0.5$, $n = 4$, $M11 = -95$, $M12 = -768$, $M21 = 12$ and $M22 = 97$ to the X_n-representation program.

For $z = \frac{1}{2}$,

$> \mathrm{su}(0.5, 4, -95, -768, 12, 97);$

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2}$$

$$\begin{pmatrix} 1 & 8 \\ 12 & 97 \end{pmatrix}$$

$> \mathrm{su}(0.5, 4, 1, 8, 12, 97);$

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{3}$$

$$\begin{pmatrix} 1 & 8 \\ 0 & 1 \end{pmatrix}$$

$> \mathrm{su}(0.5, 4, 1, 8, 0, 1);$

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^2$$

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

The second matrix of the third execution of the program is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 & 8 \\ 0 & 1 \end{pmatrix} = I$ and so execution of the program terminates. Collect the first matrix of every execution of the program. Then we have

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^3 \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^2$$

as the X_4-representation of M.

For $z = 2$,

$> \mathrm{su}(2.0, 4, -95, -768, 12, 97);$

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2}$$

$$\begin{pmatrix} 1 & 8 \\ 12 & 97 \end{pmatrix}$$

$> \mathrm{su}(2.0, 4, 1, 8, 12, 97);$

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^3$$

$$\begin{pmatrix} 1 & 8 \\ 0 & 1 \end{pmatrix}$$

$> \mathrm{su}(2.0, 4, 1, 8, 0, 1);$

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^3$$

$$\begin{pmatrix} 1 & -4 \\ 0 & 1 \end{pmatrix}$$

> su(2.0, 4, 1, −4, 0, 1);

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^0$$

$$\begin{pmatrix} 1 & -4 \\ 0 & 1 \end{pmatrix}$$

The first matrix of the fourth execution of the program is the identity matrix which is $C^{-1} = {A_n}^{-v} = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^0 = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and in this case, the modified X_n-representation algorithm outputs ϵ in Step 2 of the algorithm. Hence the X_n-representation program does not output the X_4-representation of M for $z = 2$. $\Box$

Example 10

Given $M = {B_4}^{-1}{A_4}^{-2}{B_4}^3{A_4}^2 = \begin{pmatrix} -95 & -768 \\ 392 & 3169 \end{pmatrix} \in \Gamma_4$, input $z = 0.5$, $n = 4$, $M11 = -95$, $M12 = -768$, $M21 = 392$ and $M22 = 3169$ to the X_n-representation program.

For $z = \frac{1}{2}$,

> su(0.5, 4, −95, −768, 392, 3169);

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{-1}$$

$$\begin{pmatrix} -95 & -768 \\ 12 & 97 \end{pmatrix}$$

$>$ su$(0.5, 4, -95, -768, 12, 97)$;

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2}$$

$$\begin{pmatrix} 1 & 8 \\ 12 & 97 \end{pmatrix}$$

$>$ su$(0.5, 4, 1, 8, 12, 97)$;

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{3}$$

$$\begin{pmatrix} 1 & 8 \\ 0 & 1 \end{pmatrix}$$

$>$ su$(0.5, 4, 1, 8, 0, 1)$;

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{2}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = CL = {A_n}^v L = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 & 8 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates. Collect the first matrix of every execution and then we have

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{3} \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{2}.$$

as the X_4-representation of M.

Input $z = 2.0$, $n = 4$, $M11 = -95$, $M12 = -768$, $M21 = 392$ and $M22 = 3169$ to the X_n-representation program.

For $z = 2$.

$>$ su$(2.0, 4, -95, -768, 392, 3169)$;

$$\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{-1}$$

$$\begin{pmatrix}-95 & -768\\ 12 & 97\end{pmatrix}$$

$>$ su(2.0, 4, −95, −768, 12, 97);

$$\begin{pmatrix}1 & 4\\ 0 & 1\end{pmatrix}^{-2}$$

$$\begin{pmatrix}1 & 8\\ 12 & 97\end{pmatrix}$$

$>$ su(2.0, 4, 1, 8, 12, 97);

$$\begin{pmatrix}1 & 0\\ 4 & 1\end{pmatrix}^{3}$$

$$\begin{pmatrix}1 & 8\\ 0 & 1\end{pmatrix}$$

$>$ su(2.0, 4, 1, 8, 0, 1);

$$\begin{pmatrix}1 & 4\\ 0 & 1\end{pmatrix}^{3}$$

$$\begin{pmatrix}1 & -4\\ 0 & 1\end{pmatrix}$$

$>$ su(2.0, 4, 1, −4, 0, 1);

$$\begin{pmatrix}1 & 4\\ 0 & 1\end{pmatrix}^{0}$$

$$\begin{pmatrix}1 & -4\\ 0 & 1\end{pmatrix}$$

The first matrix of the fifth execution of the program is the identity matrix which is $C^{-1} = {A_n}^{-v} = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{0} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates. In this case, the modified X_n-representation algorithm outputs ϵ in Step 2 of the algorithm and the algorithm terminates. Hence the X_n-representation program does not output the X_4-representation of M for $z = 2$. □

Example 11

Given $M = {A_4}^{-2}{B_4}^{3}{A_4}^{2}B_4 = \begin{pmatrix} -3167 & -768 \\ 400 & 97 \end{pmatrix} \in \Gamma_4$, input the six values $z = 0.5$, $n = 4$, $M11 = -3167$, $M12 = -768$, $M21 = 400$ and $M22 = 97$ to the X_n-representation program.

For $z = \frac{1}{2}$

> su(0.5, 4, −3167, −768, 400, 97);

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2}$$

$$\begin{pmatrix} 33 & 8 \\ 400 & 97 \end{pmatrix}$$

> su(0.5, 4, 33, 8, 400, 97);

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{3}$$

$$\begin{pmatrix} 33 & 8 \\ 4 & 1 \end{pmatrix}$$

> su(0.5, 4, 33, 8, 4, 1);

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{2}$$

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}$$

$>$ su$(0.5, 4, 1, 0, 4, 1)$;

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix}$$

$>$ su(0.5,4,1,0,-4,1);

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^0$$

$$\begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix}$$

The first matrix of the fifth execution of the program is the identity matrix which is $C^{-1} = {B_n}^{-v} = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^0 = I$. So execution of the program terminates and in this case, the modified X_n-representation algorithm outputs ϵ in Step 2 of the algorithm. Therefore the program does not output the X_4-representation of M for $z = \frac{1}{2}$.

For $z = 2$,

$>$ su$(2.0, 4, -3167, -768, 400, 97)$;

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2}$$

$$\begin{pmatrix} 33 & 8 \\ 400 & 97 \end{pmatrix}$$

$>$ su$(2.0, 4, 33, 8, 400, 97)$;

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^3$$

$$\begin{pmatrix} 33 & 8 \\ 4 & 1 \end{pmatrix}$$

> su(2.0, 4, 33, 8, 4, 1);

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^2$$

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}$$

> su(2.0, 4, 1, 0, 4, 1);

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^1$$

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

The second matrix of the fourth execution is the identity matrix which is $L = CL = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and collect each first matrix of every execution of the program. Then we have

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^3 \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^2 \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^1 .$$

This is the X_4-representation of M. □

Example 12

$M = {B_4}^2 {A_4}^{-2} {B_4}^3 {A_4}^2 {B_4}^{-1} = \begin{pmatrix} 2977 & -768 \\ 23440 & -6047 \end{pmatrix} \in \Gamma_4$, input the six values $z = 0.5$, $n = 4$, $M11 = 2977$, $M12 = -768$ and $M21 = 23440$ and $M22 = -6047$ to the X_n-representation program.

For $z = \frac{1}{2}$,

```
> su(0.5, 4, 2977, -768, 23440, -6047);
```

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{2}$$

$$\begin{pmatrix} 2977 & -768 \\ -376 & 97 \end{pmatrix}$$

```
> su(0.5, 4, 2977, -768, -376, 97);
```

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{-2}$$

$$\begin{pmatrix} -31 & 8 \\ -376 & 97 \end{pmatrix}$$

```
> su(0.5, 4, -31, 8, -376, 97);
```

$$\begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}^{3}$$

$$\begin{pmatrix} -31 & 8 \\ -4 & 1 \end{pmatrix}$$

```
> su(0.5, 4, -31, 8, -4, 1);
```

$$\begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}^{2}$$

$$\begin{pmatrix} 1 & 0 \\ -4 & 1 \end{pmatrix}$$

```
> su(0.5, 4, 1, 0, -4, 1);
```

$$\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^0$$

$$\begin{pmatrix}1 & 0\\-4 & 1\end{pmatrix}$$

The first matrix of the fifth execution of the program is the identity matrix which is $C^{-1} = B_n{}^{-v} = \begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^0 = I$ and so execution of the program terminates. This is the same as the modified X_n-representation algorithm outputs ϵ in Step 2 of the algorithm and the algorithm terminates. Thus the X_n-representation program does not output the X_4-representation of M for $z = \frac{1}{2}$.

For $z = 2$,

$> \mathrm{su}(2.0, 4, 2977, -768, 23440, -6047);$

$$\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^2$$

$$\begin{pmatrix}2977 & -768\\-376 & 97\end{pmatrix}$$

$> \mathrm{su}(2.0, 4, 2977, -768, -376, 97);$

$$\begin{pmatrix}1 & 4\\0 & 1\end{pmatrix}^{-2}$$

$$\begin{pmatrix}-31 & 8\\-376 & 97\end{pmatrix}$$

$> \mathrm{su}(2.0, 4, -31, 8, -376, 97);$

$$\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^3$$

$$\begin{pmatrix}-31 & 8\\-4 & 1\end{pmatrix}$$

$> \mathrm{su}(2.0, 4, -31, 8, -4, 1);$

$$\begin{pmatrix}1 & 4\\0 & 1\end{pmatrix}^2$$

$$\begin{pmatrix}1 & 0\\-4 & 1\end{pmatrix}$$

$> \mathrm{su}(2.0, 4, 1, 0, -4, 1);$

$$\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^{-1}$$

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

The second matrix of the fifth execution of the program is the identity matrix which is $L = CL = {B_n}^v {B_n}^{-1} = \begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^{-1}\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix} = I$ in Step 2 of the modified X_n-representation algorithm. So execution of the program terminates and collect each first matrix of every execution of the program. Then we have

$$\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^2\begin{pmatrix}1 & 4\\0 & 1\end{pmatrix}^{-2}\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^3\begin{pmatrix}1 & 4\\0 & 1\end{pmatrix}^2\begin{pmatrix}1 & 0\\4 & 1\end{pmatrix}^{-1}$$

as the X_4-representation of M. □

4.5 Correctness of Modified X_n-Representation Algorithm

Let $n \geq 2$ and $M \in \Gamma_n$. Assume that all exponents of the X_n-representation of $M \in \Gamma_n$ are nonzero integers. We prove the correctness of the modified X_n-representation algorithm. We consider all four types of the X_n-representations and for each X_n-representation type, we show how the modified X_n-representation algorithm works for $z = \frac{1}{2}$ and $z = 2$, respectively. For convenience, the modified X_n-representation algorithm is called the X_n-representation algorithm.

Theorem 4.5.1 Let $M = {A_n}^u \in \Gamma_n$ with a nonzero integer u. If M is input to the X_n-representation algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${A_n}^u$ as the X_n-representation of M.

Proof If $M = {A_n}^u$, then in Step 1 of the first iteration, by Lemma 4.1.1, $|L(\frac{1}{2})| = |{A_n}^u(\frac{1}{2})| = |nu + \frac{1}{2}| > 1$. So $v = \lfloor \frac{1-L(\frac{1}{2})}{n} \rfloor = \lfloor \frac{1-nu-\frac{1}{2}}{n} \rfloor = -u$, $C = {A_n}^v = {A_n}^{-u}$. In Step 2, $L = CL = {A_n}^{-u}{A_n}^u = I$ and $w = wC^{-1} = {A_n}^u$. In Step 3, since $L = I$, the algorithm outputs ${A_n}^u$ as the X_n-representation of M and then the algorithm terminates. □

Theorem 4.5.2 Let $M = {A_n}^u \in \Gamma_n$ with a nonzero integer u. If M is input to the X_n-representation algorithm ($z = 2$), then the algorithm outputs ϵ.

Proof If $M = {A_n}^u$, then in Step 1 of the first iteration, $L(2) = {A_n}^u(2) = nu + 2$.

If $n = 2$ and $u = -1$, then $|L(2)| = |{A_n}^u(2)| = |nu+2| = 0$ and the algorithm outputs ϵ. Then the algorithm terminates.

If $n = 3$ and $u = -1$, then $|L(2)| = |{A_n}^u(2)| = |nu+2| = 1$ and the algorithm outputs ϵ. Then the algorithm terminates.

Except for these two cases, in Step 1 of the first iteration, $|L(2)| = |{A_n}^u(2)| = |nu + 2| > 1$, so $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu-2}{n} \rfloor = -u - 1$ and $C = {A_n}^v = {A_n}^{-u-1}$. In Step 2, $L = CL = {A_n}^{-u-1}{A_n}^u = {A_n}^{-1}$ and $w = wC^{-1} = {A_n}^{u+1}$. In Step 3, since $L = {A_n}^{-1} \neq I$, return Step 1.

In Step 1 of the second iteration, $L(2) = {A_n}^{-1}(2) = -n + 2$.

If $n = 2$, then $|L(2)| = |-n+2| = 0$ and the algorithm outputs ϵ. Then the algorithm terminates.

If $n = 3$, then $|L(2)| = |-n+2| = 1$ and the algorithm outputs ϵ. Then the algorithm terminates.
If $n \geq 4$, then $|L(2)| = |{A_n}^{-1}(2)| = |-n+2| > 1$, so that $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1+n-2}{n} \rfloor = \lfloor \frac{n-1}{n} \rfloor = \lfloor 1 - \frac{1}{n} \rfloor = 0$ and $C = {A_n}^v = I$. In Step 2, since $C = I$, the algorithm outputs ϵ and then the algorithm terminates. □

Theorem 4.5.3 Let $M = {B_n}^u \in \Gamma_n$ with a nonzero integer u. If M is input to the X_n-representation algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ.

Proof In Step 1 of the first iteration, $L(\frac{1}{2}) = {B_n}^u(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu+1} = \frac{1}{nu+2}$.
If $n = 2$ and $u = -1$, then $L(\frac{1}{2}) = {B_n}^u(\frac{1}{2}) = \infty$ and the algorithm outputs ϵ. Then the algorithm terminates.
If $n = 3$ and $u = -1$, then $|L(\frac{1}{2})| = |{B_n}^u(\frac{1}{2})| = |\frac{1}{nu+2}| = 1$ and the algorithm outputs ϵ. Then the algorithm terminates.

Except for these two cases, in Step 1 of the first iteration, $|L(\frac{1}{2})| = |{B_n}^u(\frac{1}{2})| = |\frac{1}{nu+2}| < 1$, so $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu+2}} + \frac{1}{n} \rfloor = -u-1$ and $C = {B_n}^v = {B_n}^{-u-1}$. In Step 2, $L = CL = {B_n}^{-u-1}{B_n}^u = {B_n}^{-1}$ and $w = wC^{-1} = {B_n}^{u+1}$. In Step 3, as $L = {B_n}^{-1} \neq I$, return Step 1.
In Step 1 of the second iteration, $L(\frac{1}{2}) = {B_n}^{-1}(\frac{1}{2}) = \frac{1}{-n+2}$.
If $n = 2$, then $L(\frac{1}{2}) = {B_n}^{-1}(\frac{1}{2}) = \frac{1}{-n+2} = \infty$ and the algorithm outputs ϵ. Then the algorithm terminates.
If $n = 3$, then $|L(\frac{1}{2})| = |{B_n}^{-1}(\frac{1}{2})| = |\frac{1}{-n+2}| = 1$ and the algorithm outputs ϵ. Then the algorithm terminates.
If $n \geq 4$, then $|L(\frac{1}{2})| = |{B_n}^{-1}(\frac{1}{2})| = |\frac{1}{-n+2}| < 1$, so $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{-n+2}} + \frac{1}{n} \rfloor = \lfloor \frac{n-2}{n} + \frac{1}{n} \rfloor = \lfloor 1 - \frac{1}{n} \rfloor = 0$ and $C = {B_n}^v = I$. In Step 2, since $C = I$, the algorithm outputs ϵ and then the algorithm terminates. □

Theorem 4.5.4 Let $M = {B_n}^u \in \Gamma_n$ with a nonzero integer u. If M is input to the X_n-representation algorithm ($z = 2$), then the algorithm outputs ${B_n}^u$ as the X_n-representation of M.

Proof In Step 1 of the first iteration, by Lemma 4.1.2, $|L(2)| = |{B_n}^u(2)| = |\frac{2}{2nu+1}| = |\frac{1}{nu+\frac{1}{2}}| < 1$, so $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu+\frac{1}{2}}} + \frac{1}{n} \rfloor = \lfloor -u + \frac{1}{2n} \rfloor = -u$ and $C = {B_n}^v = {B_n}^{-u}$. In Step 2, $L = CL = {B_n}^{-u}{B_n}^u = I$ and $w = wC^{-1} = {B_n}^u$. In Step 3, since $L = I$, the algorithm outputs ${B_n}^u$ as the X_n-representation of M and then the algorithm terminates. $\square$

Theorem 4.5.5 Let a matrix $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ where odd $m \geq 3$ and u_i is a nonzero integer ($i = 1, \cdots, m$). If M is input to the X_n-representation algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ as the X_n-representation of M.

Proof Let a matrix $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ with odd m. Then in Step 1 of the first iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}(\beta_1)| = |nu_1 + \beta_1| > 1$ where $\beta_1 = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|\beta_1| < 1$ and $0 < \frac{1-\beta_1}{n} < \frac{2}{n} \leq 1$. So $v = \lfloor \frac{1-L(\frac{1}{2})}{n} \rfloor = \lfloor \frac{1-nu_1-\beta_1}{n} \rfloor = \lfloor -u_1 + \frac{1-\beta_1}{n} \rfloor = -u_1$ and $C = {A_n}^v = {A_n}^{-u_1}$. In Step 2, $L = CL = {A_n}^{-u_1}{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}$. In Step 3, since $L = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$, return Step 1.

Assume that for $1 < j < m$, $L = {A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd j and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with even j and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with even j,

$L(\frac{1}{2}) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j n u_j+1} = \frac{1}{nu_j+\frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_j}(\alpha_j)| = |\frac{1}{nu_j+\frac{1}{\alpha_j}}| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(\frac{1}{2})| > 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{2}{n} \leq 1$. So $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = -u_j$ and $C = {B_n}^{v} = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j} {B_n}^{u_j} {A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} = {A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1} {B_n}^{u_2} \cdots {A_n}^{u_{j-1}} {B_n}^{u_j}$. In Step 3, since $L = {A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ $\neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ with odd j, then by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}(\beta_j)| = |nu_j + \beta_j| > 1$ where $\beta_j = {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|\beta_j| < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$. So $v = \lfloor \frac{1-L(\frac{1}{2})}{n} \rfloor = \lfloor \frac{1-nu_j-\beta_j}{n} \rfloor = \lfloor -u_j + \frac{1-\beta_j}{n} \rceil = -u_j$ and $C = {A_n}^{v} = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j} {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ $= {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{j-1}} {A_n}^{u_j}$. In Step 3, since $L = {B_n}^{u_{j+1}} {A_n}^{u_{j+2}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} \neq I$, return Step 1.

If $j = m$, then in Step 1 of the $j = m$th iteration, $L = {A_n}^{u_m}$ and by Theorem 4.5.1, in Step 2, $L = CL = {A_n}^{-u_m} {A_n}^{u_m} = I$ and $w = wC^{-1} = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$. In Step 3, since $L = I$, the algorithm outputs ${A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ as the X_n-representation of M and then the algorithm terminates. □

Theorem 4.5.6 Let a matrix $M = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} \in \Gamma_n$ where odd $m \geq 3$ and u_i is a nonzero integer $(i = 1, \cdots, m)$. If M is input to the X_n-representation algorithm $(z = 2)$, then the algorithm outputs ϵ.

Proof Let a matrix $M = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} \in \Gamma_n$ with odd m. Then in Step 1 of the first iteration, $L(2) = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2) = {A_n}^{u_1}(\beta_1) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$, ${B_n}^{u_{m-1}}(0) = 0$ and ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$. By Theorem 4.1.3, $L(2) = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ $(2) = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-2}}(0) \in D^c$, so $|L(2)| > 1$. By Theorem 4.1.4, $\beta_1 = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_2} \cdots {A_n}^{u_{m-2}}(0) \in D$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$, ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$. Put $\gamma = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1)$ and then $|\gamma| < 1$. By Theorem 4.1.3, $L(2) = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-2}}(\gamma)$ $\in D^c$ and so $|L(2)| > 1$. By Theorem 4.1.4, $\beta_1 = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_2} \cdots {A_n}^{u_{m-2}}(\gamma) \in D$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| > 1$. By Theorem 4.1.6, $\beta_1 = {B_n}^{u_2}{A_n}^{u_3} \cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_2}{A_n}^{u_3} \cdots {B_n}^{u_{m-1}}(nu_m + 2) \in D$.

Therefore, in all cases, $|L(2)| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| > 1$ and $|\beta_1| = |{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$. In Step 1 of the first iteration, as $-1 < \beta_1 < 1$ and $0 < \frac{1-\beta_1}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu_1-\beta_1}{n} \rfloor = \lfloor -u_1 + \frac{1-\beta_1}{n} \rfloor = -u_1$ and $C = {A_n}^{v} = {A_n}^{-u_1}$. In Step 2, $L = CL = {A_n}^{-u_1}{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$ and $w = wC^{-1} = {A_n}^{u_1}$. In Step 3, as $L \neq I$, return Step 1.

Suppose that for $1 < j < m - 1$, $L = {A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd j and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{j-1}}$ in Step 2 of the $j - 1$th iteration or $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with even j and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{j-1}}$ in Step 2 of the $j - 1$th iteration.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with even j, then $L(2) = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j n u_j + 1} = \frac{1}{nu_j + \frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$, ${B_n}^{u_{m-1}}(0) = 0$ and ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$. By Theorem 4.1.4, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(0)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}$ ${B_n}^{u_{j+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-2}}(0)| > 1$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$, ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$. Put $\gamma = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1)$ and then $|\gamma| < 1$. Hence, by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(\gamma)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-2}}(\gamma)| > 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$. by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| < 1$. By Theorem 4.1.5, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| > 1$.

Therefore, in all cases, as $-1 < \frac{1}{\alpha_j} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{2}{n} \leq 1$, $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_j + \frac{1}{\alpha_j}}} + \frac{1}{n} \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = -u_j$ and $C = {B_n}^v = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j}{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{j-1}}{B_n}^{u_j}$. In Step 3, since $L \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd j, then $L(2) = {A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_j}(\beta_j) = nu_j + \beta_j$ where $\beta_j = {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$, ${B_n}^{u_{m-1}}(0) = 0$ and ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$. By Theorem 4.1.3, $|L(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| > 1$ and by Theorem 4.1.4, $|\beta_j| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(0)| < 1$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$, ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$. Put $\gamma = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1)$ and then $|\gamma| < 1$. By Theorem 4.1.3, $L(2) = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-2}}(\gamma) \in D^c$ and $|L(2)| > 1$. By Theorem 4.1.4, $|\beta| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(\gamma)| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$, by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m+2)| > 1$. By Theorem 4.1.6, $|\beta| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| < 1$.

Therefore, in all cases, $|L(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| > 1$ and $|\beta_j| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$. In Step 1 of the jth iteration, as $-1 < \beta_j < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu_j-\beta_j}{n} \rfloor = \lfloor -u_j + \frac{1-\beta_j}{n} \rfloor = -u_j$ and $C = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j}{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{j-1}}{A_n}^{u_j}$. In Step 3, as $L \neq I$, return Step 1.

If $j = m - 1$, then in Step 1 of the $m - 1$th iteration, $L = {B_n}^{u_{m-1}}{A_n}^{u_m}$ and $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$, so that $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = 0$. Hence the algorithm outputs ϵ and then the algorithm terminates.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$, so that $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$. Hence $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{-1}{-nu_{m-1}+1}} + \frac{1}{n} \rfloor = \lfloor -u_{m-1} + \frac{2}{n} \rfloor = -u_{m-1}$ and $C = {B_n}^v = {B_n}^{-u_{m-1}}$. In Step 2, $L = CL = {B_n}^{-u_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. Then in Step 1 of the $j = m$th iteration, $L = {A_n}^{u_m}$ and by Theorem 4.5.2, the X_n-representation algorithm ($z = 2$) outputs ϵ and then the algorithm terminates.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then in Step 1 of the $m - 1$th iteration, $L = {B_n}^{u_{m-1}}{A_n}^{u_m}$. ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}(nu_m + 2) \in D$. So $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) \in D$. Since $-1 < \frac{1}{nu_m+2} < 1$, $0 < 1 - \frac{1}{nu_m+2} < 2$ and $0 < \frac{1}{n}(1 - \frac{1}{nu_m+2}) < \frac{2}{n} \leq 1$, $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{nu_m+2}{nu_{m-1}(nu_m+2)+1}} + \frac{1}{n} \rfloor = \lfloor -u_{m-1} + \frac{1}{n}(1 - \frac{1}{nu_m+2}) \rfloor = -u_{m-1}$. Hence $C = {B_n}^v = {B_n}^{-u_{m-1}}$, in Step 2, $L = CL = {B_n}^{-u_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-2}}{A_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. In Step 1 of the $j = m$th iteration, $L = {A_n}^{u_m}$ and by Theorem 4.5.2, the algorithm outputs ϵ. Thus the algorithm terminates. □

Theorem 4.5.7 Let a matrix $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ where even $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$). If M is input to the X_n-representation algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ as the X_n-representation of M.

Proof In Step 1 of the first iteration, if $L = M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with even $m \geq 2$, then $L(\frac{1}{2}) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1+1} = \frac{1}{nu_1+\frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}(\alpha_1)| = |\frac{1}{nu_1+\frac{1}{\alpha_1}}| < 1$. By Theorem 4.1.3, $|\alpha_1| = |{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| > 1$ and $0 < \frac{1}{n}(1-\frac{1}{\alpha_1}) < \frac{2}{n} \leq 1$. So $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor -u_1 + \frac{1}{n}(1 - \frac{1}{\alpha_1}) \rfloor = -u_1$ and $C = {B_n}^v = {B_n}^{-u_1}$.

In Step 2, $L = CL = {B_n}^{-u_1}{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}$. In Step 3, since $L = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$, return Step 1.

Assume that for $1 < j < m$, $L = {A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd j and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ with even j and $w = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ with even j, then by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}(\beta_j)|$ $= |nu_j + \beta_j| > 1$ where $\beta_j = {B_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|\beta_j| < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$. So $v = \lfloor\frac{1-L(\frac{1}{2})}{n}\rfloor = \lfloor\frac{1-nu_j-\beta_j}{n}\rfloor = \lfloor -u_j + \frac{1-\beta_j}{n}\rceil = -u_j$ and $C = {A_n}^{v} = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j}{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{j-1}} {A_n}^{u_j}$. In Step 3, since $L = {B_n}^{u_{j+1}}{A_n}^{u_{j+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd j, $L(\frac{1}{2}) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j nu_j+1} = \frac{1}{nu_j+\frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_j}(\alpha_j)| = |\frac{1}{nu_j+\frac{1}{\alpha_j}}| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| > 1$ and $0 < \frac{1}{n}(1-\frac{1}{\alpha_j}) < \frac{2}{n} \leq 1$. So $v = \lfloor\frac{-1}{nL(\frac{1}{2})} + \frac{1}{n}\rfloor = \lfloor -u_j + \frac{1}{n}(1-\frac{1}{\alpha_j})\rfloor = -u_j$ and $C = {B_n}^{v} = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j}{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{j-1}}{B_n}^{u_j}$. In Step 3, since $L = {A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ $\neq I$, return Step 1.

If $j = m$, then in Step 1 of the $j = m$th iteration, $L = {A_n}^{u_m}$ and by Theorem 4.5.1, in Step 2, $L = CL = {A_n}^{-u_m}{A_n}^{u_m} = I$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$. In Step 3, since $L = I$, the algorithm outputs

${B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ as the X_n-representation of M and then the algorithm terminates. □

Theorem 4.5.8 Let a matrix $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ where even $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$). If M is input to the X_n-representation algorithm ($z = 2$), then the algorithm outputs ϵ.

Proof In Step 1 of the first iteration, if $L = M = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ with even $m \geq 2$, then $L(2) = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 n u_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}{B_n}^{u_3}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$, ${B_n}^{u_{m-1}}(0) = 0$ and ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$. By Theorem 4.1.4, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-2}}(0)| < 1$. By Theorem 4.1.3, $|\alpha_1| = |{A_n}^{u_2}{B_n}^{u_3}$ $\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{A_n}^{u_{m-2}}(0)| > 1$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$, ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$. Put $\gamma = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1)$ and then $|\gamma| < 1$. Hence, by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-2}}(\gamma)| < 1$. By Theorem 4.1.3, $|\alpha_1| = |{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ $(2)| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{A_n}^{u_{m-2}}(\gamma)| > 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$. by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}(nu_m{+}2)| < 1$. By Theorem 4.1.5, $|\alpha_1| = |{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}(nu_m + 2)| > 1$.

Therefore, in all cases, as $-1 < \frac{1}{\alpha_1} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_1}) < \frac{2}{n} \leq 1$, $v =$

$\lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n \frac{1}{nu_1 + \frac{1}{\alpha_1}}} + \frac{1}{n} \rfloor = \lfloor -u_1 + \frac{1}{n}(1 - \frac{1}{\alpha_1}) \rfloor = \lfloor -u_1 + \frac{1}{n}(1 - \frac{1}{\alpha_1}) \rfloor = -u_1$ and $C = {B_n}^v = {B_n}^{-u_1}$. In Step 2, $L = CL = {B_n}^{-u_1} {B_n}^{u_1} {A_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} = {A_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}$. In Step 3, since $L \neq I$, return Step 1.

Suppose that for $1 < j < m-1$, $L = {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ with odd j and $w = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j} {A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ with even j and $w = {B_n}^{u_1} {A_n}^{u_2} \cdots {A_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}$ with even j, then $L(2) = {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2) = {A_n}^{u_j}(\beta_j) = nu_j + \beta_j$ where $\beta_j = {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$, ${B_n}^{u_{m-1}}(0) = 0$ and ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$. By Theorem 4.1.3, $|L(2)| = |{A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-3}} {A_n}^{u_{m-2}}(0)| > 1$ and by Theorem 4.1.4, $|\beta_j| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)| = |{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(0)| < 1$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$, ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$. Put $\gamma = {B_n}^{u_{m-1}} {A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1)$ and then $|\gamma| < 1$. By Theorem 4.1.3, $L(2) = {A_n}^{u_1} {B_n}^{u_2} \cdots {A_n}^{u_{m-2}}(\gamma) \in D^c$ and $|L(2)| > 1$. By Theorem 4.1.4, $|\beta| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)| = |{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(\gamma)| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$, by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)| = |{A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m+2)| > 1$. By Theorem 4.1.6, $|\beta| = |{B_n}^{u_{j+1}} {A_n}^{u_{j+2}} \cdots$ ${B_n}^{u_{m-1}} {A_n}^{u_m}(2)| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| < 1$.

Therefore, in all cases, $|L(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| > 1$ and $|\beta_j| = |{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$. In Step 1 of the jth iteration, as $-1 < \beta_j < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu_j-\beta_j}{n} \rfloor = \lfloor -u_j + \frac{1-\beta_j}{n} \rfloor = -u_j$ and $C = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j}{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{j-1}}{A_n}^{u_j}$. In Step 3, as $L \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with odd j, then $L(2) = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j n u_j + 1} = \frac{1}{nu_j + \frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$, ${B_n}^{u_{m-1}}(0) = 0$ and ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$. By Theorem 4.1.4, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(0)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}$ ${B_n}^{u_{j+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-2}}(0)| > 1$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$, ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$. Put $\gamma = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1)$ and then $|\gamma| < 1$. Hence, by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-2}}(\gamma)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-2}}(\gamma)| > 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$. by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| < 1$. By Theorem 4.1.5, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| > 1$.

Therefore, in all cases, as $-1 < \frac{1}{\alpha_j} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{2}{n} \leq 1$, $v =$

$\lfloor\frac{-1}{nL(2)}+\frac{1}{n}\rfloor = \lfloor\frac{-1}{n\frac{1}{nu_j+\frac{1}{\alpha_j}}}+\frac{1}{n}\rfloor = \lfloor -u_j+\frac{1}{n}(1-\frac{1}{\alpha_j})\rfloor = \lfloor -u_j+\frac{1}{n}(1-\frac{1}{\alpha_j})\rfloor = -u_j$ and $C = {B_n}^v = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j}{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{j-1}}{B_n}^{u_j}$. In Step 3, since $L \neq I$, return Step 1.

If $j = m-1$, then in Step 1 of the $m-1$th iteration, $L = {B_n}^{u_{m-1}}{A_n}^{u_m}$ and $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$, so that $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = 0$. Hence the algorithm outputs ϵ and then the algorithm terminates.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{-1}{-nu_{m-1}+1} \in D$, so that $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$. Hence $v = \lfloor\frac{-1}{nL(2)} + \frac{1}{n}\rfloor = \lfloor\frac{-1}{n\frac{-1}{-nu_{m-1}+1}} + \frac{1}{n}\rfloor = \lfloor -u_{m-1} + \frac{2}{n}\rfloor = -u_{m-1}$ and $C = {B_n}^v = {B_n}^{-u_{m-1}}$. In Step 2, $L = CL = {B_n}^{-u_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. Then in Step 1 of the $j = m$th iteration, $L = {A_n}^{u_m}$ and by Theorem 4.5.2, the X_n-representation algorithm ($z = 2$) outputs ϵ and then the algorithm terminates.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then in Step 1 of the $m-1$th iteration, $L = {B_n}^{u_{m-1}}{A_n}^{u_m}$. ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}(nu_m + 2) \in D$. So $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) \in D$. Since $-1 < \frac{1}{nu_m+2} < 1$, $0 < 1 - \frac{1}{nu_m+2} < 2$ and $0 < \frac{1}{n}(1 - \frac{1}{nu_m+2}) < \frac{2}{n} \leq 1$, $v = \lfloor\frac{-1}{nL(2)} + \frac{1}{n}\rfloor = \lfloor\frac{-1}{n\frac{nu_m+2}{nu_{m-1}(nu_m+2)+1}} + \frac{1}{n}\rfloor = \lfloor -u_{m-1} + \frac{1}{n}(1 - \frac{1}{nu_m+2})\rfloor = -u_{m-1}$. Hence $C = {B_n}^v = {B_n}^{-u_{m-1}}$, in Step 2, $L = CL = {B_n}^{-u_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m}$ and $w = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. In Step 1 of the $j = m$th iteration, $L = {A_n}^{u_m}$ and by Theorem 4.5.2, the algorithm outputs ϵ. Thus the algorithm terminates. □

Theorem 4.5.9 Let a matrix $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ where even $m \geq 2$ and u_i is a nonzero integer $(i = 1, \cdots, m)$. If M is input to the X_n-representation algorithm $(z = \frac{1}{2})$, then the algorithm outputs ϵ.

Proof If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ with even $m \geq 2$ and a nonzero integer u_i $(i = 1, \cdots, m)$, then in Step 1 of the first iteration, $L(\frac{1}{2}) = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_1}(\beta_1) = nu_1+\beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then by definition of linear fractional transformations, ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. By Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ $(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| = |{A_n}^{u_1}(\beta_1)| > 1$ and by Theorem 4.1.4, $|\beta_1| = |{B_n}^{u_2}{A_n}^{u_3}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_2}{A_n}^{u_3}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.5, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-2}}(nu_{m-1} - 1)| = |{A_n}^{u_1}(\beta_1)| > 1$ and by Theorem 4.1.6, $|\beta_1| = |{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_2}{A_n}^{u_3}\cdots{B_n}^{u_{m-2}}(nu_{m-1} - 1)| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. So by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |{A_n}^{u_1}(\beta_1)| > 1$ and by Theorem 4.1.4, $|\beta_1| = |{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |nu_1+\beta_1| > 1$ and $|\beta_1| = |{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$. In Step 1 of the first iteration, as $-1 < \beta_1 < 1$ and $0 < \frac{1-\beta_1}{n} < \frac{2}{n} \leq 1$, $v = \lfloor\frac{1-L(\frac{1}{2})}{n}\rfloor = \lfloor\frac{1-nu_1-\beta_1}{n}\rfloor =$

$\lfloor -u_1 + \frac{1-\beta_1}{n} \rfloor = -u_1$ and $C = {A_n}^v = {A_n}^{-u_1}$. In Step 2, $L = CL = {A_n}^{-u_1}{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}$. In Step 3, since $L \neq I$, return Step 1.

Suppose that for $1 < j < m-2$, $L = {A_n}^{u_j}{B_n}^{u_{j-1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j}{A_n}^{u_{j-1}} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ with even j, then $L(\frac{1}{2}) = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j n u_j + 1} = \frac{1}{nu_j + \frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| = |{B_n}^{u_j}(\alpha_j)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.6, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| = |{B_n}^{u_j}(\alpha_j)| < 1$ and by Theorem 4.1.5, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| > 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. So by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |{B_n}^{u_j}(\alpha_j)| < 1$ and by Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}(\alpha_j)| <$

1 and $|\alpha_j| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| > 1$. In Step 1 of the jth iteration, since $-1 < \frac{1}{\alpha_j} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{2}{n} \leq 1$, $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_j + \frac{1}{\alpha_j}}} + \frac{1}{n} \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = -u_j$ and $C = {B_n}^v = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j}{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{j-1}}{B_n}^{u_j}$. In Step 3, as $L \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ with odd j, then $L(\frac{1}{2}) = {A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_j}(\beta_j) = nu_j + \beta_j$ where $\beta_j = {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. By Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| = |{A_n}^{u_j}(\beta_j)| = |nu_j + \beta_j| > 1$ and by Theorem 4.1.4, $|\beta_j| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.5, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| = |nu_j + \beta_j| > 1$ and by Theorem 4.1.6, $|\beta_j| = |{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. So by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$ and by Theorem 4.1.4, $|\beta_j| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}(\beta_j)| = |nu_j + \beta_j| > 1$ and $|\beta_j| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$. In Step 1 of the jth iteration, since $-1 < \beta_j < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$, $v =$

$\lfloor\frac{1-L(\frac{1}{2})}{n}\rfloor = \lfloor\frac{1-nu_j-\beta_j}{n}\rfloor = \lfloor -u_j + \frac{1-\beta_j}{n}\rfloor = -u_j$ and $C = {A_n}^v = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j}{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{j-1}}{A_n}^{u_j}$. In Step 3, since $L \neq I$, return Step 1.

If $j = m-2$, then $L = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-3}}$ in Step 2 of the $m-3$th iteration and consider $L(\frac{1}{2}) = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ in Step 1 of $m-2$th iteration as follows :

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. So $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |\frac{1}{nu_{m-2}}| < 1$

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$, ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$ and ${B_n}^{u_{m-2}}(nu_{m-1}-1) = \frac{nu_{m-1}}{(nu_{m-1}-1)nu_{m-2}+1} = \frac{1}{nu_{m-2}+\frac{1}{nu_{m-1}-1}} \in D$. So $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{m-2}}$ ${A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$.

In Step 1 of the $m-2$th iteration, $v = \lfloor\frac{-1}{nL(\frac{1}{2})} + \frac{1}{n}\rfloor = \lfloor\frac{-1}{n\frac{1}{nu_{m-2}}} + \frac{1}{n}\rfloor = \lfloor -u_{m-2} + \frac{1}{n}\rfloor = -u_{m-2}$ and $C = {B_n}^v = {B_n}^{-u_{m-2}}$. In Step 2, $L = CL = {B_n}^{-u_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2}\cdots$ ${B_n}^{u_{m-2}}$. In Step 3, as $L \neq I$, return Step 1.

If $j = m-1$, then $L = {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 1 of the $m-1$th iteration and consider $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$ and ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$. So $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = \infty$ and the algorithm outputs ϵ. Then the algorithm terminates.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. So $|L(\frac{1}{2})| = |{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |nu_{m-1} - 1| > 1$. Thus $v = \lfloor\frac{1-L(\frac{1}{2})}{n}\rfloor = \lfloor -u_{m-1} + \frac{2}{n}\rfloor = -u_{m-1}$ and $C = {A_n}^v = {A_n}^{-u_{m-1}}$. In Step 2, $L = CL = {A_n}^{-u_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $|L(\frac{1}{2})| = |{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |L(\frac{1}{2})| = |{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |nu_{m-1}+\frac{1}{nu_m+2}| > 1$. Hence $v = \lfloor\frac{1-L(\frac{1}{2})}{n}\rfloor = \lfloor\frac{1-nu_{m-1}-\frac{1}{nu_m+2}}{n}\rfloor$ $= \lfloor -u_{m-1} + \frac{1}{n}(1 - \frac{1}{nu_m+2})\rfloor = -u_{m-1}$ and $C = {A_n}^v = {A_n}^{-u_{m-1}}$. In Step 2, $L = CL = {A_n}^{-u_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. If $j = m$, then $L = {B_n}^{u_m}$ in Step 1 of the mth iteration. By Theorem 4.5.3, the X_n-representation algorithm outputs ϵ and then the algorithm terminates. □

Theorem 4.5.10 Let a matrix $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ where even $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$). If M is input to the X_n-representation algorithm ($z = 2$), then the algorithm outputs ${A_n}^{u_1}{B_n}^{u_2}\cdots$ ${A_n}^{u_{m-1}}{B_n}^{u_m}$ as the X_n-representation of M.

Proof Let $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ with even $m \geq 2$ and nonzero integer u_i ($i = 1, \cdots, m$). In Step 1 of the first iteration, $L(2) = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {A_n}^{u_1}(\beta_1) = nu_1+\beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| = |A_n^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| > 1$ and by Theorem 4.1.6, $|\beta_1| = |{B_n}^{u_2} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2)| < 1$. Since $-1 < \beta_1 < 1$ and $0 < \frac{1-\beta_1}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu_1-\beta_1}{n} \rfloor = \lfloor -u_1 + \frac{1-\beta_1}{n} \rfloor = -u_1$ and $C = {A_n}^v = {A_n}^{-u_1}$. In Step 2, $L = CL = {A_n}^{-u_1} {A_n}^{u_1} {B_n}^{u_2} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m} = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$. In Step 3, as $L \neq I$, return Step 1.

Assume that for $1 < j < m$, $L = {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$ and $w = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j} {A_n}^{u_{j-1}} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}$ and $w = {A_n}^{u_1} {B_n}^{u_2} \cdots {A_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration,.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j} {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$ with even j, then $L(2) = {B_n}^{u_j} {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j n u_j + 1} = \frac{1}{nu_j + \frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| = |{B_n}^{u_j} {A_n}^{u_{j+1}}$ $\cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2)| < 1$ and by Theorem 4.1.5, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$ $(2)| > 1$. Since $-1 < \frac{1}{\alpha_j} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{n}{2} \leq 1$, $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n \frac{1}{nu_j + \frac{1}{\alpha_j}}} + \frac{1}{n} \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = -u_j$ and $C = {B_n}^v = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j} {B_n}^{u_j} {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m} = {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1} {B_n}^{u_2} \cdots {A_n}^{u_{j-1}} {B_n}^{u_j}$. In Step 3, since $L \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$ with odd j, then $L(2) = nu_j + \beta_j$ where $\beta_j = {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| = |{A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2)| > 1$ and by Theorem 4.1.6, $|\beta_j| = |{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}(2)| < 1$. Since $-1 < \beta_j < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu_j-\beta_j}{n} \rfloor = \lfloor -u_j + \frac{1-\beta_j}{n} \rfloor = -u_j$ and $C = {A_n}^v = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j} {A_n}^{u_j} {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m} = {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{j-1}} {A_n}^{u_j}$. In Step 3, as $L \neq I$, return Step 1.

In the $j = m$th iteration, $L = {B_n}^{u_m}$ and by Lemma 4.1.2 and Theorem 4.5.4, $|L(2)| = |{B_n}^{u_m}(2)| < 1$, $v = -u_m$ and $C = {B_n}^v = {B_n}^{-u_m}$ in Step

1 of mth iteration. In Step 2, $L = CL = {B_n}^{-u_m}{B_n}^{u_m} = I$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$. In Step 3, as $L = I$, the algorithm outputs ${A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ as the X_n-representation of M and then the algorithm terminates. $\square$

Theorem 4.5.11 Let a matrix $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ where odd $m \geq 3$ and u_i is a nonzero integer ($i = 1, \cdots, m$). If M is input to the X_n-representation algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ.

Proof Let $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ with odd $m \geq 3$ and nonzero integer u_i ($i = 1, \cdots, m$). Then in Step 1 of the first iteration, $L = M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $L(\frac{1}{2}) = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 n u_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| = |{B_n}^{u_1}$ $(\alpha_1)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_2}{B_n}^{u_3}$ $\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.6, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-2}}(nu_{m-1} - 1)| = |{B_n}^{u_1}(\alpha_1)| < 1$ and by Theorem 4.1.5, $|\alpha_1| = |{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{B_n}^{u_{m-2}}(nu_{m-1} - 1)| > 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. So by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |{B_n}^{u_1}(\alpha_1)| < 1$ and by Theorem 4.1.3, $|\alpha_1| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}(\alpha_1)| < 1$ and $|\alpha_1| = |{A_n}^{u_2}{B_n}^{u_3}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| > 1$.

In Step 1 of the first iteration, since $-1 < \frac{1}{\alpha_1} < 1$ and $0 < \frac{1}{n}(1-\frac{1}{\alpha_1}) < \frac{2}{n} \leq 1$, $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_1+\frac{1}{\alpha_1}}} + \frac{1}{n} \rfloor = \lfloor -u_1 + \frac{1}{n}(1-\frac{1}{\alpha_1}) \rfloor = -u_1$ and $C = {B_n}^v = {B_n}^{-u_1}$. In Step 2, $L = CL = {B_n}^{-u_1}{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}$. In Step 3, as $L \neq I$, return Step 1.

Suppose that for $1 < j < m-2$, $L = {A_n}^{u_j}{B_n}^{u_{j-1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j}{A_n}^{u_{j-1}}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m}$ and $w = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ even j, then $L(\frac{1}{2}) = {A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_j}(\beta_j) = nu_j + \beta_j$ where $\beta_j = {B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. By Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| = |{A_n}^{u_j}(\beta_j)| = |nu_j+\beta_j| > 1$ and by Theorem 4.1.4, $|\beta_j| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1}-1 \in D^c$. By Theorem 4.1.5, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{B_n}^{u_{m-2}}(nu_{m-1}-1)| = |nu_j+\beta_j| > 1$ and by Theorem 4.1.6, $|\beta_j| = |{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}}\cdots{B_n}^{u_{m-2}}(nu_{m-1}-1)| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. So by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$ and by Theorem 4.1.4, $|\beta_j| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_j}(\beta_j)| = |nu_j + \beta_j| > 1$ and $|\beta_j| = |{B_n}^{u_{j+1}}{A_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$.

In Step 1 of the jth iteration, since $-1 < \beta_j < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(\frac{1}{2})}{n} \rfloor = \lfloor \frac{1-nu_j-\beta_j}{n} \rfloor = \lfloor -u_j + \frac{1-\beta_j}{n} \rfloor = -u_j$ and $C = {A_n}^{v} = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j}{A_n}^{u_j}{B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{j-1}}{A_n}^{u_j}$. In Step 3, since $L \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ with odd j, then $L(\frac{1}{2}) = {B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j nu_j+1} = \frac{1}{nu_j+\frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| = |{B_n}^{u_j}(\alpha_j)| < 1$. By Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1}-1 \in D^c$. By Theorem 4.1.6, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| = |{B_n}^{u_j}(\alpha_j)| < 1$ and by Theorem 4.1.5, $|\alpha_j| = |{A_n}^{u_{j+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}} \cdots {B_n}^{u_{m-2}}(nu_{m-1}-1)| > 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. So by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |{B_n}^{u_j}(\alpha_j)| < 1$ and by Theorem 4.1.3, $|\alpha_j| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_j}(\alpha_j)| < 1$ and $|\alpha_j| = |{A_n}^{u_{j+1}}{B_n}^{u_{j+2}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| > 1$. In Step 1 of the jth iteration, since $-1 < \frac{1}{\alpha_j} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{2}{n} \leq 1$, $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_j+\frac{1}{\alpha_j}}} + \frac{1}{n} \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = -u_j$ and $C = {B_n}^{v} = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j}{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{j-1}}{B_n}^{u_j}$. In Step 3, as $L \neq I$, return Step 1.

If $j = m-2$, then $L = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-3}}$ in Step 2 of the $m-3$th iteration and consider $L(\frac{1}{2}) = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ in Step 1 of $m-2$th iteration as follows :

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}(\infty) = \frac{\infty}{nu_{m-2}\infty+1} = \frac{1}{nu_{m-2}} \in D$. So $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |\frac{1}{nu_{m-2}}| < 1$

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$, ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$ and ${B_n}^{u_{m-2}}(nu_{m-1} - 1) = \frac{nu_{m-1}}{(nu_{m-1}-1)nu_{m-2}+1} = \frac{1}{nu_{m-2}+\frac{1}{nu_{m-1}-1}} \in D$. So $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{m-2}}$ ${A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

Therefore, in all cases, $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| < 1$.

In Step 1 of the $m-2$th iteration, $v = \lfloor \frac{-1}{nL(\frac{1}{2})} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_{m-2}}} + \frac{1}{n} \rfloor = \lfloor -u_{m-2} + \frac{1}{n} \rfloor = -u_{m-2}$ and $C = {B_n}^v = {B_n}^{-u_{m-2}}$. In Step 2, $L = CL = {B_n}^{-u_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-2}}$. In Step 3, as $L \neq I$, return Step 1.

If $j = m-1$, then $L = {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 1 of the $m-1$th iteration and consider $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = \infty$ and ${A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$. So $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = \infty$ and the algorithm outputs ϵ. Then the algorithm terminates.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and ${A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. So $|L(\frac{1}{2})| = |{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |nu_{m-1} - 1| > 1$. Thus $v = \lfloor \frac{1-L(\frac{1}{2})}{n} \rfloor = \lfloor -u_{m-1} + \frac{2}{n} \rfloor = -u_{m-1}$ and $C = {A_n}^v = {A_n}^{-u_{m-1}}$. In Step 2, $L = CL = {A_n}^{-u_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. If $j = m$, then $L = {B_n}^{u_m}$ in Step 1 of the mth iteration. By Theorem 4.5.3, the X_n-representation algorithm outputs ϵ and then the algorithm terminates.

If neither $n = 2$ and $u_m = -1$ nor $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $|L(\frac{1}{2})| = |{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |L(\frac{1}{2})| = |{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |nu_{m-1} + \frac{1}{nu_m+2}| > 1$. Hence $v = \lfloor \frac{1-L(\frac{1}{2})}{n} \rfloor = \lfloor \frac{1-nu_{m-1}-\frac{1}{nu_m+2}}{n} \rfloor = \lfloor -u_{m-1} + \frac{1}{n}(1 - \frac{1}{nu_m+2}) \rfloor = -u_{m-1}$ and $C = {A_n}^v = {A_n}^{-u_{m-1}}$. In Step 2, $L = CL = {A_n}^{-u_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m}$ and $w = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$. In Step 3, as $L \neq I$, return Step 1. If $j = m$, then $L = {B_n}^{u_m}$ in Step 1 of the mth iteration. By Theorem 4.5.3, the X_n-representation algorithm outputs ϵ and then the algorithm terminates. □

Theorem 4.5.12 Let a matrix $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ where odd $m \geq 3$ and u_i is a nonzero integer $(i = 1, \cdots, m)$. If M is input to the X_n-representation algorithm $(z = 2)$, then the algorithm outputs ${B_n}^{u_1}{A_n}^{u_2}$ $\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ as the X_n-representation of M.

Proof Let $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ with odd $m \geq 3$ and nonzero integer u_i $(i = 1, \cdots, m)$. Then in Step 1 of the first iteration, $L = M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $L(2) = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 n u_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| < 1$ and by Theorem 4.1.5, $|\alpha_1| = |{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| > 1$. Since $-1 < \frac{1}{\alpha_1} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_1}) < \frac{n}{2} \leq 1$, $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_1+\frac{1}{\alpha_1}}} + \frac{1}{n} \rfloor = \lfloor -u_1 + \frac{1}{n}(1 - \frac{1}{\alpha_1}) \rfloor = -u_1$ and $C = {B_n}^{v} = {B_n}^{-u_1}$. In Step 2, $L = CL = {B_n}^{-u_1}{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}$. In Step 3, since $L \neq I$, return Step 1.

Assume that for $1 < j < m$, $L = {A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{j-1}}$ in Step 2 of the $j-1$th iteration or $L = {B_n}^{u_j}{A_n}^{u_{j-1}}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m}$ and $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{j-1}}$ in Step 2 of the $j - 1$th iteration,.

In Step 1 of the jth iteration, if $L = {A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ with even j, then $L(2) = nu_j + \beta_j$ where $\beta_j = {B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| = |{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| > 1$ and by Theorem 4.1.6, $|\beta_j| = |{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| < 1$. Since $-1 < \beta_j < 1$ and $0 < \frac{1-\beta_j}{n} < \frac{2}{n} \leq 1$, $v = \lfloor \frac{1-L(2)}{n} \rfloor = \lfloor \frac{1-nu_j-\beta_j}{n} \rfloor = \lfloor -u_j + \frac{1-\beta_j}{n} \rfloor = -u_j$ and $C = {A_n}^{v} = {A_n}^{-u_j}$. In Step 2, $L = CL = {A_n}^{-u_j}{A_n}^{u_j}{B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{j-1}}{A_n}^{u_j}$. In Step 3, as $L \neq I$, return Step 1.

In Step 1 of the jth iteration, if $L = {B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ with odd j, then $L(2) = {B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_j}(\alpha_j) = \frac{\alpha_j}{\alpha_j n u_j + 1} = \frac{1}{nu_j + \frac{1}{\alpha_j}}$ where $\alpha_j = {A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| = |{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| < 1$ and by Theorem 4.1.5, $|\alpha_j| = |{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)| > 1$. Since $-1 < \frac{1}{\alpha_j} < 1$ and $0 < \frac{1}{n}(1 - \frac{1}{\alpha_j}) < \frac{n}{2} \leq 1$, $v = \lfloor \frac{-1}{nL(2)} + \frac{1}{n} \rfloor = \lfloor \frac{-1}{n\frac{1}{nu_j+\frac{1}{\alpha_j}}} + \frac{1}{n} \rfloor = \lfloor -u_j + \frac{1}{n}(1 - \frac{1}{\alpha_j}) \rfloor = -u_j$ and $C = {B_n}^{v} = {B_n}^{-u_j}$. In Step 2, $L = CL = {B_n}^{-u_j}{B_n}^{u_j}{A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{j+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ and $w = wC^{-1} = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{j-1}}{B_n}^{u_j}$. In Step 3, since $L \neq I$, return Step 1.

In the $j = m$th iteration, $L = {B_n}^{u_m}$ and by Lemma 4.1.2 and Theorem 4.5.4, $|L(2)| = |{B_n}^{u_m}(2)| < 1$, $v = -u_m$ and $C = {B_n}^{v} = {B_n}^{-u_m}$ in Step 1 of mth iteration. In Step 2, $L = CL = {B_n}^{-u_m}{B_n}^{u_m} = I$ and $w = wC^{-1} = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$. In Step 3, as $L = I$, the algorithm outputs ${B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ as the X_n-representation of M and then the algorithm terminates. □

Chapter 5

$X(n, S)$-Representation Algorithm

Let $n \geq 2$ be a natural number and S be an ordered set of integers, where $S = \{s_1, s_2, \cdots, s_t\}$. For each $s_i \in S$, define a matrix $M_i \in \mathrm{SL}_2(\mathbb{Z})$ by

$$M_i = {A_n}^{-s_i} B_n {A_n}^{s_i}$$

and define

$$X(n, S) = \{M_1, M_2, \cdots, M_t\}.$$

$G(n, S) = \langle M_1, M_2, \cdots, M_t \rangle$. Then it is proved that $X(n, S)$ is a free basis in Section 5.1 and $G(n, S)$ is a free group, freely generated by $\{M_1, M_2, \cdots, M_t\}$. Thus every element of $G(n, S)$ can be represented by elements of $X(n, S)^{\pm}$ and it is called the $X(n, S)$-representation. Since $G(n, S)$ is a subgroup of Γ_n, every element of $G(n, S)$ has the X_n-representation as an element of Γ_n and thus each element M_i has the X_n-representation ${A_n}^{-s_i} B_n {A_n}^{s_i}$. In Chapter 4, given $M \in \Gamma_n$, it is shown that the X_n-representation algorithm computes the X_n-representation of M. Further, the $X(n, S)$-representation algorithm given by Grigoriev and Ponomarenko [7] computes the $X(n, S)$-representation of an element M of $G(n, S)$ as a reduced word in $X(n, S)^{\pm}$ when the X_n-representation of $M \in G(n, S)$ is provided. So, in this chapter, we consider the $X(n, S)$-representation algorithm and the following sections are as follows.

In Section 5.1, we show that $X(n, S)$ is a free generating set for $G(n, S)$. In Section 5.2, we analyze the $X(n, S)$-representation algorithm. In Section 5.3, we modify the $X(n, S)$-representation algorithm to makes it efficient. In Section 5.4, we implement the modified $X(n, S)$-representation algorithm by programming it. In Section 5.5, we justify the modified $X(n, S)$-representation algorithm.

5.1 Free Basis $X(n, S)$

This section presents that $X(n, S)$ is a free basis of $G(n, S)$. Let F be a free group with a generating set X and $U = \{u_i \mid i \in \mathbb{N}\}$ be a subset of a free group F. We introduce elementary Nielsen transformation on a set $U = \{u_i \mid i \in \mathbb{N}\}$ as follows: [14]

1. replace some u_i by ${u_i}^{-1}$

2. replace some u_i by $u_i u_j$ where $j \neq i$;

3. delete some u_i where $u_i = 1$

where 1 denotes the empty word. A product of such elementary transformations is called Nielsen transformation. If all triples $v_1, v_2, v_3 \in U^{\pm}$ satisfy the following conditions: [14]

1. $v_1 \neq 1$

2. $v_1 v_2 \neq 1$ implies $|v_1 v_2| \geq |v_1|$, $|v_2|$

3. $v_1 v_2 \neq 1$ and $v_2 v_3 \neq 1$ implies $|v_1 v_2 v_3| > |v_1| - |v_2| + |v_3|$,

then U is called Nielsen reduced. The Nielsen reduced set plays an important role as it is a free generating set for the subgroup that it generates. Therefore we show that $X(n, S)$ satisfies the three conditions to be Nielsen reduced in the following.

Theorem 5.1.1 Given $n \geq 2$ and a finite set $S \subset \mathbb{Z}$, $X(n, S)$ is Nielsen reduced.

Proof Given Γ_n freely generated by two matrices A_n and B_n and $X(n, S) \subset \Gamma_n$, let $v_1 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s}$, $v_2 = {A_n}^{-t}{B_n}^{\beta}{A_n}^{t}$ and $v_3 = {A_n}^{-u}{B_n}^{\gamma}{A_n}^{u} \in X(n, S)^{\pm}$ where α, β, $\gamma \in \{1, -1\}$ and s, t, $u \in S$.

1. For $v_1 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s}$, if $s = 0$, then $v_1 = {B_n}^{\alpha} \neq 1$ and if $s \neq 0$, then $|v_1| = |{A_n}^{-s}{B_n}^{\alpha}{A_n}^{s}| = 2|s| + 1 \neq 0$ and so $v_1 \neq 1$.

2. For $v_1 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s}$ and $v_2 = {A_n}^{-t}{B_n}^{\beta}{A_n}^{t}$, $v_1 v_2 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s-t}$
${B_n}^{\beta}{A_n}^{t}$.

(Case 1) If $s = t$ and $\alpha = \beta$, then $v_1 v_2 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s-t}{B_n}^{\beta}{A_n}^{t} = {A_n}^{-s}{B_n}^{\alpha+\beta}{A_n}^{t}$ and $|v_1 v_2| = |{A_n}^{-s}{B_n}^{\alpha+\beta}{A_n}^{t}| = 2|s| + 2$. Thus $v_1 v_2 \neq 1$ and as $|v_1| = 2|s| + 1$ and $|v_2| = 2|t| + 1$, $|v_1 v_2| \geq |v_1|, |v_2|$.

(Case 2) If $s = t$ and $\alpha \neq \beta$, then $v_1 v_2 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s-t}{B_n}^{\beta}{A_n}^{t} = I$ and so $v_1 v_2 = 1$. Hence, this case can not be considered.

(Case 3) If $s \neq t$ and $\alpha = \pm\beta$, then $v_1 v_2 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s-t}{B_n}^{\beta}{A_n}^{t}$ and $|v_1 v_2| = |s| + |t| + |s - t| + 2 \geq 2|s| + 2$ by the triangle inequality $|t| + |s - t| \geq |s|$. $|v_1 v_2| = |{A_n}^{-s}{B_n}^{\alpha}{A_n}^{s-t}{B_n}^{\beta}{A_n}^{t}| = |s| + |t| + |s - t| + 2 = |s| + |t| + |t - s| + 2 \geq 2|t| + 2$ by the triangle inequality $|s| + |t - s| \geq |t|$. As $|v_1| = 2|s| + 1$ and $|v_2| = 2|t| + 1$, $|v_1 v_2| \geq |v_1|, |v_2|$.

3. For $v_1 = {A_n}^{-s}{B_n}^{\alpha}{A_n}^{s}$, $v_2 = {A_n}^{-t}{B_n}^{\beta}{A_n}^{t}$ and $v_3 = {A_n}^{-u}{B_n}^{\gamma}{A_n}^{u}$, $v_1 v_2 v_3 =$

$$A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s}A_n{}^{-t}B_n{}^{\beta}A_n{}^{t}A_n{}^{-u}B_n{}^{\gamma}A_n{}^{u} = A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s-t}B_n{}^{\beta}A_n{}^{t-u}B_n{}^{\gamma}A_n{}^{u}.$$

(Case 1) If $s = t$, $\alpha = \beta$, $t = u$ and $\beta = \gamma$, then $|v_1v_2v_3| = |A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s-t}B_n{}^{\beta}$ $A_n{}^{t-u}B_n{}^{\gamma}A_n{}^{u}| = A_n{}^{-s}B_n{}^{\alpha+\beta+\gamma}A_n{}^{u} = 2|s|+3$ and $|v_1|-|v_2|+|v_3| = 2|s|+1-2|t|-1+2|u|+1 = 2|s|+1$. Hence $|v_1v_2v_3| > |v_1|-|v_2|+|v_3|$.

(Case 2) If $s = t$, $\alpha = \beta$, $t \neq u$ and $\beta = \pm\gamma$, then $|v_1v_2v_3| = |A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s-t}B_n{}^{\beta}$ $A_n{}^{t-u}B_n{}^{\gamma}A_n{}^{u}| = |A_n{}^{-s}B_n{}^{\alpha+\beta}A_n{}^{t-u}B_n{}^{\gamma}A_n{}^{u}| = |s|+2+|t-u|+1+|u| = |s|+2+|s-u|+1+|u| = |s|+2+|u-s|+1+|u| \geq 2|u|+3$ by the triangle inequality $|s|+|u-s| \geq u$ and $|v_1|-|v_2|+|v_3| = 2|s|+1-2|t|-1+2|u|+1 = 2|u|+1$. Thus $|v_1v_2v_3| > |v_1|-|v_2|+|v_3|$.

(Case 3) If $s \neq t$, $\alpha = \pm\beta$, $t = u$ and $\beta = \gamma$, then $|v_1v_2v_3| = |A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s-t}B_n{}^{\beta}$ $A_n{}^{t-u}B_n{}^{\gamma}A_n{}^{u}| = |A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s-t}B_n{}^{\beta+\gamma}A_n{}^{u}| = |s|+1+|s-t|+2+|u| = |s|+1+|s-u|+2+|u| \geq 2|s|+3$ by the triangle inequality $|s-u|+|u| \geq |s|$ and $|v_1|-|v_2|+|v_3| = 2|s|+1-2|t|-1+2|u|+1 = 2|s|+1$. Hence $|v_1v_2v_3| > |v_1|-|v_2|+|v_3|$.

(Case 4) If $s \neq t$, $\alpha = \pm\beta$, $t \neq u$ and $\beta = \pm\gamma$, then $|v_1v_2v_3| = |A_n{}^{-s}B_n{}^{\alpha}A_n{}^{s-t}B_n{}^{\beta}$ $A_n{}^{t-u}B_n{}^{\gamma}A_n{}^{u}| = |s|+1+|s-t|+1+|t-u|+1+|u| = |s|+|s-t|+|t-u|+|u|+3 \geq |s|+|s|-|t|+|u|-|t|+|u|+3 = 2|s|-2|t|+2|u|+3$ and $|v_1|-|v_2|+|v_3| = 2|s|-2|t|+2|u|+1$. Therefore $|v_1v_2v_3| > |v_1|-|v_2|+|v_3|$. □

Theorem 5.1.2 [14] If F is a free group with a basis X and a subset Y of F is Nielsen reduced and $w = y_1 \cdots y_m$, $(m \geq 0)$, $y_i \in Y^{\pm}$ and all $y_iy_{i+1} \neq 1$, then $|w| \geq m$.

Theorem 5.1.3 [14] Let X be a subset of a group G such that $X \cap X^{-1} \neq \emptyset$.

Then X is a basis for a free subgroup of G if and only if no product $w = x_1 \cdots x_n$ is trivial, where $n \geq 1$, $x_i \in X^{\pm}$, and all $x_i x_{i+1} \neq 1$.

Theorem 5.1.4 Given $n \geq 2$ and a finite set $S \subset \mathbb{Z}$, $X(n, S)$ is a free basis for $G(n, S)$.

Proof By Theorem 5.1.1, $X(n, S)$ is Nielsen reduced and we replace the set Y in Theorem 5.1.2 by $X(n, S)$. Then given $w = w_1 \cdots w_m$ with $m \geq 0$, $w_i \in X(n, S)^{\pm}$ and all $w_i w_{i+1} \neq 1$, $|w| \geq m$ and by Theorem 5.1.3 it is proven that $X(n, S)$ is a free basis for $G(n, S)$.

5.2 Analysis of $X(n, S)$-Representation Algorithm

We describe the $X(n, S)$-representation algorithm of Grigoriev and Ponomarenko and we do analysis of the $X(n, S)$-representation algorithm. As the $X(n, S)$-representation algorithm computes the $X(n, S)$-representation of $M \in G(n, S)$, we can do test the membership for a subgroup $G(n, S)$ of Γ_n. The algorithm takes the X_n-representation of $g \in \Gamma_n$ as an input and it outputs $(i_g, w_g) \in \{0, 1\} \times W_{X(n,S)}$ such that $g \in G(n, S)$ if and only if $i_g = 1$ and w_g is the $X(n, S)$-representation of g.

$X(n, S)$**-Representation Algorithm**

For a given $g \in \Gamma_n$,

Step 1 If $g = 1_{X_n}$, then output $(1, 1_{X(n,S)})$. Otherwise, let $u = {A_n}^a {B_n}^b {A_n}^c u_0$ be the X_n-representation of g where $a, b, c \in \mathbb{Z}$ and $u_0 \in W_{X_n}$

Step 2 If either $-a \notin S$ or $(-a, b) \in S \times \{0\}$, then output $(0, 1_{X(n,S)})$. Otherwise set $u = {A_n}^{a+c} u_0$.

Step 3 Recursively find (i_h, w_h) where $h = \bar{u}$. If $i_h = 0$, then output (i_h, w_h).

Step 4 Output $(1, w_g)$ where $w_g = vw_h$ with $v = {A_n}^a {B_n}^b {A_n}^{-a}$.

The idea is to obtain the form ${A_n}^a {B_n}^b {A_n}^{-a}$ with $-a \in S$ and a nonzero integer b by setting $u = {A_n}^a {B_n}^b {A_n}^c u_0$ with $a, b, c \in \mathbb{Z}$ and $u_0 \in W_{X_n}$ in Step 1. If the X_n-representation of g is the empty word 1_{X_n} as an input, then the $X(n, S)$-representation algorithm outputs a pair $(1, 1_{X(n,S)})$. If $-a \in S$ and $b \neq 0$, then the algorithm sets $u = {A_n}^{a+c} u_0$ in Step 2 and $v = {A_n}^a {B_n}^b {A_n}^{-a}$ in Step 4. If $-a \notin S$ or $(-a, b) \in S \times \{0\}$, then the algorithm outputs $(0, 1_{X(n,S)})$ in Step 2 and it means that $g \notin G(n, S)$. For instance, if $g = {A_n}^u \in \Gamma_n$, then in Step 1, $u = {A_n}^u {B_n}^0 {A_n}^0 u_0$ where $u_0 = 1_{X_n} \in W_{X_n}$ and the exponent of B_n is 0. So the algorithm outputs a pair $(0, 1_{X(n,S)})$.

In Step 3, for $h = \bar{u}$ and $|h| < |g|$, the algorithm works recursively to find the $X(n, S)$-representation of h. In order to compute (i_h, w_h), each iteration repeats Step 1 and Step 2. Hence the number of iterations is at most the number of terms of the X_n-representation of g. In Step 3, if $i_h = 0$, then output $(i_h, w_h) = (0, 1_{X_{n,S}})$. It means that $h \notin G(n, S)$ and so $g \notin G(n, S)$. If $(i_g, w_h) = (1, w_h)$ in Step 3, then in Step 4, the algorithm concatenates $v = {A_n}^a {B_n}^b {A_n}^{-a}$ and w_h and the algorithm outputs $(1, w_g) = (1, vw_h)$ where w_h is the $X(n, S)$-representation of h.

5.3 Modified $X(n, S)$-Representation Algorithm

The purpose of this section is to make the $X(n, S)$-representation algorithm practical and efficient for implementation. So we modify the $X(n, S)$-representation algorithm of Grigoriev and Ponomarenko. We consider how the $X(n,S)$-representation algorithm works according as concrete X_n-representations. For a given element $M \in G(n,S)$, let the X_n-representation of M be

$$A_n{}^{u_1} B_n{}^{u_2} A_n{}^{u_3} B_n{}^{u_4} \cdots A_n{}^{u_{m-2}} B_n{}^{u_{m-1}} A_n{}^{u_m}$$

with $m \geq 3$ and each nonzero u_i $(i = 1, \cdots, m)$. Then the $X(n,S)$-representation of M is

$$A_n{}^{-s_{a_1}} B_n{}^{u_2} A_n{}^{s_{a_1}} A_n{}^{-s_{a_2}} B_n{}^{u_4} A_n{}^{-s_{a_2}} \cdots A_n{}^{s_{a_{\frac{m-3}{2}}}} A_n{}^{-s_{a_{\frac{m-1}{2}}}} B_n{}^{u_{m-1}} A_n{}^{s_{a_{\frac{m-1}{2}}}}$$

where for $i = 1, \cdots \frac{m-1}{2}$, $a_i \in \{1, \cdots, t\}$, $s_{a_i} \in S$, $-u_1 = s_{a_1}$, $u_{2i-1} = s_{a_{i-1}} - s_{a_i}$ $(i \geq 2)$, $s_{a_{i-1}} \neq s_{a_i}$ $(i \geq 2)$ and $u_m = s_{a_{\frac{m-1}{2}}}$. Therefore, given the X_n-representation $A_n{}^{u_1} B_n{}^{u_2} \cdots B_n{}^{u_{m-1}} A_n{}^{u_m}$ with $m \geq 3$ and each nonzero u_i $(i = 1, \cdots, m)$, we have a formula $\frac{m-1}{2}$ for computing the number of terms of the $X(n,S)$-representation of M. Moreover, the X_n-representation $A_n{}^{u_1} B_n{}^{u_2} \cdots B_n{}^{u_{m-1}} A_n{}^{u_m}$ can be written as

$$\begin{aligned}
&A_n{}^{u_1} B_n{}^{u_2} A_n{}^{-u_1} \\
&A_n{}^{u_1+u_3} B_n{}^{u_4} A_n{}^{-(u_1+u_3)} \\
&A_n{}^{u_1+u_3+u_5} B_n{}^{u_6} A_n{}^{-(u_1+u_3+u_5)} \\
&A_n{}^{u_1+u_3+u_5+u_7} B_n{}^{u_8} A_n{}^{-(u_1+u_3+u_5+u_7)} \\
&\vdots \\
&A_n{}^{u_1+u_3+u_5+u_7+\cdots+u_{2i-1}+\cdots+u_{m-2}} B_n{}^{u_{m-1}} A_n{}^{-(u_1+u_3+u_5+u_7+\cdots+u_{2i-1}+\cdots+u_{m-2})} \\
&A_n{}^{(u_1+u_3+u_5+u_7+\cdots+u_{2i-1}+\cdots+u_{m-2}+u_m)}
\end{aligned}$$

where the exponent of the last term A_n is $u_m = -(u_1 + u_3 + u_5 + u_7 + \cdots + u_{2i-1} + \cdots + u_{m-2})$.

If we regard the description above as the $X(n,S)$-representation

$$A_n{}^{-s_{a_1}} B_n{}^{u_2} A_n{}^{s_{a_1}} A_n{}^{-s_{a_2}} B_n{}^{u_4} A_n{}^{-s_{a_2}} \cdots A_n{}^{s_{a_{\frac{m-3}{2}}}} A_n{}^{-s_{a_{\frac{m-1}{2}}}} B_n{}^{u_{m-1}} A_n{}^{s_{a_{\frac{m-1}{2}}}}$$

of M, then we have the following

$$\begin{aligned}
u_1 &= -s_{a_1} \\
u_3 &= s_{a_1} - s_{a_2} \\
u_5 &= s_{a_2} - s_{a_3} \\
&\vdots \\
u_{2i-1} &= s_{a_{i-1}} - s_{a_i} \\
&\vdots \\
u_{m-2} &= s_{a_{\frac{m-3}{2}}} - s_{a_{\frac{m-1}{2}}} \\
u_m &= s_{a_{\frac{m-1}{2}}}
\end{aligned}$$

and we restate it as follows :

$$\begin{aligned}
s_{a_1} &= -u_1 \\
s_{a_2} &= -u_1 - u_3 \\
s_{a_3} &= -u_1 - u_3 - u_5 \\
&\vdots \\
s_{a_i} &= -u_1 - u_3 - u_5 - \cdots - u_{2_i-1} \\
&\vdots \\
s_{a_{\frac{m-1}{2}}} &= -u_1 - u_3 - u_5 - \cdots - u_{m-2}.
\end{aligned}$$

Therefore, we have an explicit formula

$$s_{a_i} = -(u_1 + u_3 + u_5 + \cdots + u_{2i-1})$$

to compute the $X(n, S)$-representation of M. So now we describe the modified $X(n, S)$-representation algorithm.

Modified $X(n, S)$-Representation Algorithm

For odd $m \geq 3$, let ${A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ be the X_n-representation of M as an input to the modified $X(n, S)$-representation algorithm where $u_2, \cdots, u_{m-1}$ are nonzero integers.

```
Step 0
$\mathbf{i \leftarrow 1}$.
$\mathbf{w \leftarrow {A_n}^{u_1}{B_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}}$
$\mathbf{w = 1_{X_n} \Rightarrow}$ output $\mathbf{1_{X(n,S)}}$.

Step 1
$\mathbf{e_i \leftarrow -(u_1+u_3+u_5+\cdots+u_{2i-1})}$
$\mathbf{e_i \notin S \Rightarrow}$ output $\epsilon$.
$\mathbf{e_i \in S \Rightarrow C_i \leftarrow {A_n}^{-e_i}{B_n}^{u_{2i}}{A_n}^{e_i}}$.

Step 2
$\mathbf{w \leftarrow {C_i}^{-1}w}$
$\mathbf{w = 1_{X_n} \Rightarrow}$ output $\mathbf{C_1C_2\cdots C_i}$.
Otherwise,
$\mathbf{i \leftarrow i+1}$
$\mathbf{i = \frac{m+1}{2} \Rightarrow}$ output $\epsilon$
return Step 1.
```

Now we explain how the $X(n,S)$-representation algorithm works. Assume that the X_n-representation of M is given. Then the $X(n,S)$-representation algorithm takes the X_n-representation

$${A_n}^{u_1}{B_n}^{u_2}{A_n}^{u_3}{B_n}^{u_4}\cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{u_m}$$

as an input and outputs the $X(n,S)$-representation

$${A_n}^{u_1}{B_n}^{u_2}{A_n}^{-u_1}{A_n}^{u_1+u_3}{B_n}^{u_4}{A_n}^{-u_1-u_3}\cdots {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{-u_1-u_3-u_5-\cdots-u_{m-2}}.$$

We need to say something about what the input should be if the X_n-representation of M is not the form ${A_n}^{u_1}{B_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$.

If the X_n-representation of $M \in \Gamma_n$ is

$$B_n{}^{v_1} A_n{}^{v_2} \cdots B_n{}^{v_{p-1}} A_n{}^{v_p} \text{ (even } p),$$

then input the X_n-representation

$$A_n{}^{u_1} B_n{}^{u_2} A_n{}^{u_3} \cdots B_n{}^{u_{m-1}} A_n{}^{u_m}$$

to the $X(n, S)$-representation algorithm where $u_1 = 0$ and $u_i = v_{i-1}(i = 2, 3, \cdots, m = p + 1)$.

If the X_n-representation of $M \in \Gamma_n$ is

$$A_n{}^{v_1} B_n{}^{v_2} \cdots A_n{}^{v_{p-1}} B_n{}^{v_p} \text{ (even } p),$$

then input the X_n-representation

$$A_n{}^{u_1} B_n{}^{u_2} \cdots B_n{}^{u_{m-1}} A_n{}^{m}$$

to the $X(n, S)$-representation algorithm where $u_i = v_i(i = 1, 2, \cdots, m-1 = p)$ and $u_m = 0$.

If the X_n-representation of $M \in \Gamma_n$ is

$$B_n{}^{v_1} A_n{}^{v_2} \cdots A_n{}^{v_{p-1}} B_n{}^{v_p} \text{ (odd } p),$$

then input the X_n-representation

$$A_n{}^{u_1} B_n{}^{u_2} \cdots B_n{}^{u_{m-1}} A_n{}^{m}$$

to the $X(n, S)$-representation algorithm where $u_1 = 0$, $u_i = v_{i-1}(i = 2, \cdots, m - 1 = p + 1)$ and $u_m = 0$.

5.4 Programming Implementation

This section presents implementation of the modified $X(n, S)$-representation algorithm. By using Maple version 6, we make a program. So we demonstrate how the modified $X(n, S)$-representation algorithm works correctly. Input the number m of terms of the X_n-representation ${A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$, the exponents u_1, u_2, u_3, $\cdots$,u$_{m-1}$ and u_m. Our implementation provides additional intermediate outputs that verify the checks $e_i \notin S$ and $e_i \in S$ in Step 1. In Step 1 of each $j = 1, \cdots, t$, the program outputs ${A_n}^{-e_i}B_n{A_n}^{e_i}$ if $e_i = s_j$ and ϵ if $e_i \notin s_j$. The total number of ${A_n}^{-s_i}B_n{A_n}^{s_i}$ and ϵ output by the program is $t\frac{m-1}{2}$ where t is the total number of elements of S and m is the number of terms of the input X_n-representation of M. After execution of the program, collect the subwords that appear as outputs of the program and concatenate them in order. Next, we check whether the number of the terms of the output $X(n, S)$-representation of the program is $\frac{m-1}{2}$. If the number of terms of the output $X(n, S)$-representation of M is $\frac{m-1}{2}$, then $M \in G(n, S)$ and it is the correct $X(n, S)$-representation of M. If the number of terms of the $X(n, S)$-representation of M is not $\frac{m-1}{2}$, then $M \notin G(n, S)$ and so it is not the $X(n, S)$-representation of M. We implement the following cases according as the types of input X_n-representations.

Example 1

Given $S = \{2, 3\}$ and $X(n, S) = \{{A_4}^{-2}B_4{A_4}^2, {A_4}^{-3}B_4{A_4}^3\}$, the X_4-representation of $M = \begin{pmatrix} 1393 & 17088 \\ -176 & -2159 \end{pmatrix} \in \Gamma_4$ is ${A_4}^{-2}{B_4}^3{A_4}^{-1}B_4{A_4}^3$ which is obtained by the X_n-representation algorithm. Input $m = 5$ and the exponents $u_1 = -2, u_2 = 3, u_3 = -1, u_4 = 1, u_5 = 3$ to the $X(n, S)$-representation program in the following.

```
> su:=proc()
> local i,e,m,s,u;
> m:=5;
> u[1] := -2;
> u[2] := 3;
> u[3] := -1;
> u[4] := 1;
> u[5] := 3;
> s[0] := 0;
> for i from 1 to (m - 1)/2 do
> s[i] := s[i - 1] - u[2 * i - 1];
> if s[i] = 2 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 3 then
> print(A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> end do;
> end proc:
```

The program outputs the following.

```
> su();
```

$$A^{-2}B^3A^2$$

$$\epsilon$$

$$\epsilon$$

$$A^{-3}B^1A^3$$

where $A = A_4 = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}$ and $B = B_4 = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}$. For each iteration i of the program, the program computes each term ${A_n}^{-s}{B_n}^{u_{2i}}{A_n}^s$ where $s \in S$ and nonzero $u_{2i} \in \mathbb{Z}$. Since two elements $s = 2$ and $s = 3$ of S are given in the program, for every iteration i, the program produces two outputs and so the total number of iterations is $\frac{m-1}{2} = \frac{5-1}{2} = 2$. Therefore the total number of outputs of the program is $t\frac{m-1}{2}$ where t is the number of elements of S. For the first iteration $i = 1$ of the program, the program computes $A^{-2}B^3A^2$ and the second output of the first iteration $i = 1$ of the program is ϵ. For the second iteration $i = 2$ of the program, the first output is ϵ and the second output is $A^{-3}B^1A^3$. In order to compute the $X(n, S)$-representation of M, collect the $X(n, S)$-representations appear as outputs of the program and concatenate them in order

$$A^{-2}B^3A^2A^{-3}B^1A^3.$$

This is the $X(n, S)$-representation of M obtained by the $X(n, S)$-representation program. We check whether the number of terms of the $X(n, S)$-representation of M is $\frac{m-1}{2} = 2$ and it is the same as the number 2 of terms of the $X(n, S)$-representation $\underbrace{A^{-2}B^3A^2}\underbrace{A^{-3}B^1A^3}$ (2 terms) of M which is obtained from the program. Therefore it is the correct $X(n, S)$-representation of M and so M is an element of $G(n, S)$.

Example 2

Given $S = \{2, 3, 5\}$ and $X(n, S) = \{{A_4}^{-2}B_4{A_4}^2, {A_4}^{-3}B_4{A_4}^3, {A_4}^{-5}B_4{A_4}^5\}$, the

X_4-representation of $M = \begin{pmatrix} -141951 & -1666816 \\ 17932 & 210561 \end{pmatrix} \in \Gamma_4$ is ${A_4}^{-2}{B_4}^{3}{A_4}^{-3}{B_4}^{1}$ ${A_4}^{2}{B_4}^{-1}{A_4}^{3}$ which is obtained by the X_n-representation algorithm. Input $m = 7$ and the exponents $u_1 = -2, u_2 = 3, u_3 = -3, u_4 = 1, u_5 = 2, u_6 = -1, u_7 = 3$ to the $X(n, S)$-representation program in the following.

```
> su:=proc()
> local i,e,m,s,u;
> m := 7;
> u[1] := -2;
> u[2] := 3;
> u[3] := -3;
> u[4] := 1;
> u[5] := 2;
> u[6] := -1;
> u[7] = 3;
> s[0] := 0;
> for i from 1 to (m - 1)/2 do
> s[i] := s[i - 1] - u[2 * i - 1];
> if s[i] = 2 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 3 then
> print(A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 5 then
> > print(A^{−s[i]} ∗ B^{u[2 ∗ i]} ∗ A^{s[i]});
> else
> print(epsilon);
> fi;
> end do;
> end proc:
```

The program outputs the following.

```
> su();
```

$$A^{-2}B^{3}A^{2}$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-5}B^{1}A^{5}$$

$$\epsilon$$

$$A^{-3}B^{-1}A^{3}$$

$$\epsilon$$

Collect the $X(n, S)$-representations which appear as outputs of the program and concatenate them in order. So we have

$$A^{-2}B^{3}A^{2}A^{-5}B^{1}A^{5}A^{-3}B^{-1}A^{3}.$$

The number of the terms of the $X(n, S)$-representation $\underbrace{A^{-2}B^3A^2}\underbrace{A^{-5}B^1A^5}$ $\underbrace{A^{-3}B^{-1}A^3}$ (3 terms) which is obtained by the program is 3 and it is the same as the number $\frac{m-1}{2} = \frac{7-1}{2} = 3$ of terms of the $X(n, S)$-representation of M by the formula $\frac{m-1}{2}$ to compute the number of terms of the $X(n, S)$-representation. Therefore the program outputs the correct $X(n, S)$-representation of M and M is an element of $G(n, S)$.

So far we have seen the cases that $M \in \Gamma_n$ is an element of $G(n, S)$. However, the following is an example to show how the $X(n, S)$-representation program works in case that $M \in \Gamma_n$ is not an element of $G(n, S)$.

Example 3

We show how the algorithm works in case that M is not an element of $G(n, S)$. Given $S = \{2, 3\}$ and $X(n, S) = \{{A_4}^{-2}B_4{A_4}^2, {A_4}^{-3}B_4{A_4}^3\}$, the X_4-representation of $M = \begin{pmatrix} -141951 & -1666816 \\ 17932 & 210561 \end{pmatrix} \in \Gamma_4$ is ${A_4}^{-2}{B_4}^3{A_4}^{-3}{B_4}^1{A_4}^2$ ${B_4}^{-1}{A_4}^3$ which is obtained by the X_n-representation algorithm. Input $m = 7$ and the exponents $u_1 = -2, u_2 = 3, u_3 = -3, u_4 = 1, u_5 = 2, u_6 = -1, u_7 = 3$ to the $X(n, S)$-representation program in the following.

```
> su:=proc()
> local i,e,m,s,u;
> m := 7;
> u[1] := −2;
> u[2] := 3;
> u[3] := −3;
> u[4] := 1;
> u[5] := 2;
> u[6] := −1;
> u[7] = 3;
> s[0] := 0;
> for i from 1 to (m − 1)/2 do
> s[i] := s[i − 1] − u[2 ∗ i − 1];
> if s[i] = 2 then
> print (A^{−s[i]} ∗ B^{u[2 ∗ i]} ∗ A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 3 then
> print(A^{−s[i]} ∗ B^{u[2 ∗ i]} ∗ A^{s[i]});
> else
> print(epsilon);
> fi;
> end do;
> end proc:
```

The program outputs the following.

```
> su();
```

$$A^{-2}B^{3}A^{2}$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-3}B^{-1}A^{3}$$

Collect the $X(n, S)$-representations which appear as outputs of the program and concatenate them in order. So we we have

$$A^{-2}B^3A^2A^{-3}B^{-1}A^3.$$

The number of the terms of $\underbrace{A^{-2}B^3A^2}\underbrace{A^{-3}B^{-1}A^3}$ is 2 and it is not the same as the number $\frac{m-1}{2} = \frac{7-1}{2} = 3$ of terms of the $X(n, S)$-representation of M by the formula $\frac{m-1}{2}$ where m is the number of terms of the X_n-representation of M. It means that M is not an element of $G(n, S)$ and so it is not the $X(n, S)$-representation of M.

Example 4

This implementation shows how the $X(n, S)$-representation program works in case that the type of the X_n-representation of M is ${B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ (even m). Given $S = \{0, 2, 3\}$ and $X(n, S) = \{{A_4}^0 B_4 {A_4}^0, {A_4}^{-2} B_4 {A_4}^2, {A_4}^{-3} B_4$ ${A_4}^3\}$, the X_4-representation of $M = \begin{pmatrix} -1583 & -18624 \\ -6132 & -72143 \end{pmatrix} \in \Gamma_4$ is $B_4 {A_4}^{-2} {B_4}^3 {A_4}^{-1}$ ${B_4}^{-1} {A_4}^3$ which is obtained by the X_n-representation algorithm. However, we consider the X_4-representation ${A_4}^0 B_4 {A_4}^{-2} {B_4}^3 {A_4}^{-1} {B_4}^{-1} {A_4}^3$ instead of the X_n-representation $B_4 {A_4}^{-2} {B_4}^3 {A_4}^{-1} {B_4}^{-1} {A_4}^3$ to compute the $X(n, S)$-representation of M. Input $m = 7$ and the exponents $u_1 = 0, u_2 = 1, u_3 = -2, u_4 = 3, u_5 = -1, u_6 = -1, u_7 = 3$ to the $X(n, S)$-representation program in the following.

```
> su:=proc()
> local i,e,m,s,u;
> m := 7;
> u[1] := 0;
> u[2] := 1;
> u[3] := -2;
> u[4] := 3;
> u[5] := -1;
> u[6] := -1;
> u[7] := 3;
> s[0] := 0;
> for i from 1 to (m - 1)/2 do
> s[i] := s[i - 1] - u[2 * i - 1];
> if s[i] = 0 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 2 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 3 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> end do;
> end proc:
```

The program outputs the following.

```
> su();
```

$$(A^0)^2B^1$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-2}B^3A^2$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-3}B^{-1}A^3.$$

Maple version 6 presents $A^0B^1A^0$ as $(A^0)^2B^1$. Collect the $X(n,S)$-representations appear as outputs of the program and concatenate them in order as follows :

$$A^0B^1A^0A^{-2}B^3A^2A^{-3}B^{-1}A^3.$$

The number of terms of $\underbrace{A^0B^1A^0}\underbrace{A^{-2}B^3A^2}\underbrace{A^{-3}B^{-1}A^3}$ is 3 and it is the same as $\frac{m-1}{2} = \frac{7-1}{2} = 3$ by the formula $\frac{m-1}{2}$ to compute the number of terms of the $X(n,S)$-representation of M where m is the number of terms of the X_n-representation of $M \in G(n,S)$. Therefore M is an element of $G(n,S)$ and $A^0B^1A^0A^{-2}B^3A^2A^{-3}B^{-1}A^3$ is the $X(n,S)$-representation of M.

Example 5

We show how the (n,S)-representation program works in case that the type of the X_n-representation of M is ${A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ (even m). Given $S = \{0,2,3\}$ and $X(n,S) = \{{A_4}^0B_4{A_4}^0, {A_4}^{-2}B_4{A_4}^2, {A_4}^{-3}B_4{A_4}^3\}$, the X_4-representation of $M = \begin{pmatrix} -76079 & -18624 \\ 9612 & 2353 \end{pmatrix} \in \Gamma_4$ is ${A_4}^{-2}{B_4}^3{A_4}^{-1}{B_4}^{-1}{A_4}^3B_4$

which is obtained by the X_n-representation algorithm. However, we consider the X_4-representation ${A_4}^{-2}{B_4}^3{A_4}^{-1}{B_4}^{-1}{A_4}^3B_4{A_4}^0$ instead of the X_n-representation ${A_4}^{-2}{B_4}^3{A_4}^{-1}{B_4}^{-1}{A_4}^3B_4$ to compute the $X(n,S)$-representation of M. Input $m = 7$ and the exponents $u_1 = -2, u_2 = 3, u_3 = -1, u_4 = -1, u_5 = 3, u_6 = 1, u_7 = 0$ to the $X(n,S)$-representation program in the following.

```
> su:=proc()
> local i,e,m,s,u;
> m := 7;
> u[1] := -2;
> u[2] := 3;
> u[3] := -1;
> u[4] := -1;
> u[5] := 3;
> u[6] := 1;
> u[7] := 0;
> s[0] := 0;
> for i from 1 to (m - 1)/2 do
> s[i] := s[i - 1] - u[2 * i - 1];
> if s[i] = 0 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 2 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 3 then
> print (A^{−s[i]} ∗ B^{u[2 ∗ i]} ∗ A^{s[i]});
> else
> print(epsilon);
> fi;
> end do;
> end proc:
```

The program outputs the following

```
> su();
```

$$\epsilon$$

$$A^{-2}B^{3}A^{2}$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-3}B^{-1}A^{3}.$$

$$(A^{0})^{2}B^{1}$$

$$\epsilon$$

$$\epsilon$$

Collect the $X(n, S)$-representations which appear as outputs of the program and concatenate them in order as follows :

$$A^{-2}B^3A^2A^{-3}B^{-1}A^3A^0B^1A^0.$$

The number of terms of $\underbrace{A^{-2}B^3A^2}\underbrace{A^{-3}B^{-1}A^3}\underbrace{A^0B^1A^0}$ is 3 and it is the same as $\frac{m-1}{2} = \frac{7-1}{2} = 3$ by the formula $\frac{m-1}{2}$ to compute the number of terms of the $X(n,S)$-representation of M where m is the number of terms of the X_n-representation of $M \in G(n,S)$. Hence M is an element of $G(n,S)$ and the program outputs the correct $X(n,S)$-representation of M.

Example 6

This implementation shows how the $X(n,S)$-representation program works in case that the type of the X_n-representation of M is ${B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ (odd m). Given $S=\{0,2,3\}$ and $X(n,S)=\{{A_4}^0B_4{A_4}^0, {A_4}^{-2}B_4{A_4}^2, {A_4}^{-3}B_4{A_4}^3\}$, the X_4-representation of $M = \begin{pmatrix} -76079 & -18624 \\ -294704 & -72143 \end{pmatrix} \in \Gamma_4$ is $B_4{A_4}^{-2}{B_4}^3{A_4}^{-1}$ ${B_4}^{-1}{A_4}^3B_4$ which is obtained by the X_n-representation algorithm. However, we consider the X_4-representation ${A_4}^0B_4{A_4}^{-2}{B_4}^3{A_4}^{-1}{B_4}^{-1}{A_4}^3B_4{A_4}^0$ instead of the X_n-representation $B_4{A_4}^{-2}{B_4}^3{A_4}^{-1}{B_4}^{-1}{A_4}^3B_4$ to compute the $X(n,S)$-representation of M. Input $m=9$ and the exponents $u_1=0, u_2=1, u_3=-2, u_4=3, u_5=-1, u_6=-1, u_7=3, u_8=1, u_9=0$ to the $X(n,S)$-representation program in the following.

```
> su:=proc()
> local m,i,u,s,e;
> m := 9;
> u[1] := 0;
> u[2] := 1;
> u[3] := −2;
> u[4] := 3;
> u[5] := −1;
> u[6] := −1;
> u[7] := 3;
> u[8] := 1;
> u[9] := 0;
> s[0] := 0;
> for i from 1 to (m - 1)/2 do
> s[i] := s[i - 1] - u[2 * i - 1];
> if s[i] = 0 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 2 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> if s[i] = 3 then
> print (A^{-s[i]} * B^{u[2 * i]} * A^{s[i]});
> else
> print(epsilon);
> fi;
> end do;
> end proc:
```

The program outputs the following.

```
> su();
```

$$(A^0)^2B^1$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-2}B^3A^2$$

$$\epsilon$$

$$\epsilon$$

$$\epsilon$$

$$A^{-3}B^{-1}A^3.$$

$$\left(A^0\right)^2B^1$$

$$\epsilon$$

$$\epsilon$$

Collect the $X(n,S)$-representations which appear as outputs of the program and concatenate them in order as follows :

$$A^0B^1A^0A^{-2}B^3A^2A^{-3}B^{-1}A^3A^0B^1A^0.$$

The number of terms of $\underbrace{A^0B^1A^0}\underbrace{A^{-2}B^3A^2}\underbrace{A^{-3}B^{-1}A^3}\underbrace{A^0B^1A^0}$ is 4 and it is the same as $\frac{m-1}{2}=\frac{9-1}{2}=4$ by the formula $\frac{m-1}{2}$ to compute the number of terms of the $X(n,S)$-representation of M where m is the number of the X_n-representation of $M \in G(n,S)$. Therefore M is an element of $G(n,S)$ and the program outputs the correct $X(n,S)$-representation of M.

5.5 Correctness of Modified $X(n, S)$-Representation Algorithm

In this section, we first consider several X_n-representations ${A_n}^u$, ${B_n}^u$, ${A_n}^{u_1}{B_n}^{u_2}$ and ${B_n}^{u_1}{A_n}^{u_2}$ and we prove that they are not elements of $G(n, S)$ and so the X_n-representations ${A_n}^u$, ${B_n}^u$, ${A_n}^{u_1}{B_n}^{u_2}$ and ${B_n}^{u_1}{A_n}^{u_2}$ are not applied to the modified $X(n, S)$-representation algorithm. Moreover, we justify the modified $X(n, S)$-representation algorithm.

Theorem 5.5.1 If $M = {A_n}^u$ with a nonzero integer u, then $M \notin G(n, S)$.

Proof If $M = {A_n}^u$ with a nonzero integer u, then by the X_n-representation algorithm, we obtain ${A_n}^u$ as the X_n-representation of M and the X_n-representation ${A_n}^u$ can be written as

$$ {A_n}^u = {A_n}^u {B_n}^0 {A_n}^0. $$

Since the $X(n, S)$-representation is the form ${A_n}^{-s}{B_n}^v{A_n}^s$ where $s \in S$ and a nonzero integer v, the exponent of B_n does not have to be zero, but in case of ${A_n}^{u_1}{B_n}^0{A_n}^0$, the exponent of B_n is zero. Hence, the form ${A_n}^{u_1}{B_n}^0{A_n}^0$ is not the $X(n, S)$-representation and so M does not have the $X(n, S)$-representation. Hence $M = {A_n}^u$ is not an element of $G(n, S)$. □

Theorem 5.5.2 Let $M = {B_n}^u$ with a nonzero integer u. Then

1. if $0 \in S$, then $M \in G(n, S)$

2. if $0 \notin S$, then $M \notin G(n, S)$.

Proof If $M = {B_n}^u$ with a nonzero integer u, then by the X_n-representation algorithm, we obtain ${B_n}^u$ as the X_n-representation of M and the X_n-representation ${B_n}^u$ can be written as

$$B_n{}^u = A_n{}^0 B_n{}^u A_n{}^0$$

where u is a nonzero integer. If $0 \in S$, then $A_n{}^0 B_n{}^u A_n{}^0$ is the form $A_n{}^{-s} B_n{}^v A_n{}^s$ where $s \in S$ and a nonzero integer v and thus $M = B_n{}^u$ is an element of $G(n, S)$. If $0 \notin S$, then it does not satisfy the condition that the exponent of the third term of $A_n{}^0 B_n{}^u A_n{}^0$ has to belong to S and so $M = B_n{}^u$ is not an element of $G(n, S)$. □

Theorem 5.5.3 If $M = A_n{}^{u_1} B_n{}^{u_2}$ with nonzero integers u_1 and u_2, then $M \notin G(n, S)$.

Proof Let $M = A_n{}^{u_1} B_n{}^{u_2}$ with nonzero integers u_1 and u_2. Then the X_n-representation algorithm computes $A_n{}^{u_1} B_n{}^{u_2}$ of M as the X_n-representation of M and it can be written as

$$A_n{}^{u_1} B_n{}^{u_2} A_n{}^0$$

where u_1 and u_2 are nonzero integers. Since $u_1 \neq 0$, $A_n{}^{u_1} B_n{}^{u_2} A_n{}^0$ is not the form $A_n{}^{-s} B_n{}^v A_n{}^s$ where $s \in S$ and v is nonzero integer. Hence M does not have the $X(n, S)$-representation and so $M = A_n{}^{u_1} B_n{}^{u_2}$ is not an element of $G(n, S)$. □

Theorem 5.5.4 If $M = B_n{}^{u_1} A_n{}^{u_2}$ with nonzero integers u_1 and u_2, then $M \notin G(n, S)$.

Proof If $M = B_n{}^{u_1} A_n{}^{u_2} \in \Gamma_n$ with nonzero integers u_1 and u_2, then the X_n-representation algorithm computes $B_n{}^{u_1} A_n{}^{u_2}$ as the X_n-representation of M and it can be written as

$$A_n{}^0 B_n{}^{u_1} A_n{}^{u_2}$$

where u_1 and u_2 are nonzero integers. Since $u_2 \neq 0$, $A_n{}^0 B_n{}^{u_1} A_n{}^{u_2}$ is not the form $A_n{}^{-s} B_n{}^v A_n{}^s$ where $s \in S$ and v is a nonzero integer. Hence M does not have the $X(n,S)$-representation and so M is not an element of $G(n,S)$. □

Theorem 5.5.5 If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ with each nonzero integer u_i $(i = 2,3,\cdots,m-1)$, then the modified $X(n,S)$-representation algorithm outputs

$$ {A_n}^{u_1}{B_n}^{u_2}{A_n}^{-u_1}{A_n}^{u_1+u_3}{B_n}^{u_4}{A_n}^{-(u_1+u_3)}\cdots{A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{-(u_1+u_3+u_5+\cdots+u_{m-2})} $$

as the $X(n,S)$-representation where $u_2,\cdots u_{m-1}$ are nonzero integers. Otherwise it outputs ϵ.

Proof If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ with each nonzero integer u_i $(i = 2,3,\cdots,m-1)$, the X_n-representation algorithm computes ${A_n}^{u_1}{B_n}^{u_2}\cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m}$ as the X_n-representation of M. So in Step 0 of the first iteration $i=1$, $w = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$.

In Step 1 of the first iteration, $e_1 = -u_1$.

If $e_1 \notin S$, then the modified $X(n,S)$-representation algorithm outputs ϵ and the algorithm terminates.

If $e_1 = -u_1 \in S$, then $C_1 = {A_n}^{-e_1}{B_n}^{u_2}{A_n}^{e_1} = {A_n}^{u_1}{B_n}^{-u_2}{A_n}^{-u_1}$.

In Step 2 of the first iteration,

$$\begin{aligned} w &= {C_1}^{-1}w \\ &= {A_n}^{u_1}{B_n}^{-u_2}{A_n}^{-u_1}{A_n}^{u_1}{B_n}^{u_2}{A_n}^{u_3}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \\ &= {A_n}^{u_1+u_3}{B_n}^{u_4}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}. \end{aligned}$$

As $w \neq 1_{X(n,S)}$, $i=2$ and return Step 1.

Assume that for $1 \le j-1 \le \frac{m-5}{2}$, in Step 2 of the $i = j-1$th iteration, $w = {A_n}^{(u_1+u_3+u_5+\cdots+u_{2j-3}+u_{2j-1})}{B_n}^{u_{2j}}{A_n}^{u_{2j+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$.

In Step 1 of the $i=j$th iteration, $e_j = -(u_1+u_3+u_5+\cdots+u_{2j-1})$.

If $e_j = -(u_1+u_3+u_5+\cdots+u_{2j-1}) \notin S$, then the algorithm outputs ϵ and it terminates.

If $e_j = -(u_1 + u_3 + u_5 + \cdots + u_{2j-1}) \in S$, then

$$\begin{aligned} C_j &= {A_n}^{-e_j} {B_n}^{u_{2j}} {A_n}^{e_j} \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{2j-1}} {B_n}^{u_{2j}} {A_n}^{-(u_1+u_3+u_5+\cdots+u_{2j-1})}. \end{aligned}$$

In Step 2 of the $i = j$th iteration,

$$\begin{aligned} w &= {C_j}^{-1} w \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{2j-1}} {B_n}^{-u_{2j}} {A_n}^{-(u_1+u_3+u_5+\cdots+u_{2j-1})} {A_n}^{u_1+u_3+u_5+\cdots+u_{2j-3}+u_{2j-1}} \\ &\quad {B_n}^{u_{2j}} {A_n}^{u_{2j+1}} {B_n}^{u_{2j+2}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{2j-1}+u_{2j+1}} {B_n}^{u_{2j+2}} {A_n}^{u_{2j+3}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} \end{aligned}$$

As $w \neq 1_{X(n,S)}$, $i = j + 1$ and return Step 1.

If $j = \frac{m-1}{2}$, then in Step 1, $e_{\frac{m-1}{2}} = -(u_1 + u_3 + u_5 + \cdots + u_{m-2})$.

If $e_{\frac{m-1}{2}} \notin S$, then the algorithm outputs ϵ and it terminates.

If $e_{\frac{m-1}{2}} \in S$, then

$$\begin{aligned} C_{\frac{m-1}{2}} &= {A_n}^{-e_{\frac{m-1}{2}}} {B_n}^{u_{m-1}} {A_n}^{e_{\frac{m-1}{2}}} \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}} {B_n}^{u_{m-1}} {A_n}^{-(u_1+u_3+u_5+\cdots+u_{m-2})}. \end{aligned}$$

In Step 2 of the $j = \frac{m-1}{2}$th iteration,

$$\begin{aligned} w &= {C_{\frac{m-1}{2}}}^{-1} w \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}} {B_n}^{-u_{m-1}} {A_n}^{-(u_1+u_3+u_5+\cdots+u_{m-2})} w \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}} {B_n}^{-u_{m-1}} {A_n}^{-(u_1+u_3+u_5+\cdots+u_{m-2})} {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}} {B_n}^{u_{m-1}} {A_n}^{u_m} \\ &= {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}+u_m} \end{aligned}$$

If $u_m = -(u_1 + u_3 + u_5 + \cdots + u_{m-2})$, then $w = 1_{X(n,S)}$ and the algorithm outputs

$C_1 C_2 C_3 \cdots C_{\frac{m-3}{2}} C_{\frac{m-1}{2}}$

$= {A_n}^{u_1} {B_n}^{u_2} {A_n}^{-u_1} {A_n}^{u_1+u_3} {B_n}^{u_4} {A_n}^{-(u_1+u_3)} \cdots {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}} {B_n}^{u_{m-1}} {A_n}^{-(u_1+u_3+u_5+\cdots+u_{m-2})}$

as the $X(n, S)$-representation of M.

If $u_m \neq -(u_1+u_3+u_5+\cdots+u_{m-2})$, then $w = {A_n}^{u_1+u_3+u_5+\cdots+u_{m-2}+u_m} \neq 1_{X(n,S)}$ and $i = \frac{m+1}{2}$. Therefore, the algorithm outputs ϵ and it terminates. □

Chapter 6

X_1-Representation Algorithms

Let Γ_1 be the group generated by two matrices A_1 and B_1 where

$$A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ and } B_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

and let $X_1 = \{A_1, B_1\}$. Then for a fixed $n \geq 2$, two free generating elements A_n and B_n of Γ_n can be expressed by two matrices A_1 and B_1 as follows :

$$A_n = \begin{pmatrix} 1 & n \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^n = {A_1}^n$$

and

$$B_n = \begin{pmatrix} 1 & 0 \\ n & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^n = {B_1}^n,$$

and

$${A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \text{ gives rise to } {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{nu_m}.$$
$${B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \text{ gives rise to } {B_1}^{nu_1}{A_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{nu_m}.$$
$${A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \text{ gives rise to } {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-1}}{B_1}^{nu_m}.$$
$${B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \text{ gives rise to } {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{m-1}}{B_1}^{nu_m}.$$

In this chapter, we call them the X_1-representations, but genuinely the X_1-representations are the following types

$$
\begin{gathered}
{A_1}^{e_1}{B_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m} \text{ (odd } m) \\
{B_1}^{e_1}{A_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m} \text{ (even } m) \\
{A_1}^{e_1}{B_1}^{e_2}\cdots{A_1}^{e_{m-1}}{B_1}^{e_m} \text{ (even } m) \\
{B_1}^{e_1}{A_1}^{e_2}\cdots{A_1}^{e_{m-1}}{B_1}^{e_m} \text{ (odd } m).
\end{gathered}
$$

where each e_i is a nonzero integer $(i = 1, 2, \cdots, m)$.

In Chapter 4, the X_n-representation algorithm computes the X_n-representation of an element M of Γ_n assuming the natural number $n \geq 2$ is known. So given the X_n-representation of $M \in \Gamma_n$, we can compute the X_1-representation of M. However, we need algorithms not requiring knowledge of n to break Grigoriev and Ponomarenko homomorphic public-key cryptosystem in Chapter 8. Thus we design new algorithms called the X_1-representation algorithm I and II. The X_1-representation algorithm I is for $M \in \Gamma_n$ where $n \geq 2$ is an even natural number and the X_1-representation algorithm II is for $M \in \Gamma_n$ where $n \geq 3$. We will see the behavior of the linear fractional transformations ${A_n}^u$ and ${B_n}^u$ is different in these two cases and so two algorithms are required. Because n is unknown, if $M \in \Gamma_n$ is input to the X_1-representation algorithm, then the X_1-representation algorithm outputs one of the following

$$
\begin{gathered}
{A_1}^{e_1}{B_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m} \text{ (odd } m) \\
{B_1}^{e_1}{A_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m} \text{ (even } m) \\
{A_1}^{e_1}{B_1}^{e_2}\cdots{A_1}^{e_{m-1}}{B_1}^{e_m} \text{ (even } m) \\
{B_1}^{e_1}{A_1}^{e_2}\cdots{A_1}^{e_{m-1}}{B_1}^{e_m} \text{ (odd } m).
\end{gathered}
$$

where each e_i is a nonzero integer such that $e_i = nu_i$ with a nonzero u_i $(i = 1, 2, \cdots, m)$. So now we describe the structure of the chapter.

In Section 6.1, we present the X_1-representation algorithm I. In Section 6.2, We implement the X_1-representation algorithm I by programming it and demonstrate it. In Section 6.3, we prove the correctness of the X_1-representation algorithm I. In Section 6.4, we show the X_1-representation algorithm II. In Section 6.5, we implement the X_1-representation algorithm II by programming it and demonstrate it. In Section 6.6, we prove the correctness of the X_1-representation algorithm II.

6.1 X_1-Representation Algorithm I

Assume that $n \geq 2$ is an unknown even natural number and $M \in \Gamma_n$. Then the X_1-representation algorithm I works for every even natural number $n \geq 2$. We input a matrix $M \in \Gamma_n$ to the X_1-representation algorithm and it outputs one of the four X_1-representation types which are shown before. We use two fixed values $z = \frac{1}{2}$ and $z = 2$ to compute the X_1-representation of M. If the algorithm outputs the X_1-representation of M for $z = \frac{1}{2}$, then we do not run the algorithm for $z = 2$. If not, then we have to run the algorithm for $z = 2$ to compute the X_1-representation of M. I denotes the identity matrix, w is a reduced word in ${X_1}^{\pm}$ and 1_{X_1} is the empty word in W_{X_1}.

X_1-Representation Algorithm I

```
Step 0
$\mathbf{w} \leftarrow \mathbf{1_{X_1}}$
$\mathbf{L} \leftarrow \mathbf{M}$

Step 1
$\mathbf{L(z) = 0, |L(z)| = 1, L(z) = \infty \Rightarrow}$ output $\epsilon$.
$\mathbf{|L(z)| > 1 \Rightarrow}$ go to Step 2
$\mathbf{|L(z)| < 1 \Rightarrow}$ go to Step 3

Step 2
$\mathbf{e} \leftarrow$ even number in $\{\lfloor \mathbf{L(z)} \rfloor, \lceil \mathbf{L(z)} \rceil\}$
$\mathbf{C} \leftarrow \mathbf{{A_1}^{e}}$ and $\mathbf{w} \leftarrow \mathbf{wC}$.
```

```
$\mathbf{C = I \Rightarrow}$ output $\epsilon$.
$\mathbf{L \leftarrow C^{-1}L}$
$\mathbf{L = I \Rightarrow}$ output $\mathbf{w}$. Otherwise, return Step 1.
```

Step 3

```
$\mathbf{e \leftarrow}$ even number in $\{\lfloor\frac{\mathbf{1}}{\mathbf{L(z)}}\rfloor, \lceil\frac{\mathbf{1}}{\mathbf{L(z)}}\rceil\}$
$\mathbf{C \leftarrow {B_1}^e}$ and $\mathbf{w \leftarrow wC}$.
$\mathbf{C = I \Rightarrow}$ output $\epsilon$.
$\mathbf{L \leftarrow C^{-1}L}$
$\mathbf{L = I \Rightarrow}$ output $\mathbf{w}$. Otherwise, return Step 1.
```

6.2 Programming Implementation I

In order to demonstrate how the X_1-representation algorithm I works correctly, we make a program called the X_1-representation program I by Maple version 6 and implement it. The operation of the program is one loop. Input z value and the entries $M11$, $M12$, $M21$ and $M22$ of the matrix $M \in \Gamma_n$ to the X_1-representation program I. Then for every execution of the program, it outputs two matrices. The first matrix presents a matrix C which is $C = {A_1}^e$ in Step 2 or $C = {B_1}^e$ in Step 3 of the X_1-representation algorithm I. The second matrix is $L = C^{-1}L$ in Step 2 or $L = C^{-1}L$ in Step 3 of the X_1-representation algorithm I. When one of two matrices are the identity matrix, execution of the program terminates. If the second matrix is the identity matrix, then collect each first matrix in every execution of the program and concatenate them in order. So we can obtain the X_1-representation of M.

The X_1-Representation Program I Source Code

```
 with(GaussInt):
with(linalg):
su:=proc(z::float,M11::integer,M12::integer,M21::integer,M22::integer)
```

```
local K,C,P,Q;
K:=matrix(2,2,[M11,M12,M21,M22]);
L(z):=(M11 * z + M12)/(M21 * z + M22);
if abs(L(z))=1 then
print(epsilon);
fi;
if abs(L(z))>1 then
if irem(floor(L(z)),2)=0 then
C:=matrix(2, 2, [1, 1, 0, 1])^{floor(L(z))};
P:=matrix(2,2,[1,-floor(L(z)),0,1]);
Q:=multiply(P,K);
print(C);
print(Q);
else
C:=matrix(2, 2, [1, 1, 0, 1])^{ceil(L(z))};
P:=matrix(2,2,[1,-ceil(L(z)),0,1]);
Q:=multiply(P,K);
print(C);
print(Q);
fi;
fi;
if abs(L(z))< 1 then
if irem(floor(1/(L(z))),2)=0 then
C:=matrix(2, 2, [1, 0, 1, 1])^{floor(1/(L(z))};
P:=matrix(2,2,[1,0,-floor(1/(L(z))),1]);
Q:=multiply(P,K);
print(C);
print(Q);
else
```

```
C:=matrix(2,2,[1,0,1,1])^{ceil(1/(L(z)))};

P:=matrix(2,2,[1,0,-ceil(1/(L(z))),1]);

Q:=multiply(P,K);

print(C);

print(Q);

fi;

fi;

end proc:
```

Example 1

Given $M = A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \in \Gamma_2$, input $z = 0.5$, $M11 = 1$, $M12 = 2$, $M21 = 0$ and $M22 = 1$ to the X_1-representation algorithm I.

For $z = \frac{1}{2}$,

```
> su(0.5,1,2,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^2$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-2} \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm I and so execution of the program terminates. Take the first matrix of the first execution

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^2$$

and it is the X_1-representation of M.

For $z = 2$,

> su(2.0,1,2,0,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^4$$

$$\begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}$$

> su(2.0,1,-2,0,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^\infty$$

$$\begin{pmatrix}1 & -2\\ -\infty & \infty\end{pmatrix}$$

The first matrix of the second execution of the program is an usual matrix $\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^\infty$ and it is the same as ϵ which the X_1-representation algorithm I outputs in Step 2 because $L(2) = {A_2}^{-1}(2) = 0$ in Step 2. Hence execution of the program terminates and the program does not output the X_1-representation of M for $z = 2$. □

Example 2

Given $M = A_4 = \begin{pmatrix}1 & 4\\ 0 & 1\end{pmatrix} \in \Gamma_4$, input $z = 0.5$, $M11 = 1$, $M12 = 4$, $M21 = 0$ and $M22 = 1$ to the program.

$z = \frac{1}{2}$,

> su(0.5,1,4,0,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^4$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-4}\begin{pmatrix}1 & 4\\0 & 1\end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm I. Take the first matrix of the first execution

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{4}$$

and it is the X_1-representation of M.

$z = 2$,

```
> su(2.0,1,4,0,1);
```

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{6}$$

$$\begin{pmatrix}1 & -2\\0 & 1\end{pmatrix}$$

```
> su(2.0,1,-2,0,1);
```

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}1 & -2\\-\infty & \infty\end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 2 of the X_1-representation algorithm I because $L(2) = {A_2}^{-1}(2) = 0$ in Step 2. So execution of the program terminates and the program does not output the X_1-representation of M for $z = 2$. □

Example 3

Given $M = A_6 = \begin{pmatrix}1 & 6\\0 & 1\end{pmatrix} \in \Gamma_6$, input $z = 0.5$, $M11 = 1$, $M12 = 6$, $M21 = 0$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,1,6,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{6}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-6} \begin{pmatrix} 1 & 6 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm I and so execution of the program terminates. Take the first matrix of the first execution of the program

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{6}$$

and this is the X_1-representation of M.

For $z = 2$,

```
> su(2.0,1,6,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{8}$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,-2,0,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} 1 & -2 \\ -\infty & \infty \end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 2 of the X_1-representation algorithm I because $L(2) = {A_2}^{-1}(2) = 0$ in Step 2. So the program does not output the X_1-representation of M. □

Example 4

Given $M = B_2 = \begin{pmatrix}1 & 0\\2 & 1\end{pmatrix} \in \Gamma_2$, input $z = 0.5$, $M11 = 1$, $M12 = 0$, $M21 = 2$ and $M22 = 1$ to the X_1-representation program.

For $z = \frac{1}{2}$,

> su(0.5,1,0,2,1);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{4}$$

$$\begin{pmatrix}1 & 0\\-2 & 1\end{pmatrix}$$

> su(0.5,1,0,-2,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -\infty\\-2 & 1\end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 3 of the X_1-representation algorithm I because $L(\frac{1}{2}) = \begin{pmatrix}1 & 0\\-2 & 1\end{pmatrix}(\frac{1}{2}) = {B_2}^{-1}(\frac{1}{2}) = \frac{\frac{1}{2}}{-1+1} = \infty$ in Step 3. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

> su(2.0,1,0,2,1);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^2$$

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

The second matrix of the first execution is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-2}\begin{pmatrix}1 & 0\\2 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm I. So execution of the program terminates and take the first matrix of the first execution

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^2$$

and this is the X_1-representation of M. □

Example 5

Given $M = B_4 = \begin{pmatrix}1 & 0\\4 & 1\end{pmatrix} \in \Gamma_4$, input $z = 0.5$, $M11 = 1$, $M12 = 0$, $M21 = 4$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,1,0,4,1);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^6$$

$$\begin{pmatrix}1 & 0\\-2 & 1\end{pmatrix}$$

> su(0.5,1,0,-2,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^\infty$$

$$\begin{pmatrix} \infty & -\infty \\ -2 & 1 \end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 3 of the X_1-representation algorithm I because $L(\frac{1}{2}) = {B_2}^{-1}(\frac{1}{2}) = \frac{\frac{1}{2}}{-1+1} = \infty$ in Step 3. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

> su(2.0,1,0,4,1);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{4}$$

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

The second matrix of the first execution is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-4} \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm I and execution of the program terminates. Take the first matrix of the first execution

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{4}$$

which is $C = {B_1}^{e} = {B_1}^{4}$ in Step 3 and this is the X_1-representation of M. □

Example 6

Given $M = B_6 = \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix} \in \Gamma_6$, input $z = 0.5$, $M11 = 1$, $M12 = 0$, $M21 = 6$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,1,0,6,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{8}$$

$$\begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}$$

> su(0.5,1,0,-2,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -\infty\\ -2 & 1\end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 3 of the X_1-representation algorithm I because $L(\frac{1}{2}) = \begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}(\frac{1}{2}) = \infty$ in Step 3. So execution of the program terminates and the program does not output the X_1-representation of M for $z = \frac{1}{2}$.

For $z = 2$,

> su(2.0,1,0,6,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{6}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-6}\begin{pmatrix}1 & 0\\ 6 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm. So execution of the program terminates and take the first matrix of the first execution

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{6}.$$

This is the X_1-representation of M. □

Example 7

Given $M = {A_6}^{-3}{B_6}^{-1}{A_6}^{3} = \begin{pmatrix} 109 & 1944 \\ -6 & -107 \end{pmatrix} \in \Gamma_6$, input $z = 0.5$, $M11 = 109$, $M12 = 1944$, $M21 = -6$ and $M22 = -107$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,109,1944,-6,-107);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18}$$

$$\begin{pmatrix} 1 & 18 \\ -6 & -107 \end{pmatrix}$$

```
> su(0.5,1,18,-6,-107);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6}$$

$$\begin{pmatrix} 1 & 18 \\ 0 & 1 \end{pmatrix}$$

```
>su(0.5,1,18,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{18}$$

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

The second matrix of the third execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18} \begin{pmatrix} 1 & 18 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm I. So execution of the program terminates and collect each first matrix in every execution. Then we have

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{18}.$$

as the X_1-representation of M.

For $z = 2$,

```
>su(2.0,109,1944,-6,-107);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18}$$

$$\begin{pmatrix} 1 & 18 \\ -6 & -107 \end{pmatrix}$$

```
> su(2.0,1,18,-6,-107);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6}$$

$$\begin{pmatrix} 1 & 18 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,18,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{20}$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,-2,0,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} 1 & -2 \\ -\infty & \infty \end{pmatrix}$$

The first matrix of the fourth execution of the program is an unusual matrix $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 2 of the X_1-representation algorithm I because $L(2) = C^{-1}L = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}(2) = 0$ in Step 2. So execution of the program terminates and the program does not output the X_1-representation of M for $z = 2$. □

Example 8

Given $M = {A_6}^{-3}{B_6}^{-1}{A_6}^{3}B_6 = \begin{pmatrix} 11773 & 1944 \\ -648 & -107 \end{pmatrix} \in \Gamma_6$, input $z = 0.5$, $M11 = 11773$, $M12 = 1944$, $M21 = -648$ and $M22 = -107$ to the program.

```
> su(0.5,11773,1944,-648,-107);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18}$$

$$\begin{pmatrix} 109 & 18 \\ -648 & -107 \end{pmatrix}$$

```
> su(0.5,109,18,-648,-107);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6}$$

$$\begin{pmatrix} 109 & 18 \\ 6 & 1 \end{pmatrix}$$

```
> su(0.5,109,18,6,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{18}$$

$$\begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}$$

```
> su(0.5,1,0,6,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{8}$$

$$\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$$

```
> su(0.5,1,0,-2,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} \infty & -\infty \\ -2 & 1 \end{pmatrix}$$

The first matrix of the fifth execution of the program is an unusual matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 3 of the X_1-representation algorithm I because $L(\frac{1}{2}) = C^{-1}L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} (\frac{1}{2}) = \infty$ in Step 3. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

```
>su(2.0,11773,1944,-648,-107);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18}$$

$$\begin{pmatrix} 109 & 18 \\ -648 & -107 \end{pmatrix}$$

```
> su(2.0,109,18,-648,-107);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6}$$

$$\begin{pmatrix} 109 & 18 \\ 6 & 1 \end{pmatrix}$$

```
> su(2.0,109,18,6,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{18}$$

$$\begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix}$$

```
> su(2.0,1,0,6,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{6}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6} \begin{pmatrix} 1 & 0 \\ 6 & 1 \end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm I and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-6} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{18} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{6}.$$

as the X_1-representation of M. □

Example 9

Given $M = {B_6}^2 {A_6}^{-3} {B_6}^{-1} {A_6}^3 = \begin{pmatrix} 109 & 1944 \\ 1302 & 23221 \end{pmatrix} \in \Gamma_6$, input $z = 0.5$, $M11 = 109$, $M12 = 1944$, $M21 = 1302$ and $M22 = 23221$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,109,1944,1302,23221);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{12}$$

$$\begin{pmatrix} 109 & 1944 \\ -6 & -107 \end{pmatrix}$$

> su(0.5,109,1944,-6,-107);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-18}$$

$$\begin{pmatrix} 1 & 18 \\ -6 & -107 \end{pmatrix}$$

> su(0.5,1,18,-6,-107);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-6}$$

$$\begin{pmatrix}1 & 18\\0 & 1\end{pmatrix}$$

> su(0.5,1,18,0,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{18}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-18}\begin{pmatrix}1 & 18\\0 & 1\end{pmatrix} = I$ in Step 2 of the X_1-representation of the algorithm I and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{12}\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-18}\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-6}\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{18}$$

as the X_1-representation of M.

For $z = 2$,

> su(2.0,109,1944,1302,23221);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{12}$$

$$\begin{pmatrix}109 & 1944\\-6 & -107\end{pmatrix}$$

> su(2.0,109,1944,-6,-107);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-18}$$

$$\begin{pmatrix}1 & 18\\-6 & -107\end{pmatrix}$$

> su(2.0,1,18,-6,-107);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-6}$$

$$\begin{pmatrix}1 & 18\\ 0 & 1\end{pmatrix}$$

> su(2.0,1,18,0,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{20}$$

$$\begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}$$

> su(2.0,1,-2,0,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}1 & -2\\ -\infty & \infty\end{pmatrix}$$

The first matrix of the fifth execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 2 of the X_1-representation algorithm I because $L(2) = C^{-1}L(2) = \begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}(2) = 0$ in Step 2. So execution of the program terminates and the program does not output the X_1-representation of M for $z = 2$. $\square$

Example 10

Given $M = {B_6}^2{A_6}^{-3}{B_6}^{-1}{A_6}^3B_6 = \begin{pmatrix}11773 & 1944\\ 140628 & 23221\end{pmatrix} \in \Gamma_6$, input $z = \frac{1}{2}$, $M11 = 11773$, $M12 = 1944$, $M21 = 140628$ and $M22 = 23221$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,11773,1944,140628,23221);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{12}$$

$$\begin{pmatrix}11773 & 1944\\ -648 & -107\end{pmatrix}$$

```
> su(0.5,11773,1944,-648,-107);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-18}$$

$$\begin{pmatrix}109 & 18\\ -648 & -107\end{pmatrix}$$

```
> su(0.5,109,18,-648,-107);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-6}$$

$$\begin{pmatrix}109 & 18\\ 6 & 1\end{pmatrix}$$

```
> su(0.5,109,18,6,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{18}$$

$$\begin{pmatrix}1 & 0\\ 6 & 1\end{pmatrix}$$

```
> su(0.5,1,0,6,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{8}$$

$$\begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}$$

```
> su(0.5,1,0,-2,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -\infty\\ -2 & 1\end{pmatrix}$$

The first matrix of the sixth execution of the program is an unusual matrix $\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 3 of the X_1-representation algorithm I because $L(\frac{1}{2}) = C^{-1}L(\frac{1}{2}) = \begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}(\frac{1}{2}) = \infty$ in Step 3. So execution of the program terminates and the program does not output the X_1-representation of M for $z = \frac{1}{2}$.

For $z = 2$,

```
>su(2.0,11773,1944,140628,23221);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{12}$$

$$\begin{pmatrix}11773 & 1944\\ -648 & -107\end{pmatrix}$$

```
> su(2.0,11773,1944,-648,-107);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-18}$$

$$\begin{pmatrix}109 & 18\\ -648 & -107\end{pmatrix}$$

```
> su(2.0,109,18,-648,-107);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-6}$$

$$\begin{pmatrix}109 & 18\\ 6 & 1\end{pmatrix}$$

```
> su(2.0,109,18,6,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{18}$$

$$\begin{pmatrix}1 & 0\\ 6 & 1\end{pmatrix}$$

```
> su(2.0,1,0,6,1);
```

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{6}$$

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

The second matrix of the fifth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-6}\begin{pmatrix}1 & 0\\6 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm I and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{12}\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-18}\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-6}\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{18}\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{6}.$$

as the X_1-representation of M. □

6.3 Correctness of Algorithm I

In this section, we justify the X_1-representation algorithm I and so we show how the X_1-representation algorithm I works correctly for $z = \frac{1}{2}$ and $z = 2$, respectively. In this chapter, $n(\geq 2)$ indicates an even natural number and note the fact that the natural number n is unknown.

Theorem 6.3.1 If $M = {A_n}^u$ with a nonzero u is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${A_1}^{nu}$ as the X_1-representation of M.

Proof If $M = {A_n}^u \in \Gamma_n$, then by Lemma 4.1.1, in Step 1 of the first iteration, $|L(\frac{1}{2})| = |{A_n}^u(\frac{1}{2})| = |nu + \frac{1}{2}| > 1$. $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu + \frac{1}{2}\rfloor = nu$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu + \frac{1}{2}\rceil = nu + 1$. As n is even, $\lfloor L(\frac{1}{2})\rfloor = nu$ is even. In Step 2, as n is even, $e = nu$, $C = {A_1}^{nu}$, $w = wC = {A_1}^{nu}$ and $L = C^{-1}L = {A_1}^{-nu}{A_n}^u = I$. So the algorithm outputs ${A_1}^{nu}$ as the X_1-representation of M and it terminates. □

Theorem 6.3.2 If $M = {A_n}^u$ with a nonzero integer u is input to the algorithm ($z = 2$), then the algorithm outputs ϵ.

Proof If $M = {A_n}^u \in \Gamma_n$, then $|L(2)| = |{A_n}^u(2)| = |nu + 2|$ in Step 1 of the first iteration.

If $n = 2$ and $u = -1$, then $|L(2)| = |nu + 2| = 0$ and so the algorithm outputs ϵ in Step 2 of the first iteration. Hence the algorithm terminates.

If $n \neq 2$ or $u \neq -1$, then in Step 1 of the first iteration, $|L(2)| = |{A_n}^u(2)| = |nu+2| > 1$. In Step 2, $\lfloor L(2) \rfloor = \lfloor nu+2 \rfloor = nu+2$ and $\lceil L(2) \rceil = \lceil nu+2 \rceil = nu + 2$. So $e = nu + 2$, $C = {A_1}^e = {A_1}^{nu+2}$, $w = wC = {A_1}^{nu+2}$ and $L = C^{-1}L = {A_1}^{-nu-2}{A_n}^u = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} \neq I$. Thus return Step 1. In Step 1 of the second iteration, $L(2) = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}(2) = 0$ and thus the algorithm outputs ϵ in Step 2 of the second iteration. Hence the algorithm terminates. □

Theorem 6.3.3 If $M = {B_n}^u$ with a nonzero integer u is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ.

Proof If $M = {B_n}^u \in \Gamma_n$, then $|L(\frac{1}{2})| = |{B_n}^u(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu+1}| = |\frac{1}{nu+2}|$.

If $n = 2$ and $u = -1$, then in Step 1 of the first iteration, $|L(\frac{1}{2})| = |{B_n}^u(\frac{1}{2})| = |\frac{1}{nu+2}| = \infty$. So in Step 3, the algorithm outputs ϵ and it terminates.

If $n \neq 2$ or $u \neq -1$, then in Step 1 of the first iteration, by Lemma 4.1.2, $|L(\frac{1}{2})| = |\frac{1}{nu+2}| < 1$. $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu+2 \rfloor = nu+2$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu+2 \rceil = nu+2$. In Step 3, as n is even, $e = nu + 2$, $C = {B_1}^e = {B_1}^{nu+2}$, $w = wC = {B_1}^{nu+2}$,

$L = C^{-1}L = {B_1}^{-nu-2}{B_n}^u = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \neq I$. So return Step 1. In Step 1 of the second iteration, $L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}(\frac{1}{2}) = \infty$ and so the algorithm outputs ϵ. Hence the algorithm terminates. □

Theorem 6.3.4 If $M = {B_n}^u$ with a nonzero integer u is input to the algorithm ($z = 2$), then the algorithm outputs ${B_1}^{nu}$ as the X_1-representation of M.

Proof If $M = {B_n}^u \in \Gamma_n$, then in Step 1 of the first iteration, by Lemma 4.1.2, $|L(2)| = |{B_n}^u(2)| = |\frac{2}{2nu+1}| = |\frac{1}{nu+\frac{1}{2}}| < 1$. $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu + \frac{1}{2} \rfloor = nu$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu + \frac{1}{2} \rceil = nu + 1$. In Step 3, as n is even, $e = nu$, $C = {B_1}^e = {B_1}^{nu}$, $w = wC = {B_1}^{nu}$ and $L = C^{-1}L = {B_1}^{-nu}{B_n}^u = I$. Hence the algorithm outputs ${B_1}^{nu}$ as the X_1-representation of M and it terminates.

Theorem 6.3.5 If $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M where odd $m \geq 3$ and u_i is a nonzero integer ($i = 1, \cdots, m$).

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$ with $m \geq 3$ and nonzero $u_i \in \mathbb{Z}(i = 1, \cdots, m)$, put $L(\frac{1}{2}) = {A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| = |nu_1 + \beta_1| > 1$ and by Theorem 4.1.5, $|\beta_1| = |{B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(\frac{1}{2})| < 1$. So in Step 1 of the first iteration, $|L(\frac{1}{2})| > 1$.

For $-1 < \beta_1 < 0$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_1 + \beta_1 \rfloor = nu_1 - 1$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_1 + \beta_1 \rceil = nu_1$. So in Step 2 of the first iteration, as n is even, $e = \lceil L(\frac{1}{2}) \rceil = nu_1$, $C = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}{A_1}^{n_m}$

$= {B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}} {A_1}^{n_m}$. As $L \neq I$, return Step 1.

For $\beta_1 = 0$, in Step 2 of the first iteration, as n is even, $e = \lfloor L(\frac{1}{2}) \rfloor = nu_1$, $C = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_1}^{nu_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ $= {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_1 < 1$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_1 + \beta_1 \rfloor = nu_1$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_1 + \beta_1 \rceil = nu_1 + 1$. So in Step 2 of the first iteration, as n is even, $e = \lfloor L(\frac{1}{2}) \rfloor = nu_1$, $C = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_1}^{nu_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, in the $i-1$th iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ according as $i-1$ is even or odd.

For odd i, in Step 1 of the ith iteration, put $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ $(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.3, $|L(\frac{1}{2})| > 1$ and by Theorem 4.1.4, $|\beta_i| < 1$.

For $-1 < \beta_i < 0$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i - 1$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. So in Step 2 of the ith iteration, as n is even, $e = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$, $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. In Step 2 of the ith iteration, as n is even, $e = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i + 1$. In Step 2 of the ith iteration, as n is even, $e = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For even i, let $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{nu_i\alpha_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. In Step 3 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| < 1$ and by Theorem 4.1.3, $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. So, in Step 3 of the ith iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$, $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i + 1$. So in Step 3 of the ith iteration, as n is even, $e = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in Step 1 of the mth iteration, By Theorem 6.3.1, $|L(\frac{1}{2})| = |{A_n}^{u_m}(\frac{1}{2})| = |nu_m + \frac{1}{2}| > 1$. $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_m + \frac{1}{2} \rfloor = nu_m$ and $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_m + \frac{1}{2} \rfloor = nu_m + 1$. So in Step 2 of the ith iteration, As n is even, $e = \lfloor L(\frac{1}{2}) \rfloor = nu_m$, $C = {A_1}^e = {A_1}^{nu_m}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}{A_1}^{nu_m}$ and $L = C^{-1}L = {A_1}^{-nu_m}{A_n}^{u_m} = I$. Thus the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M and the algorithm terminates. $\square$

Theorem 6.3.6 If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm $(z = 2)$, then the algorithm outputs ϵ where odd $m \geq 3$ and u_i is a nonzero integer $(i = 1, \cdots, m)$.

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$, put $L(2) = {A_n}^{u_1}{B_n}^{u_2}\cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. By Lemma 4.1.1, ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$ and by Theorem 4.1.3, $|L(2)| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| > 1$. By Theorem 4.1.4, $|\beta_1| = |{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_2}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| < 1$. So in Step 1 of the first iteration, $|L(2)| > 1$ and $|\beta_1| < 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the first iteration, by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-2}}$ ${B_n}^{u_{m-1}}(nu_m + 2)| > 1$. By Theorem 4.1.6, $|\beta_1| = |{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) =$ ${B_n}^{u_2}$
$\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m + 2)| < 1$.

For $-1 < \beta_1 < 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_1 + \beta_1\rfloor = nu_1 - 1$ and $\lceil L(\frac{1}{2})\rceil = nu_1$. So in Step 2 of the first iteration, as n is even, $e = \lceil L(\frac{1}{2})\rceil = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} = {B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_1 + \beta_1\rfloor = nu_1$ and $\lceil L(\frac{1}{2})\rceil = nu_1$. In Step 2 of the first iteration, as n is even, $e = \lfloor L(\frac{1}{2})\rfloor = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} = {B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} \neq I$. So return Step 1.

For $0 < \beta_1 < 1$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_1 + \beta_1\rfloor = nu_1$ and $\lceil L(\frac{1}{2})\rceil = nu_1 + 1$. In Step 2 of the first iteration, as n is even, $e = \lfloor L(\frac{1}{2})\rfloor = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} = {B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-2$, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in the $i-1$th iteration according as $i-1$ is even or odd.

For odd i, put $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. By Lemma 4.1.1, ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$ and by Theorem 4.1.3, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| > 1$. So in Step 1 of the ith iteration, $|L(2)| > 1$ and by Theorem 4.1.4, $|\beta_i| = |{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| < 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-2}}$ ${B_n}^{u_{m-1}}(nu_m+2)| > 1$. By Theorem 4.1.6, $|\beta_i| = |{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m + 2)| < 1$.

For $-1 < \beta_i < 0$, $\lfloor L(2)\rfloor = \lfloor nu_i + \beta_i\rfloor nu_i - 1$ and $\lceil L(2)\rceil = nu_i$. In Step

2 of the ith iteration, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_n}^{u_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_n}^{u_{i-1}}{A_i}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_1}^{nu_i}{B_1}^{nu_{i+1}}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} = {B_1}^{nu_{i+1}}\cdots{B_1}^{nu_{m-1}}{A_1}^{n_m} \neq I$. So, return Step 1.

For $\beta_i = 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. In Step 2 of the ith iteration, as n is even, $e = \lfloor L(2) \rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_1}^{nu_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i + 1$. In Step 2 of the ith iteration, as n is even, $e = \lfloor L(2) \rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_1}^{nu_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For even i, put $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{nu_i\alpha_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. By Lemma 4.1.1, ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$ and by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| < 1$. By Theorem 4.1.3, $\alpha_i = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0) \in D^c$. So in Step 1 of the ith iteration, $|L(2)| < 1$ and $|\alpha_i| > 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots$

${A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m{+}2)| < 1$. By Theorem 4.1.5, $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$ $= {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m + 2) \in D^c$. Thus in Step 1 of the ith iteration, $|L(2)| < 1$ and $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. So, in Step 3 of the ith iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha} < 1$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i + 1$. So in Step 3 of the ith iteration, as n is even, $e = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$, $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m - 1$, then in Step 1 of the $m - 1$th iteration, $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. So $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = 0$ in Step 3 of the $m - 1$th iteration and the algorithm outputs ϵ. Hence the algorithm terminates.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{m-1}}(nu_m + 2)| = |\frac{1}{nu_{m-1} + \frac{1}{nu_m+2}}| < 1$ in Step 1 of the $m - 1$th iteration. So in Step 3 of the $m - 1$th iteration, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil$.

For $-1 < \frac{1}{nu_m+2} < 0$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor = nu_{m-1} - 1$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil = nu_{m-1}$. So in Step 3 of the $m-1$th iteration, as n is even,

$e = \lceil \frac{1}{L(2)} \rceil = nu_{m-1}$, $C = {B_1}^e = {B_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{nu_m+2} < 1$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor = nu_{m-1}$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil = nu_{m-1} + 1$. So in Step 3 of the $m-1$th iteration, as n is even, $e = \lceil \frac{1}{L(2)} \rceil = nu_{m-1}$, $C = {B_1}^e = {B_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in Step 1 of the mth iteration, $|L(2)| = |{A_n}^{u_m}(2)| = |nu_m + 2|$. By Theorem 6.3.2, the algorithm outputs ϵ in Step 2 of the mth iteration and the algorithm terminates. □

Theorem 6.3.7 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ as the X_1-representation of M where even $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$).

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ with $m \geq 2$ and nonzero $u_i \in \mathbb{Z}$ ($i = 1, 2, \cdots, m$), put $L(\frac{1}{2}) = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. In Step 1 of the first iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| < 1$ and by Theorem 4.1.3, $|\alpha_1| > 1$.

For $-1 < \frac{1}{\alpha_1} < 0$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1 - 1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1$. So, in Step 2 of the first iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_1$, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$, $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ $= {A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_1} < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1 + 1$. So in Step 2 of the first iteration, as n is even, $e = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1$, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$, $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, in the $i-1$th iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ according as $i-1$ is even or odd.

For odd i, let $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{nu_i\alpha_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. In Step 3 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| < 1$ and by Theorem 4.1.3, $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. So, in Step 3 of the ith iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$, $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i + 1$. So in Step 3 of the ith iteration, as n is even, $e = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For even i, in Step 1 of the ith iteration, put $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ $(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.3, $|L(\frac{1}{2})| > 1$ and by Theorem 4.1.4, $|\beta_i| < 1$.

For $-1 < \beta_i < 0$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i - 1$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. So in Step 2 of the ith iteration, as n is even, $e = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$, $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. In Step 2 of the ith iteration, as n is even, $e = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i + 1$. In Step 2 of the ith iteration, as n is even, $e = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in Step 1 of the mth iteration, By Theorem 6.3.1, $|L(\frac{1}{2})| = |{A_n}^{u_m}(\frac{1}{2})| = |nu_m + \frac{1}{2}| > 1$. $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_m + \frac{1}{2} \rfloor = nu_m$ and $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_m + \frac{1}{2} \rfloor = nu_m + 1$. So in Step 2 of the ith iteration, As n is even, $e = \lfloor L(\frac{1}{2}) \rfloor = nu_m$, $C = {A_1}^e = {A_1}^{nu_m}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}{A_1}^{nu_m}$ and $L = C^{-1}L = {A_1}^{-nu_m}{A_n}^{u_m} = I$. Thus the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M and the algorithm terminates. □

Theorem 6.3.8 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm ($z = 2$), then the algorithm outputs ϵ where even $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$).

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ with $m \geq 2$ and nonzero $u_i \in \mathbb{Z}(i = 1, \cdots, m)$, put $L(2) = {B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1+1} = \frac{1}{nu_1+\frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. By Lemma 4.1.1, ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$ and by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| < 1$. By Theorem 4.1.3, $\alpha_i = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_2}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0) \in D^c$. So in Step 1 of the ith iteration, $|L(2)| < 1$ and $|\alpha_1| > 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-2}}$ ${B_n}^{u_{m-1}}(nu_m + 2)| < 1$. By Theorem 4.1.5, $\alpha_i = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_2}\cdots{B_n}^{u_{m-3}}{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m + 2) \in D^c$. Thus in Step 1 of the ith iteration, $|L(2)| < 1$ and $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_1} < 0$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1 - 1$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1$. So, in Step 3 of the first iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_1$, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_1} < 1$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1 + 1$. So in Step 3 of the ith iteration, as n is even, $e = \lfloor nu_i + \frac{1}{\alpha_1} \rfloor = nu_1$, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_1}^{nu_i}$, $L = C^{-1}L = {B_1}^{-nu_1}{A_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-2$, $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in the $i-1$th iteration according as $i-1$ is even or odd.

For even i, put $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. By Lemma 4.1.1, ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$ and by Theorem 4.1.3, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| > 1$. So in Step 1 of the ith iteration, $|L(2)| > 1$ and by Theorem 4.1.4, $|\beta_i| = |{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| < 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m+2)| > 1$. By Theorem 4.1.6, $|\beta_i| = |{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m + 2)| < 1$.

For $-1 < \beta_i < 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor nu_i - 1$ and $\lceil L(2) \rceil = nu_i$. In Step 2 of the ith iteration, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_n}^{u_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_n}^{u_{i-1}}{A_i}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_1}^{nu_i}{B_1}^{nu_{i+1}} \cdots {B_1}^{nu_{m-1}}{A_1}^{n_m} = {B_1}^{nu_{i+1}} \cdots {B_1}^{nu_{m-1}}{A_1}^{n_m} \neq I$. So, return Step 1.

For $\beta_i = 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. In Step 2 of the ith iteration, as n is even, $e = \lfloor L(2) \rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L =$

$C^{-1}L = {A_1}^{-nu_i}{A_1}^{nu_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i + 1$. In Step 2 of the ith iteration, as n is even, $e = \lfloor L(2) \rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_1}^{nu_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For odd i, put $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{nu_i\alpha_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. By Lemma 4.1.1, ${A_n}^{u_{m-2}}(0) = nu_{m-2} \in D^c$ and by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0)| < 1$. By Theorem 4.1.3, $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}(0) \in D^c$. So in Step 1 of the ith iteration, $|L(2)| < 1$ and $|\alpha_i| > 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m+2)| < 1$. By Theorem 4.1.5, $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-3}}{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}(nu_m + 2) \in D^c$. Thus in Step 1 of the ith iteration, $|L(2)| < 1$ and $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. So, in Step 3 of the ith iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$.

So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i + 1$. So in Step 3 of the ith iteration, as n is even, $e = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$, $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then in Step 1 of the $m-1$th iteration, $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 2$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = 0$ and ${B_n}^{u_{m-1}}(0) = 0$. So $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = 0$ in Step 3 of the $m-1$th iteration and the algorithm outputs ϵ. Hence the algorithm terminates.

If $n \neq 2$ or $u \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{m-1}}(nu_m + 2)| = |\frac{1}{nu_{m-1} + \frac{1}{nu_m+2}}| < 1$ in Step 1 of the $m-1$th iteration. So in Step 3 of the $m-1$th iteration, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil$.

For $-1 < \frac{1}{nu_m+2} < 0$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor = nu_{m-1} - 1$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil = nu_{m-1}$. So in Step 3 of the $m-1$th iteration, as n is even, $e = \lceil \frac{1}{L(2)} \rceil = nu_{m-1}$, $C = {B_1}^e = {B_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{nu_m+2} < 1$, $\lfloor \frac{1}{L(2)} \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor = nu_{m-1}$ and $\lceil \frac{1}{L(2)} \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil = nu_{m-1} + 1$. So in Step 3 of the $m-1$th iteration, as n is even, $e = \lceil \frac{1}{L(2)} \rceil = nu_{m-1}$, $C = {B_1}^e = {B_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}}$ and $L = C^{-1}L =$

${B_1}^{-nu_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in Step 1 of the mth iteration, $|L(2)| = |{A_n}^{u_m}(2)| = |nu_m + 2|$. By Theorem 6.3.2, the algorithm outputs ϵ in Step 2 of the mth iteration and the algorithm terminates. □

Theorem 6.3.9 If $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ where even $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$).

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ with even $m \geq 2$ and each nonzero $u_i \in \mathbb{Z}(i = 1, \cdots, m)$, put $L(\frac{1}{2}) = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m+1} = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ $= {A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$. So in Step 1 of the first iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$ and by Theorem 4.1.4, $|\beta_1| = |{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_2} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. So in Step 1 of the first iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$. By Theorem 4.1.4, $|\beta_1| = |{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

For $-1 < \beta_1 < 0$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_1 + \beta_1 \rfloor = nu_1 - 1$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_1 + \beta_1 \rceil = nu_1$. So in Step 2 of the first iteration, as n is even, $e = \lceil L(\frac{1}{2}) \rceil = nu_1$, $C =$

${A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m} = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $\beta_1 = 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_1 + \beta_1\rfloor = nu_1$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_1 + \beta_1\rceil = nu_1$. So in Step 2 of the first iteration, as n is even, $e = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_1 < 1$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_1+\beta_1\rfloor = nu_1$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_1+\beta_1\rceil = nu_1+1$. In Step 2 of the first iteration, as n is even, $e = \lfloor L(\frac{1}{2})\rfloor = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-3$, in the $i-1$th iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ according as $i-1$ is even or odd.

For odd i, let $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m+1} = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\infty) = nu_{m-1}+\infty = \infty$ and ${B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$ and by Theorem 4.1.4, $|\beta_i| = |{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. In Step 1 of the ith iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| =$

$|{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$. By Theorem 4.1.4, $|\beta_i| = |{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

For $-1 < \beta_i < 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_i+\beta_i\rfloor = nu_i-1$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_i+\beta_i\rceil = nu_i$. So in Step 2 of the ith iteration, as n is even, $e = \lceil L(\frac{1}{2})\rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_i}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_i + \beta_i\rfloor = nu_i$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_i + \beta_i\rceil = nu_i$. So in Step 2 of the ith iteration, as n is even, $e = \lceil L(\frac{1}{2})\rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_i}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_i+\beta_i\rfloor = nu_i$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_i+\beta_i\rceil = nu_i+1$. So in Step 2 of the i th iteration, as n is even, $e = \lfloor L(\frac{1}{2})\rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For even i, let $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m+1} = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ $= {A_n}^{u_{m-1}}(\infty) = nu_{m-1}+\infty = \infty$ and ${B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$ and by Theorem 4.1.3, $\alpha_i = {A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. In Step 1 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$. By Theorem 4.1.3, $|\alpha_i| = |{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. So in Step 3 of the ith iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_i}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i + 1$. So in Step 2, as n is even, $e = \lfloor \frac{1}{L(\frac{1}{2})} \rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}$
${B_n}^{u_m} \neq I$. So return Step 1.

If $i = m-2$, then $L = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}$ and put $L(\frac{1}{2}) = {B_n}^{u_{m-2}}(\alpha_{m-2}) = \frac{1}{nu_{m-2}+\frac{1}{\alpha_{m-2}}}$ where $\alpha_{m-2} = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then in Step 1 of the $m-2$th iteration, $L(\frac{1}{2}) = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}(\infty) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$ and $\alpha_{m-2} = {A_n}^{u_{m-1}}(\infty) = \infty$. Since —$L(\frac{1}{2})| < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \rfloor nu_{m-2} \lfloor = nu_{m-2}$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_{m-2} \rceil = nu_{m-2}$. So in Step 3 of the $m-2$th iteration, as n is even, $e = nu_{m-2}$, $C = {B_1}^e = {B_1}^{nu_{m-2}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-3}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-3}}{B_1}^{nu_{m-2}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}$ $= {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. By Lemma 4.1.2, $\alpha_{m-2} = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1}+\frac{1}{nu_m+2} \in D^c$. So in Step 1 of the $m-2$th iteration, by Lemma 4.1.2, $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = |{B_n}^{u_{m-2}}(\alpha_{m-2})| = |\frac{1}{nu_{m-2}+\frac{1}{\alpha_{m-2}}}| < 1$. Then $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}}\rfloor$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil$.

For $-1 < \frac{1}{\alpha_{m-2}} < 0$, $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}}\rfloor = nu_{m-2} - 1$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil = nu_{m-2}$. So in Step 3 of the $m-2$th iteration, as n is even, $e = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil = nu_{m-2}$, $C = {B_1}^e = {B_1}^{nu_{m-2}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}{B_1}^{nu_{m-2}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_{m-2}} < 1$, $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}}\rfloor = nu_{m-2}$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil = nu_{m-2} + 1$. So in Step 3 of the $m-2$th iteration, as n is even, $e = \lfloor nu_{m-2}+\frac{1}{\alpha_{m-2}}\rfloor = nu_{m-2}$, $C = {B_1}^e = {B_1}^{nu_{m-2}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}$ $C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}{B_1}^{nu_{m-2}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$ ${B_n}^{u_m} = {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then $L = {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\infty) = \infty$ and so, in Step 1 of the $m-1$th iteration, the algorithm outputs ϵ. Hence the algorithm terminates.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. So in Step 1 of the $m-1$th iteration, $|L(\frac{1}{2})| > 1$. Then consider $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_{m-1}+\frac{1}{nu_m+2}\rfloor$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2}\rceil$.

For $-1 < \frac{1}{nu_m+2} < 0$, then $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2}\rfloor = nu_{m-1} - 1$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2}\rceil = nu_{m-1}$. So in Step 2 of the $m-1$th iteration, as n is even, $e = \lceil L(\frac{1}{2})\rceil = nu_{m-1}$, $C = {A_1}^e = {A_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-2}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m} \neq I$. So return Step 1. If $i = m$, then in the mth iteration, $L(\frac{1}{2}) = {B_n}^{u_m}(\frac{1}{2})$. By Theorem 6.3.3, the algorithm outputs ϵ and it terminates.

For $0 < \frac{1}{nu_m+2} < 1$, then $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2}\rfloor = nu_{m-1}$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2}\rceil = nu_{m-1} + 1$. So in Step 2 of the $m-1$th iteration, as n is even, $e = \lfloor L(\frac{1}{2})\rfloor = nu_{m-1}$, $C = {A_1}^e = {A_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-2}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m} \neq I$. So return Step 1. If $i = m$, then in the mth iteration, $L(\frac{1}{2}) = {B_n}^{u_m}(\frac{1}{2})$. By Theorem 6.3.3, the algorithm outputs ϵ and it terminates. □

Theorem 6.3.10 If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = 2$), then the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M where even $m \geq 2$ and u_i is a nonzero integer $(i = 1, \cdots, m)$.

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ with even $m \geq 2$ and nonzero $u_i \in \mathbb{Z}(i = 1, \cdots, m)$, put $L(2) = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| > 1$ in Step 1 of the algorithm and by Theorem 4.1.6, $|\beta_1| < 1$.

For $-1 < \beta_1 < 0$, $\lfloor L(2)\rfloor = \lfloor nu_1 + \beta_1\rfloor = nu_1 - 1$ and $\lceil L(2)\rceil = \lceil nu_1 + \beta\rceil = nu_1$. So in Step 2, as n is even, $e = \lceil L(2)\rceil = nu_1$, $C = {A_1}^e =$

${A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $\beta_1 = 0$, $\lfloor L(2) \rfloor = \lfloor nu_1 + \beta_1 \rfloor = nu_1$ and $\lceil L(2) \rceil = \lceil nu_1 + \beta_1 \rceil = nu_1$. So in Step 2, as n is even, $e = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_1 < 1$, $\lfloor L(2) \rfloor = \lfloor nu_1 + \beta_1 \rfloor = nu_1$ and $\lceil L(2) \rceil = \lceil nu_1 + \beta_1 \rceil = nu_1 + 1$. So in Step 2, as n is even, $e = \lfloor L(2) \rfloor = nu_1$, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, in the $i-1$th iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ according as $i-1$ is even or odd.

For odd i, let $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)$.

For $-1 < \beta_i < 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta \rfloor = nu_i - 1$ and $\lceil L(2) \rceil = \lceil nu_i + \beta \rceil = nu_i$. In Step 2, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$, $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. In Step 2, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}$

${A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(2)\rfloor = \lfloor nu_i+\beta_i\rfloor = nu_i$ and $\lceil L(2)\rceil = \lceil nu_i+\beta_i\rceil = nu_i+1$. In Step 2, as n is even, $e = \lfloor L(2)\rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For even i, let $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| < 1$ and by Theorem 4.1.5, $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor L(2)\rfloor = \lfloor nu_i + \frac{1}{\alpha_i}\rfloor = nu_i - 1$ and $\lceil L(2)\rceil = \lceil nu_i + \frac{1}{\alpha_i}\rceil = nu_i$. In Step 2, as n is even, $e = \lceil L(2)\rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor L(2)\rfloor = \lfloor nu_i + \frac{1}{\alpha_i}\rfloor = nu_i$ and $\lceil L(2)\rceil = \lceil nu_i + \frac{1}{\alpha_i}\rceil = nu_i + 1$. In Step 2, as n is even, $e = \lfloor L(2)\rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then $L(2) = {B_n}^{u_m}(2)$. By Theorem 6.3.4, the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M and the algorithm terminates. □

Theorem 6.3.11 If $M = {B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ where odd $m \geq 2$ and u_i is a nonzero integer $(i = 1, \cdots, m)$.

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ with odd $m \geq 2$ and each nonzero $u_i \in \mathbb{Z}(i = 1, \cdots, m)$, put $L(\frac{1}{2}) = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 n u_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m + 1} = \frac{1}{nu_m + 2} = \infty$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ $= {A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$ and by Theorem 4.1.3, $\alpha_1 = {A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m + 2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m + 2}) = nu_{m-1} + \frac{1}{nu_m + 2} \in D^c$. In Step 1 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m + 2})| < 1$. By Theorem 4.1.3, $|\alpha_1| = |{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m + 2})| > 1$.

For $-1 < \frac{1}{\alpha_1} < 0$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1 - 1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1$. So in Step 3 of the first iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_1$, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_2}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m} = {A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_1} < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1 + 1$. So in Step 2, as n is even, $e = \lfloor \frac{1}{L(\frac{1}{2})} \rfloor = nu_1$, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-3$, in the $i-1$th iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ according as $i-1$ is even or odd.

For even i, let $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m+1} = \frac{1}{nu_m+2} = \infty$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ $= {A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$ and by Theorem 4.1.4, $|\beta_i| = |{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. In Step 1 of the ith iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$. By Theorem 4.1.4, $|\beta_i| = |{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$.

For $-1 < \beta_i < 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i - 1$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_i + \beta_i \rceil = nu_i$. So in Step 2 of the ith iteration, as n is even, $e = \lceil L(\frac{1}{2})\rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots {B_1}^{nu_{i-1}}{A_i}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(\frac{1}{2})\rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(\frac{1}{2})\rceil = \lceil nu_i + \beta_i \rceil = nu_i$. So in Step 2 of the ith iteration, as n is even, $e = \lceil L(\frac{1}{2})\rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {B_1}^{nu_{i-1}}{A_i}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$.

So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_i + \beta_i \rceil = nu_i + 1$. So in Step 2 of the i th iteration, as n is even, $e = \lfloor L(\frac{1}{2}) \rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i + 1} = \frac{1}{nu_i + \frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m + 1} = \frac{1}{nu_m + 2} = \infty$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ $= {A_n}^{u_{m-1}}(\infty) = nu_{m-1} + \infty = \infty$ and ${B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| < 1$ and by Theorem 4.1.3, $\alpha_i = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-3}}(\frac{1}{nu_{m-2}})| > 1$.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m + 2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m + 2}) = nu_{m-1} + \frac{1}{nu_m + 2} \in D^c$. In Step 1 of the ith iteration, by Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m + 2})| < 1$. By Theorem 4.1.3, $|\alpha_i| = |{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m + 2})| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. So in Step 3 of the ith iteration, as n is even, $e = \lceil \frac{1}{L(\frac{1}{2})} \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}B_i^{nu_i}$ and $L = C^{-1}L = B_i^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_i + \frac{1}{\alpha_i}\rfloor = nu_i$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_i + \frac{1}{\alpha_i}\rceil = nu_i + 1$. So in Step 2, as n is even, $e = \lfloor\frac{1}{L(\frac{1}{2})}\rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m-2$, then $L = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}$ and put $L(\frac{1}{2}) = {B_n}^{u_{m-2}}(\alpha_{m-2}) = \frac{1}{nu_{m-2}+\frac{1}{\alpha_{m-2}}}$ where $\alpha_{m-2} = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then in Step 1 of the $m-2$th iteration, $L(\frac{1}{2}) = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}(\infty) = {B_n}^{u_{m-2}}(\infty) = \frac{1}{nu_{m-2}} \in D$ and $\alpha_{m-2} = {A_n}^{u_{m-1}}(\infty) = \infty$. Since $|L(\frac{1}{2})| < 1$, $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_{m-2}\rfloor = nu_{m-2}$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_{m-2}\rceil = nu_{m-2}$. So in Step 3 of the $m-2$th iteration, as n is even, $e = nu_{m-2}$, $C = {B_1}^e = {B_1}^{nu_{m-2}}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}{B_1}^{nu_{m-2}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}$ $= {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$. By Lemma 4.1.2, $\alpha_{m-2} = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1}+\frac{1}{nu_m+2} \in D^c$. So in Step 1 of the $m-2$th iteration, by Lemma 4.1.2, $|L(\frac{1}{2})| = |{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = |{B_n}^{u_{m-2}}(\alpha_{m-2})| = |\frac{1}{nu_{m-2}+\frac{1}{\alpha_{m-2}}}| < 1$. Then $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}}\rfloor$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil$.

For $-1 < \frac{1}{\alpha_{m-2}} < 0$, $\lfloor\frac{1}{L(\frac{1}{2})}\rfloor = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}}\rfloor = nu_{m-2} - 1$ and $\lceil\frac{1}{L(\frac{1}{2})}\rceil = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil = nu_{m-2}$. So in Step 3 of the $m-2$th iteration, as n is even, $e = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}}\rceil = nu_{m-2}$, $C = {B_1}^e = {B_1}^{nu_{m-2}}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{m-3}}{B_1}^{nu_{m-2}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_{m-2}} < 1$, $\lfloor \frac{1}{L(\frac{1}{2})} \rfloor = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}} \rfloor = nu_{m-2}$ and $\lceil \frac{1}{L(\frac{1}{2})} \rceil = \lceil nu_{m-2} + \frac{1}{\alpha_{m-2}} \rceil = nu_{m-2} + 1$. So in Step 3 of the $m-2$th iteration, as n is even, $e = \lfloor nu_{m-2} + \frac{1}{\alpha_{m-2}} \rfloor = nu_{m-2}$, $C = {B_1}^e = {B_1}^{nu_{m-2}}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{m-3}}$ $C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{m-3}}{B_1}^{nu_{m-2}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-2}}{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}$ ${B_n}^{u_m} = {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then $L = {A_n}^{u_{m-1}}{B_n}^{u_m}$ and $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 2$ and $u_m = -1$, then $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\infty) = \infty$ and so, in Step 1 of the $m-1$th iteration, the algorithm outputs ϵ. Hence the algorithm terminates.

If $n \neq 2$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. So in Step 1 of the $m-1$th iteration, $|L(\frac{1}{2})| > 1$. Then consider $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil$.

For $-1 < \frac{1}{nu_m+2} < 0$, then $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor = nu_{m-1} - 1$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil = nu_{m-1}$. So in Step 2 of the $m-1$th iteration, as n is even, $e = \lceil L(\frac{1}{2}) \rceil = nu_{m-1}$, $C = {A_1}^e = {A_1}^{nu_{m-1}}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{m-2}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m} \neq I$. So return Step 1. If $i = m$, then in the mth iteration, $L(\frac{1}{2}) = {B_n}^{u_m}(\frac{1}{2})$. By Theorem 6.3.3, the algorithm outputs ϵ and it terminates.

For $0 < \frac{1}{nu_m+2} < 1$, then $\lfloor L(\frac{1}{2}) \rfloor = \lfloor nu_{m-1} + \frac{1}{nu_m+2} \rfloor = nu_{m-1}$ and $\lceil L(\frac{1}{2}) \rceil = \lceil nu_{m-1} + \frac{1}{nu_m+2} \rceil = nu_{m-1} + 1$. So in Step 2 of the $m-1$th iteration, as n is even, $e = \lfloor L(\frac{1}{2}) \rfloor = nu_{m-1}$, $C = {A_1}^e = {A_1}^{nu_{m-1}}$, $w = wC =$

${B_1}^{nu_1}{A_1}^{nu_2}\cdots {B_1}^{nu_{m-2}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m} \neq I$. So return Step 1. If $i = m$, then in the mth iteration, $L(\frac{1}{2}) = {B_n}^{u_m}(\frac{1}{2})$. By Theorem 6.3.3, the algorithm outputs ϵ and it terminates. □

Theorem 6.3.12 If $M = {B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = 2$),then the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2}\cdots {A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M where $m \geq 2$ and u_i is a nonzero integer ($i = 1, \cdots, m$).

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$, put $L(2) = {B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1+1} = \frac{1}{nu_1+\frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ (2). By Theorem 4.1.6, $|L(2)| < 1$ and by Theorem 4.1.5, $|\alpha_1| > 1$.

For $-1 < \frac{1}{\alpha_1} < 0$, $\lfloor L(2) \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1 - 1$ and $\lceil L(2) \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1$. In Step 2, as n is even, $e = \lceil L(2) \rceil = nu_1$, $C = {B_1}^{e} = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_1} < 1$, $\lfloor L(2) \rfloor = \lfloor nu_1 + \frac{1}{\alpha_1} \rfloor = nu_1$ and $\lceil L(2) \rceil = \lceil nu_1 + \frac{1}{\alpha_1} \rceil = nu_1 + 1$. In Step 2, as n is even, $e = \lfloor L(2) \rfloor = nu_1$, $C = {B_1}^{e} = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, in the $i-1$th iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ according as $i-1$ is even or odd.

For even i, let $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)$.

For $-1 < \beta_i < 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta \rfloor = nu_i - 1$ and $\lceil L(2) \rceil = \lceil nu_i + \beta \rceil = nu_i$. In Step 2, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$, $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $\beta_i = 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i$. In Step 2, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}$ ${A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \beta_i < 1$, $\lfloor L(2) \rfloor = \lfloor nu_i + \beta_i \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \beta_i \rceil = nu_i + 1$. In Step 2, as n is even, $e = \lfloor L(2) \rfloor = nu_i$, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i + 1} = \frac{1}{nu_i + \frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| < 1$ and by Theorem 4.1.5, $|\alpha_i| > 1$.

For $-1 < \frac{1}{\alpha_i} < 0$, $\lfloor L(2) \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i - 1$ and $\lceil L(2) \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i$. In Step 2, as n is even, $e = \lceil L(2) \rceil = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For $0 < \frac{1}{\alpha_i} < 1$, $\lfloor L(2) \rfloor = \lfloor nu_i + \frac{1}{\alpha_i} \rfloor = nu_i$ and $\lceil L(2) \rceil = \lceil nu_i + \frac{1}{\alpha_i} \rceil = nu_i + 1$. In Step 2, as n is even, $e = \lfloor L(2) \rfloor = nu_i$, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then $L(2) = {B_n}^{u_m}(2)$. By Theorem 6.3.4, the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M and the algorithm terminates. □

6.4 X_1-Representation Algorithm II

Let $n \geq 3$ be a natural number and $M \in \Gamma_n$. Assume that n is unknown. Input a matrix $M \in \Gamma_n$ to the X_1-representation algorithm II and then the algorithm outputs the X_1-representation of M as a reduced word in ${X_1}^{\pm}$. We use two values $z = \frac{1}{2}$ and $z = 2$. If the X_1-representation algorithm II computes the X_1-representation of M for $z = \frac{1}{2}$, we do not run the X_1-representation algorithm II for $z = 2$. Otherwise, we have to run the X_1-representation algorithm II for $z = 2$. So the X_1-representation algorithm II computes the X_1-representation of $M \in \Gamma_n$ for $z = \frac{1}{2}$ or $z = 2$. When the X_1-representation algorithm II does not output the X_1-representation of M for $z = \frac{1}{2}$, the algorithm outputs ϵ for $z = \frac{1}{2}$. So the X_1-representation algorithm outputs the X_1-representation of M or ϵ and then the algorithm terminates. Now we describe the X_1-representation algorithm II.

The X_1-Representation Algorithm II

Step 0

$\mathbf{w} \leftarrow \mathbf{1_{X_1}}$

$\mathbf{L} \leftarrow \mathbf{M}$.

```
Step 1
$\mathbf{L(z) = 0, |L(z)| = 1, L(z) = \infty \Rightarrow}$ output $\epsilon$.

$\mathbf{|L(z)| > 1 \Rightarrow}$ compute $\mathbf{e}$, $\mu$ s.t. $\mathbf{L(z) = e} + \mu$, $\mathbf{e} \in \mathbb{Z}$, $-\frac{1}{2} < \mu \leq \frac{1}{2}$
and go to Step 2.

$\mathbf{|L(z)| < 1 \Rightarrow}$ compute $\mathbf{e}$, $\mu$ s.t. $\mathbf{\frac{1}{L(z)} = e} + \mu$, $\mathbf{e} \in \mathbb{Z}$, $-\frac{1}{2} < \mu \leq \frac{1}{2}$
and go to Step 3.

Step 2
$\mathbf{C \leftarrow {A_1}^e}$ and $\mathbf{w \leftarrow wC}$.
$\mathbf{C = I \Rightarrow}$ output $\epsilon$.
$\mathbf{L \leftarrow C^{-1}L}$
$\mathbf{L = I \Rightarrow}$ output $\mathbf{w}$. Otherwise, return Step 1.

Step 3
$\mathbf{C \leftarrow {B_1}^e}$ and $\mathbf{w \leftarrow wC}$.
$\mathbf{C = I \Rightarrow}$ output $\epsilon$.
$\mathbf{L \leftarrow C^{-1}L}$
$\mathbf{L = I \Rightarrow}$ output $\mathbf{w}$. Otherwise, return Step 1.
```

6.5 Programming Implementation II

This section shows implementation of the X_1-representation algorithm II and so we demonstrate how the X_1-representation algorithm works correctly. We make a program called the X_1-representation program II with Maple version 6 and the operation of the program is one loop. The X_1-representation program II takes $z = \frac{1}{2}$ or $z = 2$, the entries $M11$, $M12$, $M21$ and $M22$ of $M \in \Gamma_n$ as inputs and then for every execution of the program, it outputs two matrices.

The first matrix presents $C = {A_1}^e$ in Step 2 or $C = {B_1}^e$ in Step 3. The second matrix presents $L = C^{-1}L$ in Step 2 or $L = C^{-1}L$ in Step 3. If the identity matrix turns up in the first matrix or the second matrix, then the program execution terminates. Also, if an usual matrix in the first matrix or the second matrix appears, then execution of the program terminates. Each example shows how the algorithm and the program work correctly to compute the X_1-representation of $M \in \Gamma_n$. The following is the X_1-representation program II source code.

X_1-Representation Program II Source Code

```
with(GaussInt):
with(linalg):
su:=proc(z::float, M11::integer, M12::integer, M21::integer, M22::integer):
local K, u, v, C, P, Q;
z;
K:=matrix(2,2,[M11, M12, M21, M22]);
L(z) := (M11 * z + M12)/(M21 * z + M22);
R(z) := (M21 * z + M22)/(M11 * z + M12);
if abs(L(z)) = 1 then
print(epsilon);
fi;
if abs(L(z)) > 1   then
u:=floor(L(z));
v:=ceil(L(z));
if abs(L(z) - u) < 0.5 then
C := matrix(2, 2, [1, 1, 0, 1])^{u};
P:=matrix(2,2,[1,-u,0,1]);
Q:=multiply(P, K);
elif abs(L(z) - u) = 0.5 then
```

```
C := matrix(2, 2, [1, 1, 0, 1])^{u};
P:=matrix(2,2,[1,-u,0,1]);
Q:=multiply(P, K);
else
C := matrix(2, 2, [1, 1, 0, 1])^{v};
P:=matrix(2,2,[1, -v,0,1]);
Q:=multiply(P, K);
fi;
print(C);
print(Q);
fi;
if abs(L(z)) < 1   then
u:=floor(R(z));
v:=ceil(R(z)))
if abs(R(z) - u) < 0.5 then
C := matrix(2, 2, [1, 0, 1, 1])^{u};
P:=matrix(2,2,[1,0,-v,1]);
Q:=multiply(P, K);
elif abs(R(z) - u) = 0.5 then
C := matrix(2, 2, [1, 0, 1, 1])^{u};
P:=matrix(2,2,[1,0,-u,1]);
Q:=multiply(P, K);
else
C := matrix(2, 2, [1, 0, 1, 1])^{v};
P:=matrix(2,2,[1,0,-v,1]);
Q:=multiply(P, K);
fi;
print(C);
print(Q);
```

```
fi;

end proc :
```

Example 1

Given $M = A_3 = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} \in \Gamma_3$, input $z = 0.5$, $M11 = 1$, $M12 = 3$, $M21 = 0$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,1,3,0,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^3$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-3} \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm II. So execution of the program terminates and take the first matrix of the first execution of the program. Then we have

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^3$$

as the X_3-representation of M.

For $z = 2$,

> su(2.0,1,3,0,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^5$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$$

> su(2.0,1,-2,0,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -2\\ -\infty & \infty\end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(2) = \begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}(2) = 0$ in Step 1 of the second iteration of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M for $z = 2$.

Example 2

Given $M = B_3 = \begin{pmatrix}1 & 0\\ 3 & 1\end{pmatrix} \in \Gamma_3$, input $z = 0.5$, $M11 = 1$, $M12 = 0$, $M21 = 3$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,1,0,3,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{5}$$

$$\begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}$$

> su(0.5,1,0,-2,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -\infty\\ -2 & 1\end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$ which is the same ϵ in Step 1 of the X_1-representation of the algorithm II because $L(\frac{1}{2} = \begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}(\frac{1}{2}) = \infty$. So execution of the program terminates and the program does not output the X_1-representation of M for $z = \frac{1}{2}$.

For $z = 2$,

```
> su(2.0,1,0,3,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{3}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-3}\begin{pmatrix}1 & 0\\ 3 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm II. So execution of the program terminates and take the first matrix of the first execution of the program. Then we have

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{3}$$

as the X_1-representation of M.

Example 3

Given $M = A_8 = \begin{pmatrix}1 & 8\\ 0 & 1\end{pmatrix} \in \Gamma_8$, input $z = 0.5$, $M11 = 1$, $M12 = 8$, $M21 = 0$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,1,8,0,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{8}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-8}\begin{pmatrix}1 & 8\\ 0 & 1\end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm II. So execution of the program terminates and take the first matrix of the first execution of the program. Then we have

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{8}$$

as the X_1-representation of M.

For $z = 2$,

```
> su(2.0,1,8,0,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{10}$$

$$\begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}$$

```
> su(2.0,1,-2,0,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}1 & -2\\ -\infty & \infty\end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(2) = \begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}(2) = 0$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

Example 4

Given $M = B_8 = \begin{pmatrix} 1 & 0 \\ 8 & 1 \end{pmatrix} \in \Gamma_8$, input $z = 0.5$, $M11 = 1$, $M12 = 0$, $M21 = 8$ and $M22 = 1$ to the program.

For $z = \frac{1}{2}$

```
> su(0.5,1,0,8,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$$

```
> su(0.5,1,0,-2,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} \infty & -\infty \\ -2 & 1 \end{pmatrix}$$

The first matrix of the second execution of the program is an unusual matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} (\frac{1}{2}) = \infty$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

```
> su(2.0,1,0,8,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{8}$$

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

The second matrix of the first execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-8}\begin{pmatrix}1 & 0\\8 & 1\end{pmatrix} = I$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and take the first matrix of the first execution of the program. Then we have

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{8}$$

as the X_1-representation of M.

Example 5

Given $M = A_7{}^{-3}B_7A_7{}^3 = \begin{pmatrix}-146 & -3087\\7 & 148\end{pmatrix} \in \Gamma_7$, input $z = 0.5$, $M11 = -146$, $M12 = -3087$, $M21 = 7$ and $M22 = 148$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,-146,-3087,7,148);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-21}$$

$$\begin{pmatrix}1 & 21\\7 & 148\end{pmatrix}$$

> su(0.5,1,21,7,148);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{7}$$

$$\begin{pmatrix}1 & 21\\0 & 1\end{pmatrix}$$

> su(0.5,1,21,0,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{21}$$

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

The second matrix of the third execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-21} \begin{pmatrix} 1 & 21 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-21} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{7} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{21}$$

as the X_1-representation of M.

For $z = 2$,

```
> su(2.0,-146,-3087,7,148);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-21}$$

$$\begin{pmatrix} 1 & 21 \\ 7 & 148 \end{pmatrix}$$

```
> su(2.0,1,21,7,148);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{7}$$

$$\begin{pmatrix} 1 & 21 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,21,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{23}$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,-2,0,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}1 & -2\\ -\infty & \infty\end{pmatrix}$$

The first matrix of the fourth matrix is an unusual matrix $\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(2) = \begin{pmatrix}1 & -2\\ 0 & 1\end{pmatrix}(2) = 0$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

Example 6

Given $M = {B_7}^{-1}{A_7}^{-3}B_7{A_7}^3 = \begin{pmatrix}-146 & -3087\\ 1029 & 21757\end{pmatrix}$, input $z = 0.5$, $M11 = -146$, $M12 = -3087$, $M21 = 1029$ and $M22 = 21757$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,-146,-3087,1029,21757);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}$$

$$\begin{pmatrix}-146 & -3087\\ 7 & 148\end{pmatrix}$$

```
> su(0.5,-146,-3087,7,148);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}$$

$$\begin{pmatrix}1 & 21\\ 7 & 148\end{pmatrix}$$

```
> su(0.5,1,21,7,148);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{7}$$

$$\begin{pmatrix}1 & 21\\ 0 & 1\end{pmatrix}$$

> su(0.5,1,21,0,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{21}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}\begin{pmatrix}1 & 21\\ 0 & 1\end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{7}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{21}$$

as the X_1-representation of M.

For $z = 2$,

> su(2.0,-146,-3087,1029,21757);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}$$

$$\begin{pmatrix}-146 & -3087\\ 7 & 148\end{pmatrix}$$

> su(2.0,-146,-3087,7,148);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}$$

$$\begin{pmatrix}1 & 21\\ 7 & 148\end{pmatrix}$$

> su(2.0,1,21,7,148);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^7$$

$$\begin{pmatrix}1 & 21\\0 & 1\end{pmatrix}$$

> su(2.0,1,21,0,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{23}$$

$$\begin{pmatrix}1 & -2\\0 & 1\end{pmatrix}$$

> su(2.0,1,-2,0,1);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}1 & -2\\-\infty & \infty\end{pmatrix}$$

The first matrix of the fifth execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(2) = \begin{pmatrix}1 & -2\\0 & 1\end{pmatrix}(2) = 0$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

Example 7

Given $M = {A_7}^{-3}B_7{A_7}^3{B_7}^{-1} = \begin{pmatrix}21463 & -3087\\1029 & 148\end{pmatrix}$, input $z = 0.5$, $M11 = 21463$, $M12 = -3087$, $M21 = -1029$ and $M22 = 148$ to the program.

For $z = \frac{1}{2}$

> su(0.5,21463,-3087,-1029,148);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}$$

$$\begin{pmatrix}-146 & 21\\ -1029 & 148\end{pmatrix}$$

```
> su(0.5,-146,21,-1029,148);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{7}$$

$$\begin{pmatrix}-146 & 21\\ -7 & 1\end{pmatrix}$$

```
> su(0.5,-146,21,-7,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{21}$$

$$\begin{pmatrix}1 & 0\\ -7 & 1\end{pmatrix}$$

```
> su(0.5,1,0,-7,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-5}$$

$$\begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}$$

```
> su(0.5,1,0,-2,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -\infty\\ -2 & 1\end{pmatrix}$$

The first matrix of the fifth execution of the program is an unusual matrix $\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(\frac{1}{2}) = \begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}(\frac{1}{2}) = \infty$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

> su(2.0,21463,-3087,-1029,148);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-21}$$

$$\begin{pmatrix}-146 & 21\\-1029 & 148\end{pmatrix}$$

> su(2.0,-146,21,-1029,148);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{7}$$

$$\begin{pmatrix}-146 & 21\\-7 & 1\end{pmatrix}$$

> su(2.0,-146,21,-7,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{21}$$

$$\begin{pmatrix}1 & 0\\-7 & 1\end{pmatrix}$$

> su(2.0,1,0,-7,1);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-7}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{7}\begin{pmatrix}1 & 0\\-7 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program. Then we have

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{-21}\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{7}\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{21}\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{-7}$$

as the X_1-representation of M.

Example 8

Given $M = {B_7}^{-1}{A_7}^{-3}B_7{A_7}^3{B_7}^{-1} = \begin{pmatrix} 21463 & -3087 \\ -151270 & 21757 \end{pmatrix} \in \Gamma_7$, input $z = 0.5$, $M11 = 21463$, $M12 = -3087$, $M21 = -151270$ and $M22 = 21757$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,21463,-3087,-151270,21757);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-7}$$

$$\begin{pmatrix} 21463 & -3087 \\ -1029 & 148 \end{pmatrix}$$

```
> su(0.5,21463,-3087,-1029,148);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-21}$$

$$\begin{pmatrix} -146 & 21 \\ -1029 & 148 \end{pmatrix}$$

```
> su(0.5,-146,21,-1029,148);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{7}$$

$$\begin{pmatrix} -146 & 21 \\ -7 & 1 \end{pmatrix}$$

```
> su(0.5,-146,21,-7,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{21}$$

$$\begin{pmatrix} 1 & 0 \\ -7 & 1 \end{pmatrix}$$

```
> su(0.5,1,0,-7,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-5}$$

$$\begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}$$

```
> su(0.5,1,0,-2,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}\infty & -\infty\\ -2 & 1\end{pmatrix}$$

The first matrix of the sixth execution of the program is an unusual matrix $\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(\frac{1}{2}) = \begin{pmatrix}1 & 0\\ -2 & 1\end{pmatrix}(\frac{1}{2}) = \infty$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

```
> su(2.0,21463,-3087,-151270,21757);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}$$

$$\begin{pmatrix}21463 & -3087\\ -1029 & 148\end{pmatrix}$$

```
> su(2.0,21463,-3087,-1029,148);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}$$

$$\begin{pmatrix}-146 & 21\\ -1029 & 148\end{pmatrix}$$

```
> su(2.0,-146,21,-1029,148);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{7}$$

$$\begin{pmatrix}-146 & 21\\ -7 & 1\end{pmatrix}$$

```
> su(2.0,-146,21,-7,1);
```

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{21}$$

$$\begin{pmatrix}1 & 0\\ -7 & 1\end{pmatrix}$$

```
> su(2.0,1,0,-7,1);
```

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}$$

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

The second matrix of the fifth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{7}\begin{pmatrix}1 & 0\\ -7 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-21}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{7}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{21}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-7}$$

as the X_1-representation of M.

Example 9

Given Given $M = A_{10}^{-3}B_{10}{A_{10}}^{3} = \begin{pmatrix}-299 & -9000\\ 10 & 301\end{pmatrix} \in \Gamma_{10}$, input $z = 0.5$, $M11 = -299$, $M12 = -9000$, $M21 = 10$ and $M22 = 301$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,-299,-9000,10,301);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30}$$

$$\begin{pmatrix} 1 & 30 \\ 10 & 301 \end{pmatrix}$$

> su(0.5,1,30,10,301);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} 1 & 30 \\ 0 & 1 \end{pmatrix}$$

> su(0.5,1,30,0,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

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

The second matrix of the third execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30} \begin{pmatrix} 1 & 30 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

as the X_1-representation of M.

For $z = 2$,

> su(2.0,-299,-9000,10,301);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30}$$

$$\begin{pmatrix} 1 & 30 \\ 10 & 301 \end{pmatrix}$$

> su(2.0,1,30,10,301);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{10}$$

$$\begin{pmatrix}1 & 30\\0 & 1\end{pmatrix}$$

> su(2.0,1,30,0,1);

$$\begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}^{32}$$

$$\begin{pmatrix}1 & -2\\0 & 1\end{pmatrix}$$

> su(2.0,1,-2,0,1);

$$\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$$

$$\begin{pmatrix}1 & -2\\-\infty & \infty\end{pmatrix}$$

The first matrix of the fourth execution of the program is an unusual matrix $\begin{pmatrix}1 & 0\\1 & 1\end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(2) = C^{-1}L = \begin{pmatrix}1 & -2\\0 & 1\end{pmatrix}(2) = 0$ in Step 1 of the X_1-representation of the algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

Example 10

Given $M = B_{10}^{-1}A_{10}^{-3}B_{10}{A_{10}}^{3} = \begin{pmatrix}-299 & -9000\\3000 & 90301\end{pmatrix} \in \Gamma_{10}$, input $z = 0.5$, $M11 = -299$, $M12 = -9000$, $M21 = 3000$ and $M22 = 90301$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,-299,-9000,3000,90301);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-10}$$

$$\begin{pmatrix} -299 & -9000 \\ 10 & 301 \end{pmatrix}$$

> su(0.5,-299,-9000,10,301);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30}$$

$$\begin{pmatrix} 1 & 30 \\ 10 & 301 \end{pmatrix}$$

> su(0.5,1,30,10,301);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} 1 & 30 \\ 0 & 1 \end{pmatrix}$$

> su(0.5,1,30,0,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = C^{-1}L = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30} \begin{pmatrix} 1 & 30 \\ 0 & 1 \end{pmatrix} = I$ in Step 2 of the X_1-representation algorithm II. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-10} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

as the X_1-representation of M.

For $z = 2$,

> su(2.0,-299,-9000,3000,90301);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-10}$$

$$\begin{pmatrix} -299 & -9000 \\ 10 & 301 \end{pmatrix}$$

```
> su(2.0,-299,-9000,10,301);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30}$$

$$\begin{pmatrix} 1 & 30 \\ 10 & 301 \end{pmatrix}$$

```
> su(2.0,1,30,10,301);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} 1 & 30 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,30,0,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{32}$$

$$\begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$$

```
> su(2.0,1,-2,0,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} 1 & -2 \\ -\infty & \infty \end{pmatrix}$$

The first matrix of the fifth execution of the program is an unusual matrix $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm because $L(2) = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}(2) = 0$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

Example 11

Given $M = A_{10}^{-3}B_{10}{A_{10}}^{3}B_{10}^{-1} = \begin{pmatrix} 89701 & -9000 \\ -3000 & 301 \end{pmatrix} \in \Gamma_{10}$, input $z = 0.5$, $M11 = 89700$, $M12 = -9000$, $M21 = -3000$ and $M22 = 301$ to the program.

For $z = \frac{1}{2}$,

```
> su(0.5,89701,-9000,-3000,301);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30}$$

$$\begin{pmatrix} -299 & 30 \\ -3000 & 301 \end{pmatrix}$$

```
> su(0.5,-299,30,-3000,301);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} -299 & 30 \\ -10 & 1 \end{pmatrix}$$

```
> su(0.5,-299,30,-10,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

$$\begin{pmatrix} 1 & 0 \\ -10 & 1 \end{pmatrix}$$

```
> su(0.5,1,0,-10,1);
```

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-8}$$

$$\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$$

```
> su(0.5,1,0,-2,1);
```

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} \infty & -\infty \\ -2 & 1 \end{pmatrix}$$

The first matrix of the fifth execution of the program is an unusual matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} (\frac{1}{2}) = \infty$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

> su(2.0,89701,-9000,-3000,301);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-30}$$

$$\begin{pmatrix} -299 & 30 \\ -3000 & 301 \end{pmatrix}$$

> su(2.0,-299,30,-3000,301);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} -299 & 30 \\ -10 & 1 \end{pmatrix}$$

> su(2.0,-299,30,-10,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

$$\begin{pmatrix} 1 & 0 \\ -10 & 1 \end{pmatrix}$$

> su(2.0,1,0,-10,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}$$

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

The second matrix of the fourth execution of the program is the identity matrix which is $L = C^{-1}L = (\frac{1}{2})\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{10}\begin{pmatrix}1 & 0\\ -10 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-30}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{10}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{30}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}$$

as the X_1-representation of M.

Example 12

Given $M = B_{10}^{-1}A_{10}^{-3}B_{10}A_{10}{}^{3}B_{10}^{-1} = \begin{pmatrix}89701 & -9000\\ -900010 & 90301\end{pmatrix} \in \Gamma_{10}$, input $z = 0.5$, $M11 = 89701$, $M12 = -9000$, $M21 = -900010$ and $M22 = 90301$ to the program.

For $z = \frac{1}{2}$,

> su(0.5,89701,-9000,-900010,90301);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}$$

$$\begin{pmatrix}89701 & -9000\\ -3000 & 301\end{pmatrix}$$

> su(0.5,89701,-9000,-3000,301);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-30}$$

$$\begin{pmatrix} -299 & 30 \\ -3000 & 301 \end{pmatrix}$$

> su(0.5,-299,30,-3000,301);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{10}$$

$$\begin{pmatrix} -299 & 30 \\ -10 & 1 \end{pmatrix}$$

> su(0.5,-299,30,-10,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{30}$$

$$\begin{pmatrix} 1 & 0 \\ -10 & 1 \end{pmatrix}$$

> su(0.5,1,0,-10,1);

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{-8}$$

$$\begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}$$

> su(0.5,1,0,-2,1);

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$$

$$\begin{pmatrix} \infty & -\infty \\ -2 & 1 \end{pmatrix}$$

The first matrix of the sixth execution of the program is an unusual matrix $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\infty}$ which is the same as ϵ in Step 1 of the X_1-representation algorithm II because $L(\frac{1}{2}) = C^{-1}L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}(\frac{1}{2}) = \infty$ in Step 1 of the X_1-representation algorithm II. So execution of the program terminates and the program does not output the X_1-representation of M.

For $z = 2$,

> su(2.0,89701,-9000,-900010,90301);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}$$

$$\begin{pmatrix}89701 & -9000\\ -3000 & 301\end{pmatrix}$$

> su(2.0,89701,-9000,-3000,301);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-30}$$

$$\begin{pmatrix}-299 & 30\\ -3000 & 301\end{pmatrix}$$

> su(2.0,-299,30,-3000,301);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{10}$$

$$\begin{pmatrix}-299 & 30\\ -10 & 1\end{pmatrix}$$

> su(2.0,-299,30,-10,1);

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{30}$$

$$\begin{pmatrix}1 & 0\\ -10 & 1\end{pmatrix}$$

> su(2.0,1,0,-10,1);

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}$$

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

The second matrix of the fifth execution of the program is the identity matrix which is $l = C^{-1}L = \begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{10}\begin{pmatrix}1 & 0\\ -10 & 1\end{pmatrix} = I$ in Step 3 of the X_1-representation algorithm II and so execution of the program terminates. Collect each first matrix in every execution of the program and concatenate them in order. Then we have

$$\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{-30}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{10}\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}^{30}\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}^{-10}$$

as the X_1-representation of M.

6.6 Correctness of Algorithm II

In this section, we justify the X_1-representation algorithm II and we also prove how the X_1-representation algorithm II works correctly. We prove several properties of the linear fractional transformations which will be used to prove the correctness of the X_1-representation algorithm II. Moreover, we explain why two X_1-representation algorithms are required to compute the X_1-representation of $M \in \Gamma_n$ from the following two Theorems. The following theorem is in fact, a special case of Lemma 4.1.2 and we prove it with another method for $z \in \mathbb{R} \cap D^c$.

Theorem 6.6.1 Let $n \geq 2$ and $z \in \mathbb{R} \cap D^c$. Then $|{B_n}^u(z)| < 1$ for a nonzero $u \in \mathbb{Z}$.

Proof Let $n \geq 2$ and $z \in \mathbb{R} \cap D^c$. Then $-1 < \frac{1}{z} < 1$ and $nu - 1 < nu + \frac{1}{z} < nu + 1$. If $u \geq 1$, then $nu - 1 \geq n - 1 \geq 1$ and so, $0 < \frac{1}{nu+\frac{1}{z}} < \frac{1}{nu-1} \leq 1$. Hence, ${B_n}^u(z) = \begin{pmatrix} 1 & 0 \\ nu & 1 \end{pmatrix}(z) = \frac{z}{nuz+1} = \frac{1}{nu+\frac{1}{z}} < 1$. If $u \leq -1$, then $nu+1 \leq -n+1 \leq -1$ and $-1 \leq \frac{1}{nu+1} < \frac{1}{nu+\frac{1}{z}} < 0$. Thus we have $|{B_n}^u(z)| < 1$. □

Theorem 6.6.2 Let $n \geq 3$ and $z \in \mathbb{R} \cap D^c$. Then $|{B_n}^u(z)| < \frac{1}{2}$ for a nonzero $u \in \mathbb{Z}$.

Proof Let $n \geq 3$ and $z \in D^c \cap \mathbb{R}$. Then $-1 < \frac{1}{z} < 1$ and $nu - 1 < nu + \frac{1}{z} < nu + 1$. If $u \geq 1$, then $nu - 1 \geq n - 1 \geq 2$ and $0 < \frac{1}{nu+\frac{1}{z}} < \frac{1}{nu-1} \leq \frac{1}{2}$. Hence, $0 < \frac{z}{nuz+1} = \frac{1}{nu+\frac{1}{z}} < \frac{1}{2}$. If $u \leq -1$, then $nu + 1 \leq -n + 1 \leq -2$ and $\frac{-1}{2} \leq \frac{1}{nu+1} < \frac{1}{nu+\frac{1}{z}} < 0$. Thus we have $|{B_n}^u(z)| < \frac{1}{2}$. □

In order to compute μ such that $\frac{-1}{2} < \mu \leq \frac{1}{2}$ in Step 1 of the X_1-representation algorithm II, Theorem 6.6.2 is required for $n \geq 3$ and so the case $n = 2$ is handled separately by the X_1-representation algorithm I. $\square$

Theorem 6.6.3 Let $n \geq 3$ and $z \in \mathbb{R}$ with $|z| < \frac{1}{2}$. Then $\frac{1}{|A_n{}^u(z)|} < \frac{2}{5}$ for a nonzero $u \in \mathbb{Z}$.

Proof Assume that $n \geq 3$ and $z \in \mathbb{R}$ such that $|z| < \frac{1}{2}$. Then by Lemma 4.1.1, $A_n{}^u(z) = nu + z \in D^c$ for a nonzero $u \in \mathbb{Z}$. For $-\frac{1}{2} < z < \frac{1}{2}$, $nu - \frac{1}{2} < nu + z < nu + \frac{1}{2}$. If $u \geq 1$, then $3 \leq nu$ and $3 - \frac{1}{2} \leq nu - \frac{1}{2} < nu + z < nu + \frac{1}{2}$, so that $0 < \frac{1}{nu+\frac{1}{2}} < \frac{1}{nu+z} < \frac{1}{nu-\frac{1}{2}} < \frac{1}{3-\frac{1}{2}} = \frac{2}{5}$. Hence, $0 < \frac{1}{A_n{}^u(z)} < \frac{2}{5}$. If $u \leq -1$, then $nu \leq -n \leq -3$ and $nu - \frac{1}{2} < nu + z < nu + \frac{1}{2} \leq -3 + \frac{1}{2} = -\frac{5}{2}$, so that $\frac{1}{-\frac{5}{2}} \leq \frac{1}{nu+\frac{1}{2}} < \frac{1}{nu+z} < \frac{1}{nu-\frac{1}{2}} < 0$. Hence, $-\frac{2}{5} < \frac{1}{A_n{}^u(z)} < 0$. Therefore $\frac{1}{|A_n{}^u(z)|} < \frac{2}{5}$. $\square$

From now, in particular, unless we mention a natural number n, n is a natural number, $n \geq 3$.

Theorem 6.6.4 If $M = A_n{}^u$ with a nonzero $u \in \mathbb{Z}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs $A_1{}^{nu}$ as the X_1-representation of M.

Proof If $M = A_n{}^u \in \Gamma_n$, then by Lemma 4.1.1, $|L(\frac{1}{2})| = |A_n{}^u(\frac{1}{2})| = |nu + \frac{1}{2}| > 1$ and so in Step 1 of the first iteration, $e = nu$ and $\mu = \frac{1}{2}$. In Step 2, $C = A_1{}^e = A_1{}^{nu}$, $w = wC = A_1{}^{nu}$, $L = C^{-1}L = A_1{}^{-nu}A_n{}^u = I$. So the algorithm outputs $A_1{}^{nu}$ as the X_1-representation of M and it terminates.

Theorem 6.6.5 If $M = A_n{}^u$ with a nonzero $u \in \mathbb{Z}$ is input to the algorithm ($z = 2$), then the algorithm outputs ϵ.

Proof If $M = {A_n}^u \in \Gamma_n$, then $L(2) = {A_n}^u(2) = nu + 2$.

If $n = 3$ and $u = -1$, then in Step 1 of the first iteration, $|L(2)| = |nu+2| = 1$. So the algorithm outputs ϵ and it terminates.

If $n \neq 3$ or $u \neq -1$, then in Step 1 of the first iteration, $|L(2)| = |nu+2| > 1$, $e = nu + 2$ and $\mu = 0$. In Step 2 of the first iteration, $C = {A_1}^e = {A_1}^{nu+2}$, $w = wC = {A_1}^{nu+2}$ and $L = C^{-1}L = {A_1}^{-nu-2}{A_n}^u = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix} \neq I$. So return Step 1. In Step 1 of the second iteration, $L(2) = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}(2) = 0$ and thus the algorithm outputs ϵ. Therefore the algorithm terminates. □

Theorem 6.6.6 If $M = {B_n}^u$ with a nonzero $u \in \mathbb{Z}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ.

Proof If $M = {B_n}^u \in \Gamma_n$, then $L(\frac{1}{2}) = {B_n}^u(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu+1} = \frac{1}{nu+2}$.

If $n = 3$ and $u = -1$, then in Step 1 of the first iteration, $|L(\frac{1}{2})| = |\frac{1}{nu+2}| = 1$. So the algorithm outputs ϵ and it terminates.

If $n \neq 3$ or $u \neq -1$, then in Step 1 of the first iteration, $|L(\frac{1}{2})| = |\frac{1}{nu+2}| < 1$ and $\frac{1}{L(\frac{1}{2})} = nu + 2$. So $e = \frac{1}{L(\frac{1}{2})} = nu + 2$ and $\mu = 0$. In Step 3 of the first iteration, $C = {B_1}^e = {B_1}^{nu+2}$, $w = wC = {B_1}^{nu+2}$ and $L = C^{-1}L = {B_1}^{-nu-2}{B_n}^u = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \neq I$. So return Step 1. In Step 1 of the second iteration, $L(\frac{1}{2}) = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix}(\frac{1}{2}) = \infty$. So the algorithm outputs ϵ and it terminates. □

Theorem 6.6.7 If $M = {B_n}^u$ is input to the algorithm ($z = 2$), then the algorithm outputs ${B_1}^{nu}$ as the X_1-representation of M.

Proof If $M = {B_n}^u \in \Gamma_n$, then in Step 1 of the first iteration, by Lemma 4.1.2, $|L(2)| = |{B_n}^u(2)| = |\frac{2}{2nu+1}| = |\frac{1}{nu+\frac{1}{2}}| < 1$ and $\frac{1}{L(2)} = nu + \frac{1}{2}$. Thus $e = nu$ and $\mu = \frac{1}{2}$. In Step 2 of the first iteration, $C = {B_1}^e = {B_1}^{nu}$, $w = wC = {B_1}^{nu}$,

$L = C^{-1}L = {B_1}^{-nu}{B_n}^{u} = I$. Hence, the algorithm outputs ${B_1}^{nu}$ as the X_1-representation of M and it terminates. □

Theorem 6.6.8 If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M where odd $m \geq 3$ and each u_i $(i = 1, 2, \cdots m)$ is a nonzero integer.

Proof If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$, then in Step 1 of the first iteration, by Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{n_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| > 1$. Put $L(\frac{1}{2}) = {A_n}^{n_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. Then by Lemma 4.1.1, ${A_n}^{u_3}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) \in D^c$ and by Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_2}({A_n}^{u_3}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$. So $e = nu_1$ and $\mu = \beta_1$. In Step 2 of the first iteration, $C = {A_1}^{e} = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{n_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, $L = C^{-1}L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 3 of the $i-1$th iteration, or $L = C^{-1}L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is even or odd.

For even i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration and $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $L(\frac{1}{2}) = {B_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) \in D$ and by Theorem 6.6.3, $|\frac{1}{\alpha_i}| < \frac{2}{5}$. So in Step 1 of the ith iteration, $\frac{1}{L(z)} = nu_i + \frac{1}{\alpha_i}$ and then $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3, $C = {B_1}^{e} = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$. So return Step 1.

For odd i, let $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration and $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. In Step 1 of the ith iteration, by Theorem 4.1,3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| > 1$ and ${A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in the mth iteration, $L = {A_n}^{u_m}$ and by Theorem 6.6.4, the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M. Thus the algorithm terminates. □

Theorem 6.6.9 If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm ($z = 2$), then the algorithm outputs ϵ where odd $m \geq 3$ and each u_i ($i = 1, 2, \cdots, m$) is a nonzero integer.

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$, put $L(2) = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. So in Step 1 of the first iteration, by Theorem 4.1.3, $L(2) = {A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$ and ${A_n}^{u_3}\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_3}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$. By Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_2}({A_n}^{u_3}\cdots{A_n}^{u_{m-2}}{B_n}^{u_{m-1}}{A_n}^{u_m}(2))| < \frac{1}{2}$. So $e = nu_1$ and $\mu = \beta_1$. In Step 2 of the first iteration, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq$

I. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m+2) \in D$. So in Step 1 of the first iteration, by Theorem 4.1.5, $|L(2)| = |{A_n}^{n_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}$ $(nu_m+2)| > 1$ and ${A_n}^{u_3} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_3} \cdots {B_n}^{u_{m-1}}(nu_m+2) \in D^c$. By Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_2}({A_n}^{u_3} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2))| < \frac{1}{2}$. So $e = nu_1$ and $\mu = \beta_1$. In Step 2 of the first iteration, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{n_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-2$, $L = C^{-1}L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = C^{-1}L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is even or odd.

For even i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration and $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i + 1} = \frac{1}{nu_i + \frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. So, in Step 1 of the ith iteration, by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})| < 1$ and by Theorem 4.1.3, ${A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+3}} \cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+2}} \cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})|$ $< \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)|} = \frac{1}{|{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})|}$ $< \frac{2}{5}$. Hence $\frac{1}{L(2)} = nu_i + \frac{1}{\alpha_i}$ and so, $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = B_i nu_i$, $w = wC = {B_i}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| < 1$ and by Theorem 4.1.5, ${A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}}(nu_m + 2) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_{i+2}}{A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)| = |{B_n}^{u_{i+2}}{A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}}(nu_m + 2)| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}} \ldots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)|} = \frac{1}{|{A_n}^{u_{i+1}} \ldots {B_n}^{u_{m-1}}(nu_m+2)|} < \frac{2}{5}$. So $\frac{1}{L(2)} = nu_i + \frac{1}{\alpha_i}$ and then $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration and $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.3, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})| > 1$ and ${A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+2}} \cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So, in Step 1 of the ith iteration, by Theorem 4.1.5, $|L(2)| = |{A_1}^{nu_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots$

${B_n}^{u_{m-1}}(nu_m+2)| > 1$ and ${A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}(nu_m+2) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_{i+1}}({A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ $= {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then in Step 1 of the $m-1$th iteration, $L = {B_n}^{u_{m-1}}{A_n}^{u_m}$ and consider $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. Since $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$, $\frac{1}{L(2)} = nu_{m-1} - 1$ and then $e = nu_{m-1} - 1$ and $\mu = 0$. In Step 3 of the $m-1$th iteration, $C = {B_1}^e = {B_1}^{nu_{m-1}-1}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}-1}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}+1}{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_1}^1{A_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = {B_1}^1{A_n}^{u_m}$ and $L(2) = {B_1}^1{A_n}^{u_m}(2) = {B_1}^1(-1) = \infty$. Hence the algorithm outputs ϵ and it terminates.

If $n \neq 3$ and $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}(nu_m + 2) = \frac{1}{nu_{m-1}+\frac{1}{nu_m+2}} \in D$. Since $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| < 1$, $\frac{1}{L(2)} = nu_{m-1}+\frac{1}{nu_m+2}$ and then $e = nu_{m-1}$ and $\mu = \frac{1}{nu_m+2}$. In Step 3 of the $m-1$th iteration, $C = {B_1}^e = {B_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = {A_n}^{u_m}$ and by Theorem 6.6.5, the algorithm outputs ϵ. Thus the algorithm terminates. □

Theorem 6.6.10 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2} \cdots {B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M where even $m \geq 2$ and each u_i ($i = 1, 2, \cdots, m$) is a nonzero integer.

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$, put $L(\frac{1}{2}) = {B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 n u_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| < 1$ and by Theorem 4.1.3, ${A_n}^{u_4} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_3} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_1|} = \frac{1}{|{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})|} < \frac{2}{5}$. Hence $\frac{1}{L(z)} = nu_1 + \frac{1}{\alpha_1}$ and then $e = nu_1$ and $\mu = \frac{1}{\alpha_1}$. In Step 3, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is odd or even.

For even i, in Step 1 if the ith iteration, $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ and put $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1,3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| > 1$ and ${A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) \in D^c$. So by Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}} ({A_n}^{u_{i+2}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$ and then $e = nu_i$ and $\mu = \beta_i$. In Step 2, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration and put $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i n u_i + 1} = \frac{1}{nu_i + \frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| < 1$ and ${A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2}) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_{i+2}}{A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(\frac{1}{2})|} < \frac{2}{5}$ and $\frac{1}{L(z)} = nu_i + \frac{1}{\alpha_i}$. So $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$.

In Step 3, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$. So return Step 1.

If $i = m$, then in the ith iteration, $L = {A_n}^{u_m}$ and in Step 2, by Theorem 6.6.4, the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2}\cdots {B_1}^{nu_{m-1}}{A_1}^{nu_m}$ as the X_1-representation of M. Thus the algorithm terminates. □

Theorem 6.6.11 If $M = {B_n}^{u_1}{A_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}$ is input to the algorithm ($z = 2$), then the algorithm outputs ϵ where even $m \geq 2$ and each u_i ($i = 1, 2, \cdots, m$) is a nonzero integer.

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \in \Gamma_n$, put $L(2) = {B_n}^{u_1}{A_n}^{u_2}\cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1+1} = \frac{1}{nu_1+\frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}\cdots {B_n}^{u_{m-1}}$ ${A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. So, in Step 1 of the first iteration, by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})| < 1$ and by Theorem 4.1.3, ${A_n}^{u_4}{B_n}^{u_5}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_4}{B_n}^{u_5}\cdots {A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_3}{A_n}^{u_4}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_3}{A_n}^{u_4}\cdots {A_n}^{u_{m-2}}$ $(\frac{1}{nu_{m-1}-1})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_1|} = \frac{1}{|{A_n}^{u_2}({B_n}^{u_3}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2))|} < \frac{2}{5}$ and $\frac{1}{L(2)} = nu_1 + \frac{1}{\alpha_1}$. So $e = nu_1$ and $\mu = \frac{1}{\alpha_1}$. In Step 3 of the first iteration, $C = {B_1}^e = B_1 nu_1$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots$ ${B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m+2) \in D$. So in Step 1 of the first iteration, by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots {B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_1}{A_n}^{u_2}\cdots {B_n}^{u_{m-1}}$

$(nu_m+2)| < 1$ and ${A_n}^{u_4}{B_n}^{u_5}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_4}{B_n}^{u_5}\cdots{B_n}^{u_{m-1}}(nu_m+2) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_3}{A_n}^{u_4}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_3}{A_n}^{u_4}\cdots{B_n}^{u_{m-1}}(nu_m+2)| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_1|}| = \frac{1}{|{A_n}^{u_2}({B_n}^{u_3}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2))|} < \frac{2}{5}$. So $e = nu_1$ and $\mu = \frac{1}{\alpha_1}$. In Step 3 of the first iteration, $C = {B_1}^e = B_1nu_1$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-2$, $L = C^{-1}L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = C^{-1}L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is odd or even.

For even i, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration and put $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.3, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})| > 1$ and ${A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+2}}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m+2) \in D$. So, in Step 1 of the ith iteration, by Theorem 4.1.5, $|L(2)| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}(nu_m+2)| > 1$ and ${A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}(nu_m+2) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}(nu_m+$

$2))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$ in Step 1 of the ith iteration, $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.4, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})| < 1$ and ${A_n}^{u_{i+3}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+3}}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1}) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_{i+2}}\cdots{A_n}^{u_{m-2}}(\frac{1}{nu_{m-1}-1})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}}({B_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2))|} < \frac{2}{5}$. So $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}$ ${B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{m-1}}(nu_m + 2) \in D$. So in Step 1 of the ith iteration, by Theorem 4.1.6, $|L(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots$ ${B_n}^{u_{m-1}}(nu_m+2)| < 1$ and ${A_n}^{u_{i+3}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {A_n}^{u_{i+3}}\cdots{B_n}^{u_{m-1}}(nu_m+2) \in D^c$. By Theorem 6.6.2, ${B_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2) = {B_n}^{u_{i+2}}\cdots{B_n}^{u_{m-1}}(nu_m+2) \in D$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}(2)|} = \frac{1}{|{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}(nu_m+2)|} < \frac{2}{5}$. So $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}C = {B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_{i+1}}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then in Step 1 of the $m-1$th iteration, $L = {B_n}^{u_{m-1}}{A_n}^{u_m}$ and consider $L(2) = {B_n}^{u_{m-1}}{A_n}^{u_m}(2)$.

If $n = 3$ and $u_m = -1$, then ${A_n}^{u_m}(2) = nu_m + 2 = -1$ and ${B_n}^{u_{m-1}}(-1) = \frac{1}{nu_{m-1}-1} \in D$. Since $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |\frac{1}{nu_{m-1}-1}| < 1$, $\frac{1}{L(2)} = nu_{m-1} - 1$ and then $e = nu_{m-1} - 1$ and $\mu = 0$. In Step 3 of the $m-1$th iteration, $C = {B_1}^e = {B_1}^{nu_{m-1}-1}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}-1}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}+1}{B_n}^{u_{m-1}}{A_n}^{u_m} = {B_1}^1{A_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = {B_1}^1{A_n}^{u_m}$ and $L(2) = {B_1}^1{A_n}^{u_m}(2) = {B_1}^1(-1) = \infty$. Hence the algorithm outputs ϵ and it terminates.

If $n \neq 3$ or $u_m \neq -1$, then ${A_n}^{u_m}(2) = nu_m + 2 \in D^c$ and by Lemma 4.1.2, ${B_n}^{u_{m-1}}(nu_m + 2) = \frac{1}{nu_{m-1}+\frac{1}{nu_m+2}} \in D$. Since $|L(2)| = |{B_n}^{u_{m-1}}{A_n}^{u_m}(2)| = |\frac{1}{nu_{m-1}+\frac{1}{nu_m+2}}| < 1$, $\frac{1}{L(2)} = nu_{m-1} + \frac{1}{nu_m+2}$ and as for $n = 4$ and $u_m = -1$, $|nu_m + 2| = 2$ is a minimum and $|\frac{1}{nu_m+2}| \leq \frac{1}{2}$, $e = nu_{m-1}$ and $\mu = \frac{1}{nu_m+2}$. In Step 3 of the $m-1$th iteration, $C = {B_1}^e = {B_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{A_1}^{nu_{m-2}}{B_1}^{nu_{m-1}}$ and $L = C^{-1}L = {B_1}^{-nu_{m-1}}{B_n}^{u_{m-1}}{A_n}^{u_m} = {A_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = {A_n}^{u_m}$ and by Theorem 6.6.5, the algorithm outputs ϵ. Thus the algorithm terminates. □

Theorem 6.6.12 If $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), the algorithm outputs ϵ where even $m \geq 2$ and each u_i ($i = 1, 2, \cdots, m$) is a nonzero integer.

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$, put $L(\frac{1}{2}) = {A_n}^{u_1}{B_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| = 1$ and as

$n \geq 3$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. In Step 1 of the first iteration, by Theorem 4.1.5, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| > 1$ and ${A_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_3} \cdots {B_n}^{u_{m-2}}(nu_{m-1}-1) \in D^c$. So by Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_2}({A_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$. Thus $e = nu_1$ and $\mu = \beta_1$. In Step 2 of the first iteration, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| < 1$. By Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| > 1$ and ${A_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_3} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. So by Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_2}({A_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$ and then $e = nu_1$ and $\mu = \beta_1$. In Step 2 of the first iteration, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$, $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-2$, $L = C^{-1}L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = C^{-1}L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is even or odd.

For even i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 1 of the ith iteration and $L(\frac{1}{2}) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}{B_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| = 1$ and as $n \geq 3$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(-1) = nu_{m-1}-1 \in D^c$. By Theorem 4.1.6, $|L(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {B_n}^{u_{m-2}}(nu_{m-1}-1)| < 1$ and ${A_n}^{u_{i+3}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{i+3}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1) \in D^c$. By Theorem 6.6.2, ${B_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_{i+2}} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1) \in D$ and by

Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|A_n{}^{u_{i+1}}\cdots B_n{}^{u_{m-2}}A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2})|} = \frac{1}{|A_n{}^{u_{i+1}}\cdots B_n{}^{u_{m-2}}(nu_{m-1}-1)|}$ $< \frac{2}{5}$. Then $\frac{1}{L(\frac{1}{2})} = nu_i + \frac{1}{\alpha_i}$ and so, $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = B_1{}^e = B_1{}^{nu_i}$, $w = wC = A_1{}^{nu_1}B_1{}^{nu_2}\cdots A_1{}^{nu_{i-1}}C = A_1{}^{nu_1}B_1{}^{nu_2}\cdots A_1{}^{nu_{i-1}}B_1{}^{nu_i}$, $L = C^{-1}L = B_1{}^{-nu_i}B_n{}^{u_i}A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m} = A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then $B_n{}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m+1} = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $A_n{}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |B_n{}^{u_i}A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}$ $B_n{}^{u_m}(\frac{1}{2})| = |B_n{}^{u_i}A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$ and $A_n{}^{u_{i+3}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{i+3}}\cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. By Theorem 6.6.2, $|B_n{}^{u_{i+2}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2})| = |B_n{}^{u_{i+2}}\cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|A_n{}^{u_{i+1}}\cdots B_n{}^{u_{m-2}}A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2})|}$ $= \frac{1}{|A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2})|} < \frac{2}{5}$. Then $\frac{1}{L(\frac{1}{2})} = nu_i + \frac{1}{\alpha_i}$ and so, $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = B_1{}^e = B_1{}^{nu_i}$, $w = wC = A_1{}^{nu_1}B_1{}^{nu_2}\cdots A_1{}^{nu_{i-1}}C = A_1{}^{nu_1}B_1{}^{nu_2}\cdots A_1{}^{nu_{i-1}}B_1{}^{nu_i}$ and $L = C^{-1}L = B_1{}^{-nu_i}$ $B_n{}^{u_i}A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m} = A_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m} \neq I$. So return Step 1.

For odd i, let $L(\frac{1}{2}) = A_n{}^{u_i}B_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = B_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then $|B_n{}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| = 1$ and as $n \geq 3$, $A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.5, $|L(\frac{1}{2})| = |A_n{}^{u_i}B_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2})| = |A_n{}^{u_i}B_n{}^{u_{i+1}}\cdots B_n{}^{u_{m-2}}(nu_{m-1} - 1)| > 1$ and $A_n{}^{u_{i+2}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{i+2}}\cdots B_n{}^{u_{m-2}}(nu_{m-1} - 1) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |B_n{}^{u_{i+1}}(A_n{}^{u_{i+2}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = A_1{}^e = A_1{}^{nu_i}$, $w = wC = A_1{}^{nu_1}B_1{}^{nu_2}\cdots B_1{}^{u_{i-1}}C = A_1{}^{nu_1}B_1{}^{nu_2}\cdots B_1{}^{u_{i-1}}A_1{}^{nu_i}$ and $L = C^{-1}L = A_1{}^{-nu_i}A_n{}^{u_i}B_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m} = B_n{}^{u_{i+1}}\cdots A_n{}^{u_{m-1}}B_n{}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| < 1$. By Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2})|$ > 1 and ${A_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. So by Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$. So $e = nu_i$ and $\mu = \beta$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{u_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{u_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}$ ${A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then $L = {A_n}^{u_{m-1}}{B_n}^{u_m}$ and consider $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and $|L(\frac{1}{2})| = |{A_n}^{u_{m-1}}{B_n}^{u_m}$ $(\frac{1}{2}) = {A_n}^{u_{m-1}}(-1)| = |nu_{m-1} - 1| > 1$. So in Step 1 of the $m-1$th iteration, $e = nu_{m-1} - 1$ and $\mu = 0$. In Step 2 of the $m-1$th iteration, $C = {A_1}^e = {A_1}^{nu_{m-1}-1}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}-1}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}+1}{A_n}^{u_{m-1}}{B_n}^{u_m} = A_1{B_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = A_1{B_n}^{u_m}$ and $L(\frac{1}{2}) = A_1{B_n}^{u_m}(\frac{1}{2}) = A_1(-1) = 0$. Hence the algorithm outputs ϵ and it terminates.

If $n \neq 3$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and $|L(\frac{1}{2})| = |{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})|$ $= |{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| = |nu_{m-1} + \frac{1}{nu_m+2}| > 1$. Because for $n = 4$ and $u_m = -1$, $|nu_m + 2| = 2$ is a minimum, $|\frac{1}{nu_m+2}| \leq \frac{1}{2}$, $e = nu_{m-1}$ and $\mu = \frac{1}{nu_m+2}$. In Step 2 of the $m-1$th iteration, $C = {A_1}^e = {A_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = {B_n}^{u_m}$ and by Theorem 6.6.6, the algorithm outputs ϵ and it terminates. □

Theorem 6.6.13 If $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = 2$), then the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the

X_1-representation of M where even $m \geq 2$ and each u_i $(i = 1, 2, \cdots, m)$ is a nonzero integer.

Proof Given $M = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$, put $L(2) = {A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = nu_1 + \beta_1$ where $\beta_1 = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| > 1$ and ${A_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) \in D^c$. So, by Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_2}({A_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2))| < \frac{1}{2}$ and then $e = nu_1$ and $\mu = \beta_1$. In Step 2 of the first iteration, $C = {A_1}^e = {A_1}^{nu_1}$, $w = wC = {A_1}^{nu_1}$ and $L = C^{-1}L = {A_1}^{-nu_1}{A_n}^{u_1}{B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is even or odd.

For even i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ in Step 1 of the ith iteration and $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i + 1} = \frac{1}{nu_i + \frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| < 1$ and ${B_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) \in D$. So by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}}({B_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2))|} < \frac{2}{5}$ and then $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ in Step 1 of the ith iteration and $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| > 1$ and ${A_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(2) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2))| < \frac{1}{2}$ and thus, $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e =$

${A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in the mth iteration, $L = {B_n}^{u_m}$ and by Theorem 6.6.7, in Step 3 of the mth iteration, the algorithm outputs ${A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M. Thus the algorithm terminates. □

Theorem 6.6.14 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = \frac{1}{2}$), then the algorithm outputs ϵ where odd $m \geq 3$ and each u_i ($i = 1, 2, \cdots, m$) is a nonzero integer.

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$, put $L(\frac{1}{2}) = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1 + 1} = \frac{1}{nu_1 + \frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}{B_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m + 1}| = |\frac{1}{nu_m + 2}| = 1$ and as $n \geq 3$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.6, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| < 1$ in Step 1 of the first iteration and by Theorem 4.1.5, ${A_n}^{u_4} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_4} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_3} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_3} \cdots {B_n}^{u_{m-2}}(nu_{m-1} - 1)| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_1|} = \frac{1}{|{A_n}^{u_2}({B_n}^{u_3} \cdots {B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}))|} < \frac{2}{5}$. Then $\frac{1}{L(\frac{1}{2})} = nu_1 + \frac{1}{\alpha_1}$ and so, $e = nu_1$ and $\mu = \frac{1}{\alpha_1}$. In Step 3 of the first iteration, $C = {B_1}^{e} = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$, $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m + 1} = \frac{1}{nu_m + 2} \in D$ and by Lemma 4.1.1, ${A_n}^{u_{m-1}}(\frac{1}{nu_m + 2}) \in D^c$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}$

${B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$ and by Theorem 4.1.3, ${A_n}^{u_4}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_4}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. By Theorem 6.6.2, $|{B_n}^{u_3}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{B_n}^{u_3}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_1|} = \frac{1}{|{A_n}^{u_2}({B_n}^{u_3}\cdots{B_n}^{u_{m-2}}{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}))|} < \frac{2}{5}$. Then $\frac{1}{L(\frac{1}{2})} = nu_1 + \frac{1}{\alpha_1}$ and because for $n = 4$ and $u_m = -1$, $|nu_m + 2| = 2$ is a minimum, $\frac{1}{|nu_m+2|} \leq \frac{1}{2}$. Thus $e = nu_1$ and $\mu = \frac{1}{\alpha_1}$. In Step 3 of the first iteration, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Suppose that for $1 \leq i-1 < m-2$, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is odd or even.

For even i, let $L(\frac{1}{2}) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| = 1$ and as $n \geq 3$, ${A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. By Theorem 4.1.5, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{B_n}^{u_{m-2}}(nu_{m-1} - 1)| > 1$ and ${A_n}^{u_{i+2}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{i+2}}\cdots{B_n}^{u_{m-2}}(nu_{m-1} - 1) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$ and so, $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{u_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{u_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then $|{B_n}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| < 1$. By Theorem 4.1.3, $|L(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})| = |{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots{A_n}^{u_{m-1}}(\frac{1}{nu_m+2})|$

> 1 and $A_n{}^{u_{i+2}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{i+2}} \cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. By Theorem 6.6.2, $|\beta_i| = |B_n{}^{u_{i+1}}(A_n{}^{u_{i+2}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}))| < \frac{1}{2}$ and so, $e = nu_i$ and $\mu = \beta_i$. In Step 2 of the ith iteration, $C = A_1{}^e = A_1{}^{nu_i}$, $w = wC = A_1{}^{nu_1} B_1{}^{nu_2} \cdots B_1{}^{nu_{i-1}} C = A_1{}^{nu_1} B_1{}^{nu_2} \cdots B_1{}^{nu_{i-1}} A_1{}^{nu_i}$ and $L = C^{-1}L = A_1{}^{-nu_i}$ $A_n{}^{u_i} B_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m} = B_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m} \neq I$. So return Step 1.

For odd i, let $L = B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m}$ and $L(\frac{1}{2}) = B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}}$ $B_n{}^{u_m}(\frac{1}{2}) = B_n{}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = A_n{}^{u_{i+1}} B_n{}^{u_{i+2}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m}$ $(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then in Step 1 of the ith iteration, $|B_n{}^{u_m}(\frac{1}{2})| = |\frac{\frac{1}{2}}{\frac{1}{2}nu_m+1}| = |\frac{1}{nu_m+2}| = 1$ and as $n \geq 3$, $A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{m-1}}(-1) = nu_{m-1}-1 \in D^c$. By Theorem 4.1.6, $|L(\frac{1}{2})| = |B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2})| = |B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots B_n{}^{u_{m-2}}(nu_{m-1}-1)| < 1$ and by Theorem 4.1.5, $A_n{}^{u_{i+3}} \cdots A_n{}^{u_{m-1}}$ $B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{i+3}} \cdots B_n{}^{u_{m-2}}(nu_{m-1} - 1) \in D^c$. By Theorem 6.6.2, $B_n{}^{u_{i+2}} \cdots$ $A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}) = B_n{}^{u_{i+2}} \cdots B_n{}^{u_{m-2}}(nu_{m-1} - 1) \in D$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|A_n{}^{u_{i+1}}(B_n{}^{u_{i+2}} \cdots B_n{}^{u_{m-2}} A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}))|} < \frac{2}{5}$. Then $\frac{1}{L(\frac{1}{2})} = nu_i + \frac{1}{\alpha_i}$ and so, $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 2 of the ith iteration, $C = B_1{}^e = B_1{}^{nu_i}$, $w = wC = A_1{}^{nu_1} B_1{}^{nu_2} \cdots A_1{}^{nu_{i-1}} C = A_1{}^{nu_1} B_1{}^{nu_2} \cdots A_1{}^{nu_{i-1}} B_1{}^{nu_i}$, $L = C^{-1}L = B_1{}^{-nu_i} B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m} = A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m} \neq I$. So return Step 1.

If $n \neq 3$ or $u_m \neq -1$, then $B_n{}^{u_m}(\frac{1}{2}) = \frac{\frac{1}{2}}{\frac{1}{2}nu_m+1} = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $A_n{}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. By Theorem 4.1.4, $|L(\frac{1}{2})| = |B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}}$ $B_n{}^{u_m}(\frac{1}{2})| = |B_n{}^{u_i} A_n{}^{u_{i+1}} \cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2})| < 1$ and by Theorem 4.1.3, $A_n{}^{u_{i+3}}$ $\cdots A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}) = A_n{}^{u_{i+3}} \cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2}) \in D^c$. By Theorem 6.6.2, $|B_n{}^{u_{i+2}} \cdots A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2})| = |B_n{}^{u_{i+2}} \cdots A_n{}^{u_{m-1}}(\frac{1}{nu_m+2})| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|A_n{}^{u_{i+1}}(B_n{}^{u_{i+2}} \cdots B_n{}^{u_{m-2}} A_n{}^{u_{m-1}} B_n{}^{u_m}(\frac{1}{2}))|} < \frac{2}{5}$. Then $\frac{1}{L(\frac{1}{2})} = nu_i + \frac{1}{\alpha_i}$ and thus, in Step 1 of the ith iteration, $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$, $L = C^{-1}L = {B_1}^{-nu_i}{B_n}^{u_i}{A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m-1$, then $L = {A_n}^{u_{m-1}}{B_n}^{u_m}$ and consider $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$.

If $n = 3$ and $u_m = -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} = -1$ and $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2})$ $= {A_n}^{u_{m-1}}(-1) = nu_{m-1} - 1 \in D^c$. So in Step 1 of the $m-1$th iteration, $e = nu_{m-1} - 1$ and $\mu = 0$. In Step 2 of the $m-1$th iteration, $C = {A_1}^e = {A_1}^{nu_{m-1}-1}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}-1}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}+1}{A_n}^{u_{m-1}}{B_n}^{u_m} = A_1{B_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = A_1{B_n}^{u_m}$ and $L(\frac{1}{2}) = A_1{B_n}^{u_m}(\frac{1}{2}) = A_1(-1) = 0$. Hence the algorithm outputs ϵ and it terminates.

If $n \neq 3$ or $u_m \neq -1$, then ${B_n}^{u_m}(\frac{1}{2}) = \frac{1}{nu_m+2} \in D$ and by Lemma 4.1.1, $L(\frac{1}{2}) = {A_n}^{u_{m-1}}{B_n}^{u_m}(\frac{1}{2}) = {A_n}^{u_{m-1}}(\frac{1}{nu_m+2}) = nu_{m-1} + \frac{1}{nu_m+2} \in D^c$. Since for $n = 4$ and $u_m = -1$, $|nu_m + 2| = 2$ is a minimum, $\frac{1}{|nu_m+2|} \leq \frac{1}{2}$. So in Step 1 of the $m-1$th iteration, $e = nu_{m-1}$ and $\mu = \frac{1}{nu_m+2}$. In Step 2 of the $m-1$th iteration, $C = {A_1}^e = {A_1}^{nu_{m-1}}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2} \cdots {B_1}^{nu_{m-2}}{A_1}^{nu_{m-1}}$ and $L = C^{-1}L = {A_1}^{-nu_{m-1}}{A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_m} \neq I$. So return Step 1. In Step 1 of the mth iteration, $L = {B_n}^{u_m}$ and by Theorem 6.6.6, the algorithm outputs ϵ. Thus the algorithm terminates. □

Theorem 6.6.15 If $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ is input to the algorithm ($z = 2$), then the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2} \cdots {A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M where odd $m \geq 3$ and each u_i $(i = 1, 2, \cdots, m)$ is a nonzero integer.

Proof Given $M = {B_n}^{u_1}{A_n}^{u_2} \cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$, put $L(2) = {B_n}^{u_1}{A_n}^{u_2} \cdots$

${A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_1}(\alpha_1) = \frac{\alpha_1}{\alpha_1 nu_1+1} = \frac{1}{nu_1+\frac{1}{\alpha_1}}$ where $\alpha_1 = {A_n}^{u_2}\cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| < 1$ and by Theorem 4.1.5, ${A_n}^{u_4}\cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(2) \in D$. By Theorem 6.6.2, ${B_n}^{u_3}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) \in D$ and by Theorem 6.6.3, $\frac{1}{|\alpha_1|} = \frac{1}{|{A_n}^{u_2}({B_n}^{u_3}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2))|} < \frac{2}{5}$. Thus $e = nu_1$ and $\mu = \frac{1}{\alpha_1}$. In Step 3 of the first iteration, $C = {B_1}^e = {B_1}^{nu_1}$, $w = wC = {B_1}^{nu_1}$ and $L = C^{-1}L = {B_1}^{-nu_1}{B_n}^{u_1}{A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_2}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

Assume that for $1 \leq i-1 < m-1$, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 3 of the $i-1$th iteration or $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}$ in Step 2 of the $i-1$th iteration according as $i-1$ is odd or even.

For even i, $L = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$, put $L(2) = {A_n}^{u_i}{B_n}^{u_{i+1}}\cdots$ ${A_n}^{u_{m-1}}{B_n}^{u_m}(2) = nu_i + \beta_i$ where $\beta_i = {B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)$. By Theorem 4.1.5, $|L(2)| > 1$ and ${A_n}^{u_{i+2}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) \in D^c$. By Theorem 6.6.2, $|\beta_1| = |{B_n}^{u_{i+1}}({A_n}^{u_{i+2}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2))| < \frac{1}{2}$ and so, $e = nu_i$ and $\mu = \beta_i$ in Step 1 of the ith iteration. Then in Step 2 of the ith iteration, $C = {A_1}^e = {A_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots {B_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots {B_1}^{nu_{i-1}}{A_1}^{nu_i}$ and $L = C^{-1}L = {A_1}^{-nu_i}{A_n}^{u_i}{B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {B_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

For odd i, let $L = {B_n}^{u_i}{A_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \in \Gamma_n$ and put $L(2) = {B_n}^{u_i}{A_n}^{u_{i+1}}$ $\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2) = {B_n}^{u_i}(\alpha_i) = \frac{\alpha_i}{\alpha_i nu_i+1} = \frac{1}{nu_i+\frac{1}{\alpha_i}}$ where $\alpha_i = {A_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(2)$. By Theorem 4.1.6, $|L(2)| < 1$ and by Theorem 4.1.5, ${A_n}^{u_{i+3}}\cdots {A_n}^{u_{m-1}}$ ${B_n}^{u_m}(2) \in D$. By Theorem 6.6.2, $|{B_n}^{u_{i+2}}{A_n}^{u_{i+3}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2)| < \frac{1}{2}$ and by Theorem 6.6.3, $\frac{1}{|\alpha_i|} = \frac{1}{|{A_n}^{u_{i+1}}({B_n}^{u_{i+2}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m}(2))|} < \frac{2}{5}$. So $e = nu_i$ and $\mu = \frac{1}{\alpha_i}$. In Step 3 of the ith iteration, $C = {B_1}^e = {B_1}^{nu_i}$, $w = wC = {A_1}^{nu_1}{B_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}C = {A_1}^{nu_1}{B_1}^{nu_2}\cdots {A_1}^{nu_{i-1}}{B_1}^{nu_i}$ and $L = C^{-1}L = {B_1}^{-nu_i}$ ${B_n}^{u_i}{A_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} = {A_n}^{u_{i+1}}\cdots {A_n}^{u_{m-1}}{B_n}^{u_m} \neq I$. So return Step 1.

If $i = m$, then in the mth iteration, $L = {B_n}^{u_m}$ and by Theorem 6.6.7, the algorithm outputs ${B_1}^{nu_1}{A_1}^{nu_2}\cdots{A_1}^{nu_{m-1}}{B_1}^{nu_m}$ as the X_1-representation of M in Step 3 of the mth iteration. Thus the algorithm terminates. □

Chapter 7

Homomorphic Public-Key Cryptosystem

Grigoriev and Ponomarenko [7] proposed a new homomorphic public-key cryptosystem over an arbitrary finite group based on the difficulty of the membership problem for groups of integer matrices. This scheme is a probabilistic public-key scheme and a homomorphic public-key scheme with a homomorphic property which comes from the group homomorphism. Homomorphic public-key schemes are proven to be useful in many cryptographic protocols such as electronic elections, computing and data delegations, protecting mobile agents and so on. [7]. Related previous work includes two probabilistic public-key schemes based on computations in the group $\mathrm{SL}_2(\mathbb{Z})$ which are not homomorphic schemes [23, 24] and they were already broken [1, 20]. There is another homomorphic public-key cryptosystem [6] over an arbitrary finite group, but its security is related to the intractability of integer factoring.

In this chapter, we describe Grigoriev and Ponomarenko homomorphic public-key cryptosystem and we analyze key generation algorithm, encryption algorithm and decryption algorithm from a practical point of view. Because description of Grigoriev and Ponomarenko homomorphic public-key scheme is very vague, it is necessary to do much more detailed and clear analysis on this homomorphic public-key scheme. Then encryption scheme and decryption scheme are justified. In addition, we show an example to demonstrate its implementation for practical applications and we compare Grigoriev and Ponomarenko' description with our description.

7.1 Description

We introduce Grigoriev and Ponomarenko homomorphic public-key cryptosystem. For the time being, we ignore practical implementation issues, which we consider in Section 7.2.

7.1.1 Setting Up The Scheme

The message space is given as a finite presentation $\langle X|\Re\rangle$ of a nontrivial finite group H where $X = \{x_1, x_2, \cdots, x_t\}$ is a set of generators with $t \geq 2$ and $\Re = \{w_1, w_2, \cdots, w_m\}$ is a set of relations. Let F be a free group generated by X and N be the normal closure of $\Re$. Then $H = F/N$. The set $\Re$ defines an equivalence relation $\equiv$ defined by $w_1 \equiv w_2$ iff $w_1{w_2}^{-1} \in N$ where w_1 and w_2 are words in $X^{\pm}$ and each equivalence class corresponds to a group element of H.

Let n be a natural number with $n \geq 2$ and $S = \{s_1, s_2, \cdots, s_t\}$ be a set of integers. Let

$$\phi : F \to G(n, S)$$

be an isomorphism such that for each $i = 1, \cdots, t$,

$$\phi(x_i) = M_i$$

where $x_i \in X$ and $M_i \in X(n, S)$. Randomly choose words $r_1, r_2, \cdots, r_t \in N$ and let $R = \{r_1, r_2, \cdots, r_t\}$. Define words $y_1, y_2, \cdots, y_t$ by

$$y_i = x_i r_i \ (i = 1, 2, \cdots, t)$$

where $x_i \in X$ and $r_i \in R$. Also, define matrices $Y_1, Y_2, \cdots, Y_t \in G(n, S)$ by

$$\begin{aligned} Y_i &= \phi(x_i r_i) \\ &= \phi(y_i). \end{aligned}$$

for $i = 1, 2, \cdots, t$. Let $G = \langle Y_1, Y_2, \cdots, Y_t \rangle$. Then G is a subgroup of $G(n, S)$ generated by the matrices $Y_1, Y_2, \cdots, Y_t$ and it is the ciphertext space of Grigoriev and Ponomarenko homomorphic public-key scheme. By Theorem 3.1.8, G is a free group as a subgroup of the free group $G(n, S)$, but the set $\{Y_1, Y_2, \cdots, Y_t\}$ is not necessarily a free basis.

7.1.2 The Keys

The public key is $\{Y_1, Y_2, \cdots, Y_t\}$ and the secret key consists of n and S.

7.1.3 The Scheme Itself

To encrypt a given message $h \in H$, let

$$x_{a_1}{}^{\epsilon_1} x_{a_2}{}^{\epsilon_2} \cdots x_{a_u}{}^{\epsilon_u}$$

be a representative of $h \in H$ where $a_i \in \{1, 2, \cdots, t\}$ and $\epsilon_i \in \{1, -1\}$. At random choose a word $r \in N$, write

$$r = x_{b_1}{}^{\delta_1} x_{b_2}{}^{\delta_2} \cdots x_{b_v}{}^{\delta_v}$$

where $b_i \in \{1, 2, \cdots, t\}$ and $\delta_i \in \{1, -1\}$. Define two matrices M_r and M_h by

$$M_r = Y_{b_1}{}^{\delta_1} Y_{b_2}{}^{\delta_2} \cdots Y_{b_v}{}^{\delta_v}$$

$$M_h = Y_{a_1}{}^{\epsilon_1} Y_{a_2}{}^{\epsilon_2} \cdots Y_{a_u}{}^{\epsilon_u}$$

where $x_{a_i} \mapsto Y_{a_i}$ and $x_{b_i} \mapsto Y_{b_i}$.
Let a matrix M be

$$M = M_r M_h$$

and then $E(h) = M$. The matrix M is the ciphertext of h.

To decrypt the ciphertext $M \in \mathrm{SL}_2(\mathbb{Z})$, express M as a word in $X(n,S)^{\pm}$ by the X_n-representation algorithm and the $X(n,S)$-representation algorithm, and write

$$M = {M_{c_1}}^{\gamma_1} {M_{c_2}}^{\gamma_2} \cdots {M_{c_w}}^{\gamma_w}$$

where $c_i \in \{1, 2, \cdots, t\}$ and $\gamma_i \in \{1, -1\}$. Let

$${x_{c_1}}^{\gamma_1} {x_{c_2}}^{\gamma_2} \cdots {x_{c_w}}^{\gamma_w}$$

be its corresponding word in F which represents $h \in H$ by $x_{c_i} \mapsto M_{c_i}$. Then $D(M) = h$ as the plaintext.

7.2 Key Generation in Practice

This section is related to implementation of Grigoriev and Ponomarenko homomorphic public-key cryptosystem. In practice, we analyze the cryptosystem in terms of a security parameter k.

Many of the methods in computational group theory depend on whether the group is represented as a group of permutations, a group of matrices, or by means of a presentation using generators and relations. So far the greatest success in computational group theory has come in connection with permutation groups on finite sets, finite solvable groups, and finitely presented groups. From the viewpoint of computational group theory, Grigoriev and Ponomarenko used a finitely presented group as the message space and a group of matrices as the ciphertext space. There are three methods commonly used to represent groups on a computer, namely, as groups of permutations of a finite set, groups of matrices over a ring, and as groups defined by a finite presentation. In this cryptosystem, we represent groups either as matrices or using generators and relations.

7.2.1 Construction of Message Space H

For a finite group H, a fixed finite presentation $\langle X|\Re\rangle$ is given where a set of generators, $X = \{x_1, x_2, \cdots, x_t\}$ with $t \geq 2$ and a set of relations, $\Re = \{w_1, w_2, \cdots, w_m\}$. Since the finite presentation of H and in particular, the cardinality t of the generating set X do not depend on the security parameter k. We choose in some way a concrete representation of each element $h \in H$, namely concrete representative presenting h. So we have one-to-one correspondence between the concrete representatives and elements of H. This set of representatives will be used to represent a plaintext in this homomorphic public-key scheme.

7.2.2 Generating Random Factors n, S and R

We discuss how in practice we choose the private key.

Choose at random a natural number $n \geq 2$ with $\ell(n) = k$ where $\ell(n)$ is the bit size of n.

Choose at random integers $s_1, s_2, \cdots, s_t \in \mathbb{Z}$ with $\ell(s_i) = k$ where $\ell(s_i)$ is the bit size of each integer s_i, write $S = \{s_1, s_2, \cdots, s_t\}$.

For each $i \in \{1, \cdots, t\}$, we do the following.

Choose $a_1, \cdots, a_k \in \{1, \cdots, m\}$ uniformly at random and set

$$r_i = w_{a_1} w_{a_2} \cdots w_{a_k}$$

where $w_{a_i} \in \Re^{\pm}$. Write $R = \{r_1, \cdots, r_t\}$. Note that the sum of bit sizes of n, S and R is $O(k)$.

7.2.3 Construction of Ciphertext Space G

For each generator $x_i (i = 1, 2, \cdots, t)$, the corresponding matrix Y_i is defined by $\phi(x_i r_i)$ where $x_i \in X$ and $r_i \in R$. The bit size $|Y_i|$ of a matrix Y_i is defined as the sum of bit sizes of the entries of Y_i. Note that for each $M_i =$

$\begin{pmatrix} 1-n^2s_i & -n^3{s_i}^2 \\ n & n^2s_i+1 \end{pmatrix} \in X(n,S)$, $\ell(M_i) = \ell(1-n^2s_i) + \ell(-n^3{s_i}^2) + \ell(n) + \ell(n^2s_i+1)$. Since Y_i is a product of the matrices $M_i \in X(n,S)$, $\ell(Y_i) = O(k)$.

7.3 Encryption in Practice

A message $h \in H$ is given by a concrete representative ${x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}\cdots{x_{a_u}}^{\epsilon_u}$ where $a_i \in \{1,2,\cdots,t\}$ and $\epsilon_i \in \{1,-1\}$.

To encrypt $h \in H$, the following steps are carried out.

Step 1 Obtain authentic public key $\{Y_1, Y_2, \cdots, Y_t\}$.

Step 2 Compute a matrix

$$M_h = {Y_{a_1}}^{\epsilon_1}{Y_{a_2}}^{\epsilon_2}\cdots{Y_{a_u}}^{\epsilon_u}$$

by corresponding $x_{a_i} \to Y_{a_i}$.

Step 3 Randomly choose a word $r \in N$ by randomly choosing $b_1, b_2, \cdots, b_k \in \{1,\cdots,m\}$ and defining

$$r = w_{b_1}w_{b_2}\cdots w_{b_k}.$$

Write

$$r = {x_{d_1}}^{\delta_1}{x_{d_2}}^{\delta_2}\cdots{x_{d_v}}^{\delta_v}$$

where $d_i \in \{1,\cdots,t\}$ and $\delta_i \in \{1,-1\}$.

Step 4 Compute a matrix

$$M_r = {Y_{d_1}}^{\delta_1}{Y_{d_2}}^{\delta_2}\cdots{Y_{d_v}}^{\delta_v}$$

by corresponding $x_{d_i} \to Y_{d_i}$.

Step 5 Compute a matrix $M = M_r M_h$.

Step 6 Output M as the ciphertext of the message h.

7.4 Decryption in Practice

To decrypt the ciphertext $M \in \mathrm{SL}_2(\mathbb{Z})$, the following steps are carried out.

Step 1 The X_n-representation algorithm computes the X_n-representation of M

Step 2 The $X(n, S)$-representation algorithm computes the $X(n, S)$-representation of M

$$M_{c_1}{}^{\gamma_1} M_{c_2}{}^{\gamma_2} \cdots M_{c_w}{}^{\gamma_w}$$

where $c_i \in \{1, 2, \cdots, t\}$ and $\gamma_i \in \{1, -1\}$.

Step 3 Find the group element $h \in H$ corresponding to the word in $X^{\pm}$

$$x_{c_1}{}^{\gamma_1} x_{c_2}{}^{\gamma_2} \cdots x_{c_w}{}^{\gamma_w}.$$

given by $x_{c_i} \mapsto M_{c_i}$.

Step 4 Output the group element h of H as the plaintext.

Note that the decrypted representative of $h \in H$ might not be the same as the original representative of h.

7.5 Justification

In this section, we justify the encryption and decryption schemes. That is to say, we show that for a given message, the two schemes work correctly to recover the plaintext from the ciphertext.

Correctness of Encryption

For a given message $h \in H$, let $E : H \to G$ be the encryption function and the encryption function is an injection $H \to G$. Then select an arbitrary representative ${x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}\cdots{x_{a_u}}^{\epsilon_u}$ corresponding to h and a random word $r = {x_{d_1}}^{\delta_1}{x_{d_2}}^{\delta_2}\cdots{x_{d_v}}^{\delta_v}$, and encrypt as follows :

$$\begin{aligned} E(h) &= E({x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}\cdots{x_{a_u}}^{\epsilon_u}) \\ &= M_r M_h \\ &= {Y_{d_1}}^{\delta_1}{Y_{d_2}}^{\delta_2}\cdots{Y_{d_v}}^{\delta_v}{Y_{a_1}}^{\epsilon_1}{Y_{a_2}}^{\epsilon_2}\cdots{Y_{a_u}}^{\epsilon_u} \\ &= M \end{aligned}$$

where Y_{a_i}, $Y_{d_i} \in \{Y_1, Y_2, \cdots, Y_t\}$, $a_i, d_i \in \{1, \cdots, t\}$ and $\epsilon_i, \delta_i \in \{-1, 1\}$.

Correctness of Decryption

By Theorem 3.1.9, there is an isomorphism $\phi : F \to G(n, S)$ and define an epimorphism $g \circ \phi^{-1} : G(n, S) \to F/N$ where ϕ^{-1} is the inverse of ϕ and $g : F \to F/N$ is a natural epimorphism. Then define the restriction map f of the epimorphism $g \circ \phi^{-1}$ by $f : G \to F/N$ and this restriction map f coincides with the decryption function. Note that the group H is fixed, but the epimorphism f and the group G depend on the security parameter k because the construction of f and G depends on our choice of n, S and R. Given the ciphertext M, let $D : G \to H$ be the decryption function. Then

$$\begin{aligned} D(M) &= D(E(h)) \\ &= D({M_{c_1}}^{\gamma_1}{M_{c_2}}^{\gamma_2}\cdots {M_{c_w}}^{\gamma_w}) \\ &= f({M_{c_1}}^{\gamma_1}{M_{c_2}}^{\gamma_2}\cdots {M_{c_w}}^{\gamma_w}) \\ &= g\circ\phi^{-1}({M_{c_1}}^{\gamma_1}{M_{c_2}}^{\gamma_2}\cdots {M_{c_w}}^{\gamma_w}) \\ &= g({x_{c_1}}^{\gamma_1}{x_{c_2}}^{\gamma_2}\cdots {x_{c_w}}^{\gamma_w}) \end{aligned}$$

But since $r_1, r_2, \cdots, r_t$ and $r \in N$,

$$g({x_{c_1}}^{\gamma_1}{x_{c_2}}^{\gamma_2}\cdots {x_{c_w}}^{\gamma_w}) = g({x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}\cdots {x_{a_u}}^{\epsilon_u}) = h$$

as

$$\begin{aligned} D(M) &= D(M_r M_h) \\ &= f(M_r M_h) \\ &= f(M_r)f(M_h) \\ &= f(M_h) \\ &= f({Y_{a_1}}^{\epsilon_1}{Y_{a_2}}^{\epsilon_2}\cdots {Y_{a_u}}^{\epsilon_u}) \\ &= g\circ\phi^{-1}({Y_{a_1}}^{\epsilon_1}{Y_{a_2}}^{\epsilon_2}\cdots {Y_{a_u}}^{\epsilon_u}) \\ &= g({x_{a_1}}^{\epsilon_1}{r_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}{r_{a_2}}^{\epsilon_2}\cdots {x_{a_u}}^{\epsilon_u}{r_{a_u}}^{\epsilon_u}) \\ &= g({x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}\cdots {x_{a_u}}^{\epsilon_u}) \\ &= h. \end{aligned}$$

7.6 Example

We give an example to show how Grigoriev and Ponomarenko homomorphic public-key cryptosystem works. The message space H is the dihedral group D_4 which consists of 4 reflections, 3 rotations and the identity transformation. For $i = 0, 1, 2, 3$ and $z \in \mathbb{C}$, the rotations are given by

$$z \;\to\; \omega^i z$$

and the reflections are given by

$$z \;\rightarrow\; \omega^i \bar{z}$$

where $\omega_i = e^{\frac{2\pi}{n}i}$. So the dihedral group D_4 is

$$\{\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}, \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}, \begin{pmatrix}-1 & 0\\ 0 & -1\end{pmatrix}, \begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix},$$

$$\begin{pmatrix}-1 & 0\\ 0 & 1\end{pmatrix}, \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \begin{pmatrix}0 & -1\\ -1 & 0\end{pmatrix}\}$$

and its finite presentation is $\langle X|\Re\rangle = \langle x_1, x_2 | {x_1}^2 = {x_2}^4 = 1, x_1x_2x_1 = {x_2}^{-1}\rangle$ where $X = \{x_1, x_2\}$ and $\Re = \{w_1 = {x_1}^2, w_2 = {x_2}^4, w_3 = (x_1x_2)^2\}$.

Given a security parameter $k = 2$, Bob chooses $n = 3$ with $\ell(3) = 2$ and $S = \{s_1, s_2\}$ with $s_1 = 2$, $s_2 = 3$ and $\ell(s_i) = 2$ $(i = 1, 2)$. Bob sets $X_3 = \{A_3, B_3\}$ and Bob constructs the group $\Gamma_3 = \langle X_3\rangle$ where $A_3 = \begin{pmatrix}1 & 3\\ 0 & 1\end{pmatrix}$ and $B_3 = \begin{pmatrix}1 & 0\\ 3 & 1\end{pmatrix}$. Let $X(n, S) = \{M_1, M_2\}$ where

$$M_1 = {A_3}^{-2}B_3{A_3}^2 = \begin{pmatrix}-17 & -108\\ 3 & 19\end{pmatrix}$$

$$M_2 = {A_3}^{-3}B_3{A_3}^3 = \begin{pmatrix}-26 & -243\\ 3 & 28\end{pmatrix}.$$

and the group $G(n, S) = \langle X(n, S)\rangle$. Bob generates random words $r_1 = w_1w_1$ and $r_2 = w_2w_1$, set $R = \{r_1, r_2\}$. Thus Bob generates the public key $\{Y_1, Y_2\}$ by

$$\begin{aligned} Y_1 &= \phi(x_1r_1)\\ &= \phi(x_1w_1w_1)\\ &= \phi(x_1x_1x_1x_1x_1)\\ &= M_1M_1M_1M_1M_1\\ &= {M_1}^5\\ &= \begin{pmatrix}-89 & -540\\ 15 & 91\end{pmatrix}\end{aligned}$$

$$
\begin{aligned}
Y_2 &= \phi(x_2r_2) \\
&= \phi(x_2w_2w_1) \\
&= \phi(x_2x_2x_2x_2x_2x_1x_1) \\
&= M_2M_2M_2M_2M_2M_1M_1 \\
&= {M_2}^5{M_1}^2 \\
&= \begin{pmatrix} -2600 & -16011 \\ 291 & 1792 \end{pmatrix}
\end{aligned}
$$

Encryption

Alice encrypts a message $h = x_2x_1$ by using Bob's public-key $\{Y_1, Y_2\}$.

For a message $h = x_2x_1$,

$$
\begin{aligned}
x_2r_2x_1r_1 &= x_2w_2w_1x_1w_1w_1 \\
&= x_2x_2x_2x_2x_2x_1x_1x_1x_1x_1x_1x_1 \\
&= {x_2}^5{x_1}^7
\end{aligned}
$$

and so Alice computes a matrix M_h corresponding to x_2x_1

$$
\begin{aligned}
M_h &= Y_2Y_1 \\
&= {M_2}^5{M_1}^7 \\
&= \begin{pmatrix} -2600 & -16011 \\ 291 & 1792 \end{pmatrix} \begin{pmatrix} -89 & -540 \\ 15 & 91 \end{pmatrix} \\
&= \begin{pmatrix} -8765 & -53001 \\ 981 & 5932 \end{pmatrix}.
\end{aligned}
$$

Alice generates a random word $r = w_1w_2 = x_1x_1x_2x_2x_2x_2 = {x_1}^2{x_2}^4$ in N and computes a matrix M_r

$$
\begin{aligned}
M_r &= Y_1Y_1Y_2Y_2Y_2Y_2 \\
&= {Y_1}^2{Y_2}^4 \\
&= {M_1}^{10}({M_2}^5{M_1}^2)^4 \\
&= \begin{pmatrix} -89 & -540 \\ 15 & 91 \end{pmatrix}^2 \begin{pmatrix} -2600 & -16011 \\ 291 & 1792 \end{pmatrix}^4
\end{aligned}
$$

$$= \begin{pmatrix} -79717571533043 & -490907171298024 \\ 13361344425294 & 82280225932885 \end{pmatrix}.$$

Then Alice computes a matrix $M = M_r M_h$ as the ciphertext

$$\begin{aligned} M &= M_r M_h \\ &= \begin{pmatrix} -79717571533043 & -490907171298024 \\ 13361344425294 & 82280225932885 \end{pmatrix} \begin{pmatrix} -8765 & -53001 \\ 981 & 5932 \end{pmatrix} \\ &= \begin{pmatrix} -9440101397885399 & -58132898241657525 \\ 1056557456928825 & 6506365190512276 \end{pmatrix}. \end{aligned}$$

Decryption

Given the ciphertext $M = \begin{pmatrix} -9440101397885399 & -58132898241657525 \\ 1056557456928825 & 6506365190512276 \end{pmatrix}$, Bob computes the X_n-representation of M, write

$$A_3{}^{-2}B_3{}^{10}A_3{}^{-1}B_3{}^{5}A_3B_3{}^{2}A_3{}^{-1}B_3{}^{5}A_3B_3{}^{2}A_3{}^{-1}B_3{}^{5}$$

$$A_3B_3{}^{2}A_3{}^{-1}B_3{}^{5}A_3B_3{}^{2}A_3{}^{-1}B_3{}^{5}A_3B_3{}^{7}A_3{}^{2}$$

and Bob computes the $X(n, S)$-representation of M, write

$$A_3{}^{-2}B_3{}^{10}A_3{}^{2}A_3{}^{-3}B_3{}^{5}A_3{}^{3}A_3{}^{-2}B_3{}^{2}A_3{}^{2}A_3{}^{-3}B_3{}^{5}A_3{}^{3}A_3{}^{-2}B_3{}^{2}A_3{}^{2}A_3{}^{-3}B_3{}^{5}$$

$$A_3{}^{3}A_3{}^{-2}B_3{}^{2}A_3{}^{2}A_3{}^{-3}B_3{}^{5}A_3{}^{3}A_3{}^{-2}B_3{}^{2}A_3{}^{2}A_3{}^{-3}B_3{}^{5}A_3{}^{3}A_3{}^{-2}B_3{}^{7}A_3{}^{2}$$

$$= M_1{}^{10}(M_2{}^{5}M_1{}^{2})^4 M_2{}^{5}M_1{}^{7}.$$

Bob finds a word in $X^{\pm}$ corresponding to the $X(n, S)$-representation of M, write

$$x_1{}^{10}x_2{}^{5}x_1{}^{2}x_2{}^{5}x_1{}^{2}x_2{}^{5}x_1{}^{2}x_2{}^{5}x_1{}^{2}x_2{}^{5}x_1{}^{7}$$

and computes the word again to obtain the normal form

$$\begin{aligned}
&{x_1}^{10}{x_2}^5{x_1}^2{x_2}^5{x_1}^2{x_2}^5{x_1}^2{x_2}^5{x_1}^2{x_2}^5{x_1}^7\\
&= 1x_21x_21x_21x_21x_2x_1\\
&= {x_2}^4x_2x_1\\
&= 1x_2x_1\\
&= x_2x_1
\end{aligned}$$

by using ${x_1}^2 = 1$ and ${x_2}^4 = 1$ and then output h corresponding to the normal form x_2x_1. Therefore, Bob recovers the plaintext h from the ciphertext M. □

7.7 Comparison

The description of Grigoriev and Ponomarenko homomorphic public-key cryptosystem is not concrete in [7] and in fact, they give only a theoretical idea to design a new homomorphic public-key cryptosystem over an arbitrary finite group. So we first clarify the description of Grigoriev and Ponomarenko homomorphic public-key cryptosystem so that key generation, encryption and decryption schemes work correctly in practice in terms of the security parameter k and so practical issues are discussed.

Grigoriev and Ponomarenko use a presentation $\langle X|\Re\rangle$ of the finite group H which is finitely generated as the message space whereas we have a finitely presented group as the message space because a finitely presented group is used to represent groups on a computer because there is no obvious way of representing $\Re$ on a computer unless $\Re$ is finite.

Grigoriev and Ponomarenko use bijections between X and $X(n, S)$ and between X and R in key generation scheme and decryption scheme. However, we describe them implicitly by the correspondences $x_i \mapsto M_i$ and $x_i \mapsto r_i$. Although we do not mention the bijections, we do not lose the generality. In fact, this simplifies the description of the public and secret keys. Moreover, in key generation algorithm, we provide precise algorithm to generate the random factors which Alice and Bob generate respectively in terms of the security parameter k from a practical point of view.

For encryption, Grigoriev and Ponomarenko represent the message h as a representative $x_{a_1}x_{a_2}\cdots x_{a_u}$ where $x_{a_i} \in X$, but they have not explained how they choose such a form and so, their scheme is not clear about how an element of H is represented. We describe the representative of h as ${x_{a_1}}^{\epsilon_1}{x_{a_2}}^{\epsilon_2}\cdots {x_{a_u}}^{\epsilon_u}$ to represent h where $x_{a_i} \in X$, $a_i \in \{1, \cdots, t\}$ and $\epsilon_i \in \{1, -1\}$ and thus, our scheme is more explicit about this.

In particular, the decryption scheme works in theory, but not in practice because they have not considered how Bob verifies that the representative ${x_{c_1}}^{\gamma_1}{x_{c_2}}^{\gamma_2}\cdots {x_{c_w}}^{\gamma_w}$ presents the plaintext h. In theory, the representative ${x_{c_1}}^{\gamma_1}{x_{c_2}}^{\gamma_2}\cdots {x_{c_w}}^{\gamma_w}$ must present the plaintext h, but in connection with its implementation, the length of the word in $X^{\pm}$ may be different from the original representative of the message h because some of letters can be canceled according as the choices of random words. So Grigoriev and Ponomarenko do not describe the method to obtain the plaintext h, but we clearly show how to obtain the plaintext by using a concrete representation of h and a normal form of h. In addition, we modified the X_n-representation algorithm in Chapter 4 and the $X(n, S)$-representation algorithm in Chapter 5 to make Grigoriev and Ponomarenko homomorphic public-key cryptosystem efficient. Therefore, through this chapter, we have made Grigoriev and Ponomarenko homomorphic public-key cryptosystem work correctly and efficient in practice.

Chapter 8

Cryptanalysis of A Homomorphic Public-Key Cryptosystem

In this chapter, it is shown how to break Grigoriev and Ponomarenko homomorphic public-key cryptosystem and so, it is proved that this new homomorphic public-key cryptosystem is vulnerable to our attacks. Given the public key $\{Y_1, Y_2, \cdots, Y_t\}$ and the ciphertext M, our task is to find the corresponding plaintext $h \in H$. Clearly we can do this if we find the secret key n and S. So, this chapter presents several attack methods to find the private key n and S and the attack method to recover the plaintext without knowing the private key n and S including each example to demonstrate how each attack method works and each attack method is written in a separate section.

In Section 8.1, we show the attack methods to compute n. In Section 8.1.1, we use the X_1-representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ and the ciphertext M, respectively to compute n. In Section 8.1.2, we show our experiment results to demonstrate how our attack methods are efficient. In Section 8.1.3, we compute n only by using the entries of the public key matrices $Y_1, Y_2, \cdots, Y_t$ or the ciphertext matrix M without using their X_1-representations.

In Section 8.2, because we require elements of S to decrypt the ciphertext M, we show the methods to compute elements of S.

In Section 8.3, in order to recover the plaintext, ordering the elements of S is also required to know one-to-one correspondence between X and $X(n, S)$. In fact, ordering the elements of S is implicitly secret and thus, we do exhaustive search for it.

In Section 8.4, we give an attack method to recover the plaintext without knowing the private key n and S. This attack method is to recover the original X_1-representation of $Y_i \in \{Y_1, Y_2, \cdots, Y_t\}$ from the partial X_1-representation of Y_i appearing in the X_1-representation of the ciphertext M because some of letters of the original X_1-representation of $Y_i \in \{Y_1, Y_2, \cdots, Y_t\}$ may be canceled. Therefore, this attack method demonstrates that knowing the private key is not always required to obtain the plaintext in Grigoriev and Ponomarenko homomorphic public-key scheme. So this attack method requires only the X_1-representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ and the ciphertext M.

In Section 8.5, we compare the attack methods above and summarize them.

8.1 Finding n

We first propose several attack methods to compute n by using the public key $\{Y_1, Y_2, \cdots, Y_t\}$ and the ciphertext M.

8.1.1 The X_1-Representations of Public Key and Ciphertext

Now we explain the attack method using the X_1-representations of Y_i or M. Remember that the main purpose of the X_1-representation algorithm is to break Grigoriev and Ponomarenko homomorphic public-key cryptosystem and so the X_1-representation algorithm is also one of parts for cryptanalysis of Grigoriev and Ponomarenko homomorphic public-key cryptosystem. In addition, as the ciphertext M is a product of the public-key matrices $Y_1, Y_2, \cdots, Y_t$, this attack method can be applied to the ciphertext M by the same way.

Let ${A_1}^{e_1}{B_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m}$ be the X_1-representation of Y_i with each nonzero integer $e_i = nu_i$. Since the exponents $e_1 = nu_1, e_2 = nu_2, \cdots, e_m = nu_m$ of the X_1-representation are multiples of n, n is one of divisors of the greatest common divisor of all exponents of the X_1-representation of Y_i. Moreover, as the ciphertext M is encrypted by the public-key matrices $Y_1, Y_2, \cdots, Y_t$, the X_1-representation of M also contain information about n. Therefore, the first attack method uses mainly the X_1-representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ or M to compute the private key n. In practice, the X_1-representation algorithm in Chapter 6 efficiently produces the X_1-representation of Y_i and then the following program made with Maple version 6 computes the greatest common divisor of $e_1 = nu_1, e_2 = nu_2, \cdots, e_m = nu_m$.

Computing GCD

```
> g:=proc(e1::integer, e2::integer, e3::integer, e4::integer, e5::integer)
> local g1, g2, g3, g4, g5;
> g1:=gcd(e1,e2);
> g2:=gcd(g1,e3);
> g3:=gcd(g2,e4);
> g4:=gcd(g3,e5);
> print(g4);
> end proc:
> g(e1, e2, e3, e4, e5);
```

This program is the case that the X_1-representation of Y_i is ${A_1}^{e_1}{B_1}^{e_2}{A_1}^{e_3}{B_1}^{e_4}{A_1}^{e_5}$ with each nonzero integer $e_i = nu_i$. In the program, $e1, e2, e3, e4$ and $e5$ indicate the exponents e_1, e_2, e_3, e_4 and e_5 of the X_1-representation of Y_i, respectively and $g4$ indicates the greatest common divisor of e_1, e_2, e_3, e_4 and e_5. In general, when we run the program to compute the greatest common divisor of the exponents $e_1, e_2, \cdots, e_m$ of the X_1-representation of Y_i, let d_i be their greatest common divisor. Then for $i = 1, \cdots, t$, d_i is the greatest common divisor of the exponents of each Y_i. We input $e1 = d_1, e2 = d_2, \cdots, et = d_t$ to the program and then the program outputs the greatest common divisor n' of $d_1, d_2, \cdots, d_t$. Therefore, the correct secret key n must be one of divisors of n'. We will also consider how likely n is n' with experiments and in practice we will show $n = n'$ in the following section 8.2. Note that this attack can be mounted on ciphertext M, rather than the public key matrices $Y_1, Y_2, \cdots, Y_t$ where the public key is not known.

Let $\langle X|\Re\rangle$ be any presentation. Let r be any element of N and let $\Re' = \Re \cup \{r\}$. Then it is clear that $\langle X|\Re\rangle$ and $\langle X|\Re'\rangle$ define isomorphic groups [14]. So we use $\langle X|\Re\rangle = \langle x_1, x_2|{x_1}^2 = 1, {x_2}^4 = 1, (x_1x_2)^4 = 1, (x_1x_2)^2 = 1\rangle$ as the finite presentation of the dihedral group D_4 instead of $\langle X|\Re\rangle = \langle x_1, x_2|{x_1}^2 = 1, {x_2}^4 = 1, (x_1x_2)^2 = 1\rangle$

Example 1

Given a finite presentation $\langle X|\Re\rangle = \langle x_1, x_2|{x_1}^2 = 1, {x_2}^4 = 1, (x_1x_2)^4 = 1, (x_1x_2)^2 = 1\rangle$ of the dihedral group D_4 where $X = \{x_1, x_2\}$ and $\Re = \{w_1 =$

${x_1}^2, w_2 = {x_2}^4, w_3 = (x_1x_2)^4, w_4 = (x_1x_2)^2\}$ and a security parameter $k = 3$, Bob chooses $n = 4$ and $S = \{s_1, s_2\}$ by $s_1 = 4$ and $s_2 = 6$, and generates random words $r_1 = w_1w_2w_3$ and $r_2 = w_2w_1w_1$, write $R = \{r_1, r_2\}$. Bob constructs the group $\Gamma_4 = \langle A_4, B_4\rangle$ by using his private key $n = 4$ where $A_4 = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}$ and $B_4 = \begin{pmatrix} 1 & 0 \\ 4 & 1 \end{pmatrix}$ and the group $G(n, S) = \langle M_1, M_2\rangle$ where $M_1 = {A_4}^{-4}B_4{A_4}^4 = \begin{pmatrix} -63 & -1024 \\ 4 & 65 \end{pmatrix}$ and $M_2 = {A_4}^{-6}B_4{A_4}^6 = \begin{pmatrix} -95 & -2304 \\ 4 & 97 \end{pmatrix}$. Thus Bob generates the public key $\{Y_1, Y_2\}$ by

$$\begin{aligned} Y_1 &= \phi(x_1r_1) \\ &= \phi(x_1w_1w_2w_3) \\ &= \phi({x_1}^3{x_2}^4(x_1x_2)^4) \\ &= {M_1}^3{M_2}^4(M_1M_2)^4 \\ &= \begin{pmatrix} 25583195842347841 & 620605483871411200 \\ -1607410491319748 & -38993086395179839 \end{pmatrix}. \end{aligned}$$

$$\begin{aligned} Y_2 &= \phi(x_2r_2) \\ &= \phi(x_2w_2w_1w_1) \\ &= \phi(x_2x_2x_2x_2x_2x_1x_1x_1x_1) \\ &= {M_2}^5{M_1}^4 \\ &= \begin{pmatrix} -62175 & -998656 \\ 2596 & 41697 \end{pmatrix}. \end{aligned}$$

Attack 1

For given the public key matrices Y_1 and Y_2, we use the X_1-representation algorithm to compute the private key n. The X_1-representation algorithm computes the X_1-representation of Y_1

$$ {A_1}^{-16}{B_1}^{12}{A_1}^{-8}{B_1}^{16}{A_1}^{8}{B_1}^{4}{A_1}^{-8}{B_1}^{4}{A_1}^{8}{B_1}^{4}{A_1}^{-8} $$

$$ {B_1}^{4}{A_1}^{8}{B_1}^{4}{A_1}^{-8}{B_1}^{4}{A_1}^{8}{B_1}^{4}{A_1}^{-8}{B_1}^{4}{A_1}^{24} $$

and the X_1-representation of Y_2

$$A_1{}^{-24}B_1{}^{20}A_1{}^{8}B_1{}^{16}A_1{}^{16}$$

where $A_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $B_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. Then
for Y_1, $d_1 = \gcd(-16, 12, -8, 16, 8, 4, -8, 4, 8, 4, -8) = 4$
for Y_2, $d_2 = \gcd(-24, 20, 8, 16, 16) = 4$
for both Y_1 and Y_2,

$$n' = \gcd(-16, 12, -8, 16, 8, 4, -8, 4, 8, 4, -8, -24, 20, 8, 16, 16) = 4.$$

Hence, we have the correct $n' = n = 4$. □

Example 2

As it is mentioned before, we can also use the ciphertext M to compute n.

Given a finite presentation $\langle X|\Re\rangle = \langle x_1, x_2 | x_1{}^2 = 1, x_2{}^4 = 1, (x_1x_2)^4 = 1, (x_1x_2)^2 = 1\rangle$ of the dihedral group D_4 where $X = \{x_1, x_2\}$ and $\Re = \{w_1 = x_1{}^2, w_2 = x_2{}^4, w_3 = (x_1x_2)^4, w_4 = (x_1x_2)^2\}$ and a security parameter $k = 2$, Bob chooses $n = 2$ and $S = \{s_1, s_2\}$ by $s_1 = 2$ and $s_2 = 3$, and generates random words by $r_1 = w_1w_1$ and $r_2 = w_2w_1$, write $R = \{r_1, r_2\}$. Bob constructs the group $\Gamma_2 = \langle A_2, B_2\rangle$ by using his private key $n = 2$ where $A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ and $B_2 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$ and the group $G(n, S) = \langle M_1, M_2\rangle$ where $M_1 = A_2{}^{-2}B_2A_2{}^2 = \begin{pmatrix} -7 & -32 \\ 2 & 9 \end{pmatrix}$ and $M_2 = A_2{}^{-3}B_2A_2{}^3 = \begin{pmatrix} -11 & -72 \\ 2 & 13 \end{pmatrix}$.
Then Bob generates the public key $\{Y_1, Y_2\}$ by

$$\begin{aligned} Y_1 &= \phi(x_1r_1) \\ &= \phi(x_1w_1w_1) \\ &= \phi(x_1{}^5) \\ &= M_1{}^5 \\ &= \begin{pmatrix} -39 & -160 \\ 10 & 41 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} Y_2 &= \phi(x_2 r_2) \\ &= \phi(x_2 w_2 w_1) \\ &= \phi({x_2}^5 {x_1}^2) \\ &= {M_2}^5 {M_1}^2 \\ &= \begin{pmatrix} -555 & -2344 \\ 94 & 397 \end{pmatrix}. \end{aligned}$$

Let $x_1 x_2$ be a concrete representative of a message h. To encrypt it, Alice computes a matrix M_h

$$\begin{aligned} M_h &= Y_1 Y_2 \\ &= {M_1}^5 {M_2}^5 {M_1}^2 \\ &= \begin{pmatrix} 6605 & 27896 \\ -1696 & -7163 \end{pmatrix} \end{aligned}$$

and Alice chooses the random word $r = {r_1}^{-1} = {w_1}^{-1}{w_1}^{-1} = {x_1}^{-4}$. Then Alice computes a matrix M_r

$$\begin{aligned} M_r &= {Y_1}^{-4} \\ &= {M_1}^{-20} \\ &= \begin{pmatrix} 161 & 640 \\ -40 & -159 \end{pmatrix}. \end{aligned}$$

Alice computes the ciphertext M by

$$\begin{aligned} M &= M_r M_h \\ &= \begin{pmatrix} 161 & 640 \\ -40 & -159 \end{pmatrix} \begin{pmatrix} 6605 & 27896 \\ -1696 & -7163 \end{pmatrix} \\ &= \begin{pmatrix} -22035 & -93064 \\ 5464 & 23077 \end{pmatrix}. \end{aligned}$$

Attack 2

Given the ciphertext M, we use the X_1-representation algorithm to compute n. The X_1-representation algorithm computes the X_1-representation of M

$$ {A_1}^{-4}{B_1}^{-30}{A_1}^{-2}{B_1}^{10}{A_1}^{2}{B_1}^{4}{A_1}^{4} $$

and we compute

$$\gcd(-4,-30,-2,10,2,4,4)=2$$

Hence, we have the correct $n = n' = 2$. □

8.1.2 Experiment Results

In this section, we do several experiments to demonstrate how efficiently our attack methods in Section 8.1.1 work. The greatest common divisor n' of the exponents of all the X_1- representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ is a multiple of n, but in practice, our experiments show we have $n' = n$. In order to convince the reader that $n' = n$, our implementations are given as follows.

Experiment 1

This experiment is to demonstrate the relation between the number of terms of the X_1-representation of $Y_i \in \{Y_1, Y_2, \cdots, Y_t\}$ or M and how likely n' is equal to n.

The idea comes from the following fact. Regardless of whatever the natural number $n \geq 2$ is, we only consider the integers $u_1, u_2, \cdots, u_m$ of the exponents of the X_1-representation ${A_1}^{nu_1}{B_1}^{nu_2}\cdots{B_1}^{nu_{m-1}}{A_1}^{nu_m}$ of Y_i because $n' = n$ means $\gcd(u_1, u_2, \cdots, u_m) = 1$ and $n < n'$ means $\gcd(u_1, u_2, \cdots, u_m) \neq 1$. We count the number of the X_1-representations with $n' = n$ and the number of the X_1-representations with $n' \neq n$. So, we can estimate how likely n' is equal to n. Note that we use the Maple version 6 to make programs for all experiments and our programming source codes are shown in Appendix.

Let the X_1-representation of $Y_i \in \{Y_1, Y_2, \cdots, Y_t\}$ or M be ${A_1}^{e_1}{B_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m}$

where $e_1 = nu_1, e_2 = nu_2, \cdots, e_{m-1} = nu_{m-1}, e_m = nu_m$. The experiment is carried out under the condition that the bit size $\ell(u_i)$ of the exponents of the X_1-representation of Y_i is 3 and the number of terms of the X_1-representation of Y_i is $m = 3, 5, 7$ and 9. Because for the X_1-representation with longer terms than 9, it takes quite a long time to do this experiment in reality when we run the program. So we do this experiment for several cases $m = 3, 5, 7$ and 9 and $u_i = 1, 2, 3, 4, 5, 6, 7$. For example, in case of $m = 5$, let the X_1-representation of Y_i be ${A_1}^{nu_1}{B_1}^{nu_2}{A_1}^{nu_3}{B_1}^{nu_4}{A_1}^{nu_5}$. Then we can consider totally 7^5 X_1-representations because each u_i have seven cases from 1 up to 7.

The experiment result shows that among the total 16807 X_1-representations, 16531 X_1-representations have the case $n' = n$ ($\gcd(u_1, u_2, u_3, u_4, u_5) = 1$) and 276 X_1-representations have the case $n < n'$ ($\gcd(u_1, u_2, u_3, u_4, u_5) \neq 1$). Therefore, we have the case $n' = n$ in most of the X_1-representations and we do not have the correct n directly in a few of the X_1-representations, but one of divisors of n' must be the correct n.

Table 8.1: Experiment Result 1

no. of terms	m=3	m=5	m=7	m=9
n$' = n$	329	16531	821227	40333411
n$' \neq n$	14	276	2316	20196
percentage of n$' = n$	95.9	98.4	99.7	99.9

The table shows that the percentage of $n' = n$ is getting close to 100 percent if the number of terms of the X_1-representation is getting larger. Therefore, it turns out that in practice, how our attack methods using the X_1-representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ is efficient to compute the secret key n. $\square$

Experiment 2

This experiment is to demonstrate the relation between the size of the integer exponents of the X_1-representation and how likely n' is equal to n.

Let us fix the X_1-representation ${A_1}^{nu_1}{B_1}^{nu_2}{A_1}^{nu_3}{B_1}^{nu_4}{A_1}^{nu_5}$ of Y_i with $m = 5$ and then we consider three cases, the bit size $\ell(u_i)$ of the exponents of the X_1-representation is 2, 3 and 4. Thus the range of u_i is $1 \leq u_i \leq 2^{\ell(u_i)} - 1$. Then we count the number of the X_1-representations with $n' = n$ and the number of the X_1-representations with $n' \neq n$ to see how likely n' is equal to n.

Let us see the case the X_1-representation ${A_1}^{nu_1}{B_1}^{nu_2}{A_1}^{nu_3}{B_1}^{nu_4}{A_1}^{nu_5}$ and $\ell(u_i) = 4$, that is, $u_i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 12, 14, 15$. Then among the total 759375 X_1-representations, 739201 X_1-representations have the case $n' = n$ and 20174 X_1-representations have $n' \neq n$.

Table 8.2: Experiment Result 2

$\ell(u_i)$	$\ell(u_i) = 2$	$\ell(u_i) = 3$	$\ell(u_i) = 4$
n$' = n$	241	16531	739201
n$' \neq n$	2	276	20174
percentage of n$' = n$	99.2	98.4	97.3

The table shows the percentage of $n' = n$ decreases when the bit sizes of the integer exponents increases. In other words, for very large integer exponents of the X_1-representation, we have less chance to have $n' = n$. □

Experiment 3

The difference between Experiment 2 and Experiment 3 is the size of u_i of the integer exponents of the X_1-representation of Y_i. Experiment 2 has been done for the cases $\ell(u_i) = 2, 3, 4$, whereas Experiment 3 is for the cases $1 \leq u_i \leq 2^\alpha$, $\alpha = 5, 10, 15, 20, 25, 30, 35$ and then we collect randomly 100 X_1-representations as random samples for each $1 \leq u_i \leq 2^\alpha$ to estimate how likely n' is equal to n. Among randomly chosen 100 X_1-representations, we only count the number of X_1-representations with $n' = n$. For instance, for $1 \leq u_i \leq 2^5$, there are 95 X_1-representations with $n' = n$ among random 100 X_1-representations. Therefore, Experiment 3 provides us more general information to know the relation between the size of the integer exponents and how likely n' is equal to n.

Table 8.3: Experiment Result 3

2^α	2^5	2^{10}	2^{15}	2^{20}	2^{25}	2^{30}	2^{35}
n$' = n$	95	95	92	95	94	93	95

From the table, we can see the average percentage of $n' = n$ is 94 percent. It means that even in case of the large integer exponents of the X_1-representation of Y_i, the average percentage of $n' = n$ is high as 94 percent although we have random choices of the X_1-representations. Therefore, it shows that using X_1-representations to compute the private key n is very efficient. □

Experiment 4

Let the X_1-representation of Y_i be ${A_1}^{nu_1}{B_1}^{nu_2}{A_1}^{nu_3}{B_1}^{nu_4}{A_1}^{nu_5}$ and the bit size $\ell(u_i) = 3$, that is, $u_i = 1, 2, 3, 4, 5, 6, 7$. Then in case of $n' \neq n$, we consider how n' is close to n. In this case, we consider all possible greatest common divisors, 2,3,4,5,6,7. The total number of the X_1-representations with $n \neq n'$

is 276 and for each possible greatest common divisor, we count the number of the X_1-representations among 276. We also compute the percentage of each case.

Table 8.4: Experiment Result 4

gcd	2	3	4	5	6	7
no. of X_1-rep.	241	31	1	1	1	1
percentage	87.3	11.2	0.4	0.4	0.4	0.4

Among 276 X_1-representations, 241 X_1-representations has the greatest common divisor 2 of u_1, u_2, u_3, u_4, u_5 and the percentage is about 87 percent. It means that for $n' \neq n$, the closest, that is, the largest divisor of n' which is not equal to n' must be n. □

Experiment 5

This experiment has been carried out by the same way as experiment 4 and the difference between them is that we do experiment for the bit size $\ell(u_i) = 4$ and so, all possible greatest common divisors of u_1, u_2, u_3, u_4, u_5 are 2,3,4,5,6,7,8,9,10,11,12,13,14 and 15. However, we have done our implementations for 2,3,4,5,6,7,8,9,10 and 11 because the program running time is very long.

Table 8.5: Experiment Result 5

gcd	2	3	4	5	6	7	8	9	10	11
no. of X_1-rep.	16531	3091	241	241	31	31	1	1	1	1
percentage	81.9	15.3	1.2	1.2	0.2	0.2	0.0	0.0	0.0	0.0

The table shows that apparently n is the closest integer to n' except for the case $n' = n$ because the case of the greatest common divisor 2 has the most

X_1-representations as 16531 X_1-representations and also the percentage is the largest percent, about 82 percent. Therefore, the largest divisor of n' with $n < n'$ is the correct n in the most of cases. $\square$

8.1.3 Observing Matrix Entries of Public Key and Ciphertext

In this section, we show that without using the X_1-representations of $Y_i \in \{Y_1, Y_2, \cdots, Y_t\}$ or M, we can also compute n only by calculating the greatest common divisor of all the entries of the matrix $Y_i \in \{Y_1, Y_2, \cdots, Y_t\}$ or M.

This attack method needs some properties of elements of the group Γ_n. Because $Y_i \in \{Y_1, \cdots, Y_t\}$ or M is in the group $G(n, S)$ generated by $X(n, S)$ and $G(n, S)$ is a subgroup of Γ_n, $Y_i \in \{Y_1, \cdots, Y_t\}$ and M are in Γ_n. Since exponents of the X_n-representations of elements of Γ_n have the common divisor n, matrices of Γ_n may leak information about n. Therefore, we prove some properties of the group Γ_n from our observation on Γ_n and we demonstrate this attack method.

Theorem 8.1.3.1 For $n \geq 2$, let $M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \in \Gamma_n$. Then n divides $M_{11} - 1, M_{12}, M_{21}$ and $M_{22} - 1$.

Proof Let $n \geq 2$ be a natural number. Since $M \in \Gamma_n$, M has the X_n-representation. We prove it by the induction on the number m of terms of the X_n-representation of M.

For $m = 1$, M has only one term in its X_n-representation. There are two cases : one is $M = {A_n}^u$ and the other is $M = {B_n}^u$.
If $M = {A_n}^u$, then $M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} = \begin{pmatrix} 1 & nu \\ 0 & 1 \end{pmatrix}$ and thus, n divides

$M_{11} - 1 = 0, M_{12} = nu, M_{21} = 0$ and $M_{22} - 1 = 0$. Hence, the theorem follows in this case.

If $M = {B_n}^u$, then $M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ nu & 1 \end{pmatrix}$ and n divides $M_{11} - 1 = 0, M_{12} = 0, M_{21} = nu$ and $M_{22} - 1 = 0$. Thus, the theorem follows in this case as well.

For $m \geq 1$, as the inductive hypothesis, we assume that any matrix $M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}$ having the X_n-representation with m terms has the property that n divides $M_{11} - 1, M_{12}, M_{21}$ and $M_{22} - 1$. Let $M' = \begin{pmatrix} {M_{11}}' & {M_{12}}' \\ {M_{21}}' & {M_{22}}' \end{pmatrix}$ have the X_n-representation with $m + 1$ terms. Then the X_n-representation of M' is one of the following forms :

$$\begin{gathered} {A_n}^{u_1} {B_n}^{u_2} \cdots {B_n}^{u_{m-1}} {A_n}^{u_m} {B_n}^{u_{m+1}} \\ {A_n}^{u_1} {B_n}^{u_2} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m} {A_n}^{u_{m+1}} \\ {B_n}^{u_1} {A_n}^{u_2} \cdots {A_n}^{u_{m-1}} {A_n}^{u_m} {B_n}^{u_{m+1}} \\ {B_n}^{u_1} {A_n}^{u_2} \cdots {A_n}^{u_{m-1}} {B_n}^{u_m} {A_n}^{u_{m+1}}. \end{gathered}$$

Simply we have either $M' = M{A_n}^{u_{m+1}}$ or $M' = M{B_n}^{u_{m+1}}$ where M has the X_n-representation with m terms.

If $M' = M{A_n}^{u_{m+1}}$, then

$$\begin{pmatrix} {M_{11}}' & {M_{12}}' \\ {M_{21}}' & {M_{22}}' \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} 1 & nu_{m+1} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} M_{11} & M_{11}nu_{m+1} + M_{12} \\ M_{21} & M_{21}nu_{m+1} + M_{22} \end{pmatrix}.$$

By the inductive hypothesis that n divides $M_{11} - 1, M_{12}, M_{21}, M_{22} - 1$, clearly n divides ${M_{11}}' - 1 = M_{11} - 1$, ${M_{12}}' = M_{11}nu_{m+1} + M_{12}$, ${M_{21}}' = M_{21}$ and ${M_{22}}' - 1 = M_{21}nu_{m+1} + (M_{22} - 1)$.

If $M' = M{B_n}^{u_{m+1}}$, then

$$\begin{pmatrix} {M_{11}}' & {M_{12}}' \\ {M_{21}}' & {M_{22}}' \end{pmatrix} = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ nu_{m+1} & 1 \end{pmatrix} = \begin{pmatrix} M_{11} + M_{12}nu_{m+1} & M_{12} \\ M_{21} + M_{22}nu_{m+1} & M_{22} \end{pmatrix}.$$

By the inductive hypothesis that n divides $M_{11} - 1, M_{12}, M_{21}$ and $M_{22} - 1$, ${M_{12}}' = M_{12}$ and ${M_{22}}' = M_{22} - 1$, n divides $M'_{11} - 1 = (M_{11} - 1) + M_{12}nu_{m+1}$,

$M'_{12} = M_{12}$, $M'_{21} = M_{21} + M_{22}nu_{m+1}$ and $M_{22}{}' - 1 = M_{22} - 1$. Therefore, the theorem follows by the induction. □

Corollary 8.1.3.2 Let $n \geq 2$ and $M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \in \Gamma_n$. Then n is a divisor of the greatest common divisor of $M_{11} - 1, M_{12}, M_{21}$ and $M_{22} - 1$.

Proof It is trivially proved by Theorem 8.1.3.1.

Corollary 8.1.3.2 implies that the matrix $Y_i \in \{Y_1, \cdots, Y_t\}$ and the ciphertext M have information about the secret key n. For the public key matrix $Y_i = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \in \{Y_1, Y_2, \cdots, Y_t\}$, we compute

$$d_i = \gcd(Y_{i11} - 1, Y_{i12}, Y_{i21}, Y_{i22} - 1)$$

and for all the public-key matrices $Y_1, Y_2, \cdots, Y_t$, let

$$n' = \gcd(d_1, d_2, \cdots, d_t)$$

where each $i = 1, 2, \cdots, t$, d_i corresponds to Y_i. Then the secret key n must be a divisor of n'. Similarly, for the ciphertext $M = \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix} \in \Gamma_n$, we compute

$$n' = \gcd(M_{11} - 1, M_{12}, M_{21}, M_{22} - 1)$$

and then the private n must be one of divisors of n'.

Now, we consider how n is close to n' by comparing the case of calculating the greatest common divisor of all the entries of the public key matrices $Y_1, Y_2, \cdots, Y_t$ and the case of calculating the greatest common divisor of the entries of the ciphertext M in the following examples. It is shown that the greatest common divisor n' of the entries of the public key matrices is more likely to be n. We apply these attack methods to Example 2 in Section 8.1.1.

Example 1

The public key matrices $Y_1 = \begin{pmatrix} -39 & -160 \\ 10 & 41 \end{pmatrix}$ and $Y_2 = \begin{pmatrix} -555 & -2344 \\ 94 & 397 \end{pmatrix}$ are given.

Attack 3

We compute for Y_1,

$$\gcd(-39-1, -160, 10, 41-1) = 10$$

and for Y_2,

$$\gcd(-555-1, -2344, 94, 397-1) = 2.$$

Therefore, the greatest common divisor n' of all the entries of the public key matrices Y_1 and Y_2 is

$$n' = \gcd(10, 2) = 2.$$

Thus we have the correct $n' = n = 2$.

Example 2

The ciphertext $M = \begin{pmatrix} -22035 & -93064 \\ 5464 & 23077 \end{pmatrix}$ is given.

Attack 4

We compute

$$n' = \gcd(-22035-1, -93064, 5464, 23077-1) = 4.$$

Hence, we obtain $n' = 4$ and in fact, the correct $n = 2$ is a divisor of $n' = 4$.

8.2 Finding S

Assume that we have the private key n by using one of techniques in Section 8.1. We now try to find elements $s_1, s_2, \cdots, s_t$ of S as the other part of the private key. We present two ways to collect the elements of S. One is to use the X_1-representation algorithm and the other is to use the X_n-representation algorithm. The difference between them is whether the X_n-representation of the ciphertext M is obtained by the X_1-representation algorithm or the X_n-representation algorithm.

We explain the first method to compute the X_n-representation of Y_i or M. The X_1-representation algorithm takes the public key matrix Y_i or the ciphertext M as an input and then it outputs the X_1-representation of M. Let the X_1-representation of M be ${A_1}^{e_1}{B_1}^{e_2}\cdots{B_1}^{e_{m-1}}{A_1}^{e_m}$ with $e_i = nu_i$. Since n is revealed by our attacks, we can compute each nonzero u_i when we divide e_i by n and so we can obtain the X_n-representation ${A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$.

As the second method to obtain the X_n-representation of Y_i or M, the X_n-representation algorithm takes n and the public key matrix Y_i(or the ciphertext M) as two inputs and it outputs the X_n-representation ${A_n}^{u_1}{B_n}^{u_2}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}$.

As it is shown in Chapter 5, the X_n-representation can be written as

$$\begin{aligned}&{A_n}^{u_1}{B_n}^{u_2}{A_n}^{u_3}{B_n}^{u_4}\cdots{B_n}^{u_{m-1}}{A_n}^{u_m}\\ =&\mathrm{A}_n^{-s_{a_1}}{B_n}^{u_2}{A_n}^{s_{a_1}-s_{a_2}}{B_n}^{u_4}{A_n}^{s_{a_2}-s_{a_3}}\cdots{A_n}^{s_{a_{\frac{m-3}{2}}}-s_{a_{\frac{m-1}{2}}}}{B_n}^{u_{m-1}}{A_n}^{s_{a_{\frac{m-1}{2}}}}\end{aligned}$$

where $a_i \in \{1, 2, \cdots, t\}$, $s_{a_i} \in S$ and

$$u_1 = -s_{a_1}$$

$$u_3 = s_{a_1} - s_{a_2}$$

$$u_5 = s_{a_2} - s_{a_3}$$

$$\vdots$$

$$
\begin{aligned}
u_{2i-1} &= s_{a_{i-1}} - s_{a_i} \\
&\vdots \\
u_{m-2} &= s_{a_{\frac{m-3}{2}}} - s_{a_{\frac{m-1}{2}}} \\
u_m &= s_{a_{\frac{m-1}{2}}}.
\end{aligned}
$$

Note that $-u_1 = s_{a_1}$ is always an element of S and we can compute elements $s_{a_1}, s_{a_2}, \cdots, s_{a_{\frac{m-1}{2}}} \in S$ as follows :

$$
\begin{aligned}
s_{a_1} &= -u_1 \\
s_{a_2} &= -u_1 - u_3 \\
s_{a_3} &= -u_1 - u_3 - u_5 \\
&\vdots \\
s_{a_i} &= -u_1 - u_3 - u_5 - \cdots - u_{2_i-1} \\
&\vdots \\
s_{a_{\frac{m-1}{2}}} &= -u_1 - u_3 - u_5 - \cdots - u_{m-2}.
\end{aligned}
$$

Therefore, we can collect elements of S up to $\frac{m-1}{2}$. If we can not find all the elements of S from the public key matrices $Y_1, Y_2, \cdots, Y_t$ or the ciphertext matrix M, then we can use another attack method generating many ciphertexts to get more information about elements of S. However, in practice, the public key matrices $Y_1, Y_2, \cdots, Y_t$ seem to give enough information so that we can collect all the elements of S. The following example shows how this attack method works and we apply this attack method to Example 2 in Section 8.1.1.

Example

Two public key matrices $Y_1 = \begin{pmatrix} -39 & -160 \\ 10 & 41 \end{pmatrix}$, $Y_2 = \begin{pmatrix} -555 & -2344 \\ 94 & 397 \end{pmatrix}$ and the ciphertext matrix $M = \begin{pmatrix} -22035 & -93064 \\ 5464 & 23077 \end{pmatrix}$ are given.

Attack 5

By the X_1-representation algorithm, we have the X_1-representation of Y_1

$$A_1{}^{-4}B_1{}^{10}A_1{}^{4}$$

and the X_1-representation of Y_2

$$A_1{}^{-6}B_1{}^{10}A_1{}^{2}B_1{}^{4}A_1{}^{4}.$$

When we divide exponents of the X_1-representations of Y_1 and Y_2 by $n = 2$, we have the X_n-representation of Y_1

$$A_2{}^{-2}B_2{}^{5}A_2{}^{2}$$

and the X_n-representation of Y_2

$$A_2{}^{-3}B_2{}^{5}A_2B_2{}^{2}A_2{}^{2}.$$

Therefore, from the X_n-representation of Y_1, we have

$$s_{a_1} = -u_1 = -(-2) = 2$$

and from the X_n-representation of Y_2, we have

$$\begin{aligned} s_{a_1} &= -u_1 = -(-3) = 3 \\ s_{a_2} &= -u_1 - u_3 = 3 - 1 = 2. \end{aligned}$$

Hence, we can compute $S = \{2, 3\}$ from X_n-representations of the public key matrices Y_1 and Y_2.

Attack 6

For the ciphertext M, the X_1-representation algorithm computes the X_1-representation of M

$$A_1{}^{-4}B_1{}^{-30}A_1{}^{-2}B_1{}^{10}A_1{}^{2}B_1{}^{4}A_1{}^{4}.$$

and when we divide exponents of the X_1-representation of M by $n = 2$, we have the X_n-representation of M

$$A_2{}^{-2}B_2{}^{-15}A_2{}^{-1}B_2{}^{5}A_2B_2{}^{2}A_2{}^{2}.$$

So we have

$$\begin{aligned} s_{a_1} &= -u_1 = -(-2) = 2 \\ s_{a_2} &= -u_1 - u_3 = 2 - (-1) = 3 \\ s_{a_3} &= -u_1 - u_3 - u_5 = 3 - 1 = 2. \end{aligned}$$

and hence, we can compute $S = \{2, 3\}$ from the X_n-representation of the ciphertext M.

The preimage of the public key matrix Y_i is $x_i r_i$ where $x_i \in X$ and $r_i \in R$ with the length k (security parameter). It means that the choice of the security parameter k affects the number of the terms of the X_n-representation. Hence, as the security parameter k increases, the number m of the terms of the X_n-representation of Y_i goes up and the number m of the terms of the X_n-representation of the ciphertext M also grows because the ciphertext M is generated by the public key matrices $Y_1, Y_2, \cdots, Y_t$. It implies that the large security parameter k is likely to recover all the elements of S.

8.3 Ordering The Elements of S

Suppose that we have the private key n and S. Then we have to find S as an ordered set to recover the plaintext and there are $t!$ ways of ordering the elements of S because $|S| = t$. One of them must be correct ordering the elements of S. For each ordering, we can test if this ordering is correct by encrypting several plaintexts with the public key and then after decrypting it, we check whether the original plaintext is produced.

Example

We apply this method to Example 2 in Section 8.1.1. Let $n = 2$ and $S = \{2, 3\}$. Then we compute two matrices ${A_2}^{-2}B_2{A_2}^2 = \begin{pmatrix} -7 & -32 \\ 2 & 9 \end{pmatrix}$ and ${A_2}^{-3}B_2{A_2}^3 = \begin{pmatrix} -11 & -72 \\ 2 & 13 \end{pmatrix}$.

Attack 7

If the first ordering way is

$$M_1 = {A_2}^{-2}B_2{A_2}^2 = \begin{pmatrix} -7 & -32 \\ 2 & 9 \end{pmatrix}$$

and

$$M_2 = {A_2}^{-3}B_2{A_2}^3 = \begin{pmatrix} -11 & -72 \\ 2 & 13 \end{pmatrix},$$

then for $X = \{x_1, x_2\}$ and $X(n, S) = \{M_1, M_2\}$, $x_1 \mapsto M_1$ and $x_2 \mapsto M_2$. In order to recover the plaintext from the ciphertext $M = \begin{pmatrix} -22035 & -93064 \\ 5464 & 23077 \end{pmatrix}$, the X_1-representation algorithm computes the X_1-representation of M

$${A_1}^{-4}{B_1}^{-30}{A_1}^{-2}{B_1}^{10}{A_1}^2{B_1}^4{A_1}^4,$$

and the X_n-representation algorithm computes the X_n-representation of M by taking n as an input

$${A_2}^{-2}{B_2}^{-15}{A_2}^{-1}{B_2}^5A_2{B_2}^2{A_2}^2.$$

Then the $X(n, S)$-representation algorithm computes the $X(n, S)$-representation of M

$${A_2}^{-2}{B_2}^{-15}{A_2}^2{A_2}^{-3}{B_2}^5{A_2}^3{A_2}^{-2}{B_2}^2{A_2}^2$$

$$= {M_1}^{-15}{M_2}^5{M_1}^2$$

because elements of S are obtained by the attack methods in Section 8.2.

By $x_1 \mapsto M_1$ and $x_2 \mapsto M_2$, we can compute the representative ${x_1}^{-15}{x_2}^5{x_1}^2$ and by the relations ${x_1}^2 = 1$ and ${x_2}^4 = 1$,

$$\begin{aligned}&{x_1}^{-15}{x_2}^5{x_1}^2\\ &= {x_1}^{-1}x_2\\ &= x_1x_2\\ &= h.\end{aligned}$$

So we find the representative x_1x_2 which is the same as the original representative of h. Therefore, we can obtain the correct plaintext h.

If the second ordering way is

$$M_1 = {A_2}^{-3}B_2{A_2}^3 = \begin{pmatrix} -11 & -72 \\ 2 & 13 \end{pmatrix}$$

and

$$M_2 = {A_2}^{-2}B_2{A_2}^2 = \begin{pmatrix} -7 & -32 \\ 2 & 9 \end{pmatrix},$$

then from the $X(n,S)$-representation

$$\begin{aligned}&{A_2}^{-2}{B_2}^{-15}{A_2}^2{A_2}^{-3}{B_2}^5{A_2}^3{A_2}^{-2}{B_2}^2{A_2}^2\\ &= {M_2}^{-15}{M_1}^5{M_2}^2,\end{aligned}$$

we have the representative ${x_2}^{-15}{x_1}^5{x_2}^2$ and by the relations ${x_1}^2 = 1$, ${x_2}^4 = 1$ and $(x_1x_2)^2 = 1$,

$$
\begin{aligned}
&{x_2}^{-15}{x_1}^{5}{x_2}^{2}\\
&= {x_2}^{-3}x_1{x_2}^{2}\\
&= x_2x_1{x_2}^{2}\\
&= x_2(x_1x_2)x_2\\
&= x_2({x_2}^{-1}{x_1}^{-1})x_2\\
&= (x_2{x_2}^{-1}){x_1}^{-1}x_2\\
&= {x_1}^{-1}x_2\\
&= x_1x_2\\
&= h.
\end{aligned}
$$

So we can also find the same plaintext h in this case, but we might have a different representative unlike this second case. However, one of the two ordering ways works to recover the plaintext h.

8.4 Recovering The X_1-Representation of The Public Key from The X_1-Representation of The Ciphertext

In this section, we show that we can also recover the plaintext without knowing the private key n and S. The attack method uses only the X_1-representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ and the ciphertext matrix M.

Since the ciphertext M is generated by the public key matrices $Y_1, Y_2, \cdots, Y_t$, we can often see much of the X_1-representations of the public key matrices appearing in the X_1- representation of the ciphertext M because significant cancelation has not occured . However, for some Y_i, some letters of Y_i may be canceled and that's why the partial X_1-representation of Y_i appears in the X_1- representation of the ciphertext M. Thus the key idea to attack is to recover the whole X_1-representation of Y_i from the partial X_1-representation of Y_i in the X_1- representation of the ciphertext M. Next, we can describe the X_1-representation of M in terms of the public key matrices $Y_1, Y_2, \cdots, Y_t$ and then by the public one-to-one correspondence between $X = \{x_1, \cdots, x_t\}$ and $\{Y_1, Y_2, \cdots, Y_t\}$, we find a word in $X^{\pm}$. Lastly, we compute a concrete representative or its normal form corresponding to the plaintext and then we obtain the plaintext.

Example

Given a finite presentation $\langle X|\Re\rangle = \langle x_1, x_2|{x_1}^2 = 1, {x_2}^4 = 1, (x_1x_2)^4 = 1, (x_1x_2)^2 = 1\rangle$ of the dihedral group D_4 where $X = \{x_1, x_2\}$ and $\Re = \{w_1 = {x_1}^2, w_2 = {x_2}^4, w_3 = (x_1x_2)^4, w_4 = (x_1x_2)^2\}$ and a security parameter $k = 2$, Bob chooses $n = 2$ and $S = \{s_1, s_2\}$ by $s_1 = 1$ and $s_2 = 2$ and generates random words by $r_1 = w_1w_1$ and $r_2 = w_1w_2$, write $R = \{r_1, r_2\}$. Bob constructs the group $\Gamma_2 = \langle A_2, B_2\rangle$ by his private key $n = 2$ where $A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ and $B_4 = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$ and the group $G(n, S) = \langle M_1, M_2\rangle$ where $M_1 = {A_2}^{-1}B_2{A_2}^1 = \begin{pmatrix} -3 & -8 \\ 2 & 5 \end{pmatrix}$ and $M_2 = {A_2}^{-2}B_2{A_2}^2 = \begin{pmatrix} -7 & -32 \\ 2 & 9 \end{pmatrix}$. Bob generates the public key $\{Y_1, Y_2\}$ by

$$\begin{aligned} Y_1 &= \phi(x_1r_1) \\ &= \phi(x_1w_1w_1) \\ &= \phi({x_1}^5) \\ &= {M_1}^5 \\ &= \begin{pmatrix} -19 & -40 \\ 10 & 21 \end{pmatrix}, \end{aligned}$$

$$
\begin{aligned}
Y_2 &= \phi(x_2 r_2) \\
&= \phi(x_2 w_1 w_2) \\
&= \phi(x_2 x_1 x_1 x_2 x_2 x_2 x_2) \\
&= \phi(x_2 {x_1}^2 {x_2}^4) \\
&= M_2 {M_1}^2 {M_2}^4 \\
&= \begin{pmatrix} 1041 & 4304 \\ -290 & -1199 \end{pmatrix}.
\end{aligned}
$$

Given a representative $x_1 x_2$ of a message h, to encrypt it, Alice chooses the random word $r = w_2 w_1$. Then

$$
\begin{aligned}
& rh \\
&= w_2 w_1 x_1 x_2 \\
&= x_2 x_2 x_2 x_2 x_1 x_1 x_1 x_2 \\
&= {x_2}^4 {x_1}^3 x_2
\end{aligned}
$$

and the ciphertext M is

$$
\begin{aligned}
M &= M_r M_h \\
&= {Y_2}^4 {Y_1}^3 Y_2 \\
&= \begin{pmatrix} 1041 & 4304 \\ -290 & -1199 \end{pmatrix}^4 \begin{pmatrix} -19 & -40 \\ 10 & 21 \end{pmatrix}^3 \begin{pmatrix} 1041 & 4304 \\ -290 & -1199 \end{pmatrix} \\
&= \begin{pmatrix} -120550653510779 & -498415519208360 \\ 33582590741180 & 138846898902381 \end{pmatrix}.
\end{aligned}
$$

where $M_r = {Y_2}^4 {Y_1}^2$ and $M_h = Y_1 Y_2$.

Attack 8

The X_1-representation algorithm computes the X_1-representation of Y_1

$$ {A_1}^{-2} {B_1}^{10} {A_1}^2, $$

the X_1-representation of Y_2

$$ {A_1}^{-4} {B_1}^2 {A_1}^2 {B_1}^4 {A_1}^{-2} {B_1}^8 {A_1}^4. $$

and the X_1-representation of M

$$A_1{}^{-4}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{10}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{10}A_1{}^{2}B_1{}^{4}A_1{}^{-2}$$

$$B_1{}^{10}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{2}B_1{}^{30}A_1{}^{-2}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}.$$

When we observe the X_1-representation of M, we can see the partial X_1-representations of Y_1 and Y_2 in the X_1-representation of M and hence, we split the X_1-representation of the ciphertext M into the partial X_1-representations of Y_1 and Y_2 as follows :

$$\begin{aligned}M =&A_1{}^{-4}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{10}\\ &A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{10}\\ &A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{10}\\ &A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}\\ &A_1{}^{2}B_1{}^{30}\\ &A_1{}^{-2}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}.\end{aligned}$$

The representation of each line above presents the partial X_1-representation of Y_1 or Y_2. To recover the original X_1-representation of Y_1 and Y_2, we insert some letters likely to be parts of the X_1-representations of Y_1 and Y_2. So we can recover the original X_1-representations of Y_1 and Y_2. Therefore, we can obtain the following representation of the ciphertext M from the straightforward computation

$$\begin{aligned}M =&[A_1{}^{-4}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}]\\ &[A_1{}^{-4}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}]\\ &[A_1{}^{-4}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}]\\ &[A_1{}^{-4}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}]\\ &[A_1{}^{-4}A_1{}^{2}B_1{}^{30}A_1{}^{2}]\\ &[A_1{}^{-2}A_1{}^{-2}B_1{}^{2}A_1{}^{2}B_1{}^{4}A_1{}^{-2}B_1{}^{8}A_1{}^{4}].\end{aligned}$$

Therefore, we have $M = {Y_2}^4{Y_1}^3Y_2$ and by the public correspondence $x_1 \mapsto Y_1$ and $x_2 \mapsto Y_2$, we compute a representative ${x_2}^4{x_1}^3x_2$. By the relations ${x_1}^2 = 1$ and ${x_2}^4 = 1$,

$${x_2}^4{x_1}^3x_2 = x_1x_2 = h.$$

Therefore, we recover the plaintext h from the given ciphertext M.

8.5 Comparison

In order to recover the plaintext, we compute the private key n and S to construct the secret basis $X(n, S)$ from Section 8.1 to Section 8.3. In Section 8.1.1, we compute n by the X_1-representation of Y_i or M, whereas in Section 8.1.3, we compute n without using the X_1-representations of Y_i or M. After computing n, we compute elements of S in Section 8.2 and we compute implicitly another secret factor, one-to-one correspondence between X and $X(n, S)$, namely ordering the elements of S in Section 8.3. So these two different approaching ways depend on whether or not we use the X_1-representation of Y_i or M, but in common both ways need ordering the elements of S to recover the plaintext.

Another approaching way is regardless of the secret key n and S, only by using the X_1-representations of the public key matrices $Y_1, Y_2, \cdots, Y_t$ and the ciphertext M, we can obtain the plaintext in Section 8.4. So there is a clear difference between the approaches above and this approaching way to recover the plaintext. However, in both Section 8.1 and Section 8.4 we utilize the X_1-representations of the public key matrices and the ciphertext matrix. Since ordering the elements of S is implicitly secret, we have to do exhaustive search for the one-to-one correspondence between X and $X(n, S)$ and it means that there is still ambiguity to find the correct plaintext relying on ordering the elements of S. Thus such a situation produces many likely representations of the plaintext according as ordering the elements of S. If the size of the generating set X is small, then there might not be much difference between the two ways. However, if the number of elements of the generating set X is large, then it would take a time to find the plaintext because we do exhaustive search for ordering the elements of S. In contrast, the attack method in Section 8.4 needs only public one-to-one correspondence between $X = \{x_1, x_2, \cdots, x_t\}$ and $\{Y_1, Y_2, \cdots, Y_t\}$ by $x_i \mapsto Y_i$ and obviously there is no ambiguity to find the plaintext because we have only one representation for the plaintext. So in in some case, the attack method in Section 8.4 may be efficient way to recover the plaintext.

Appendix

The followings are the source codes of experiment results of Section 9.1.3 of Chapter 9.

Experiment 1

(1) $k = 3, \mathrm{length} = 3;$

```
> su:=proc()
> local u1, u2, w;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> w:=gcd(u1, u2);
> if not (w = 1) then
> print(u1, u2, w);
> fi;
> od;
> od;
> end:
> su();
```

(2) $k = 3, \mathrm{length} = 5;$

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if not (w4 = 1) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(3) $k = 3, \text{length} = 7;$

```
> su:=proc()
> local u1, u2, u3, u4, u5, u6, u7, w1, w2, w3, w4, w5, w6;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> for u6 from 1 to 7 do
> for u7 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> w5 := gcd(w4, u6);
> w6 := gcd(w5, u7);
> if not (w6 = 1) then
> print(u1, u2, u3, u4, u5, u6, u7, w6);
> fi;
> od;
> od;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(4) $k = 3, \text{length} = 9$;

```
> su:=proc()
> local u1, u2, u3, u4, u5, u6, u7, u8, u9, w1, w2, w3, w4, w5, w6, w7, w8;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> for u6 from 1 to 7 do
> for u7 from 1 to 7 do
> for u8 from 1 to 7 do
> for u9 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> w5 := gcd(w4, u6);
> w6 := gcd(w5, u7);
> w7 := gcd(w6, u8);
> w8 := gcd(w7, u9);
> if not (w8 = 1) then
> print(u1, u2, u3, u4, u5, u6, u7, u8, u9, w8);
> fi;
> od;
> od;
> od;
> od;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

Experiment 2

(1)$k = 2, \text{length} = 5$

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 3 do
> for u2 from 1 to 3 do
> for u3 from 1 to 3 do
> for u4 from 1 to 3 do
> for u5 from 1 to 3 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if not (w4 = 1) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(2)$k = 3$, length $= 5$

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if not (w4 = 1) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(3)$k = 4$, length $= 5$

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if not (w4 = 1) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

Experiment 3

```
> f1 := rand(1..2^5) :
> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> localb1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5);
> print(b4);
> end  proc :

> f2(11, 6, 9, 2, 2);
> f2(16, 28, 26, 17, 27);
> f2(6, 29, 6, 12, 22);
> f2(7, 31, 11, 14, 30);
> f2(22, 15, 7, 2, 23);
> f2(24, 10, 5, 4, 11);
> f2(18, 29, 5, 6, 24);
> f2(28, 19, 6, 1, 31);
> f2(9, 6, 14, 22, 16);
> f2(14, 21, 4, 4, 20);
> f2(14, 18, 22, 10, 4);

> f2(3, 9, 20, 24, 20);

> f2(8, 23, 22, 21, 27);

> f2(26, 31, 20, 24, 3);

> f2(2, 16, 23, 2, 29);

> f2(11, 6, 9, 2, 2);

> f2(16, 15, 20, 17, 4);

> f2(4, 30, 4, 30, 10);

> f2(13, 20, 13, 28, 10);

> f2(17, 28, 8, 29, 22);

> f2(20, 27, 19, 3, 31);

> f2(4, 10, 13, 29, 19);

> f2(31, 24, 24, 25, 12);

> f2(1, 32, 14, 27, 9);

> f2(3, 13, 31, 4, 5);

> f2(12, 23, 12, 21, 19);

> f2(20, 21, 15, 9, 29);

> f2(28, 1, 8, 20, 24);

> f2(8, 3, 11, 31, 26);

> f1(2, 16, 23, 2, 29);

> f2(1, 1, 27, 6, 18);

> f2(10, 23, 29, 1, 32);

> f2(15, 8, 28, 6, 29);

> f2(26, 6, 9, 13, 13);

> f2(20, 29, 21, 8, 27);

> f2(15, 14, 17, 18, 22);

> f2(14, 15, 19, 21, 20);

> f2(4, 16, 15, 28, 18);

> f2(26, 16, 15, 28, 5);
```

$> f2(12, 23, 12, 21, 19);$

$> f2(20, 21, 15, 9, 29);$

$> f2(28, 1, 8, 20, 24);$

$> f2(8, 3, 11, 31, 26);$

$> f1(2, 16, 23, 2, 29);$

$> f2(28, 7, 15, 8, 14);$

$> f2(18, 5, 16, 22, 5);$

$> f2(32, 19, 23, 5, 31);$

$> f2(21, 26, 17, 2, 15);$

$> f2(27, 8, 10, 6, 30);$

$> f2(23, 14, 16, 23, 28);$

$> f2(24, 14, 12, 22, 28);$

$> f2(27, 14, 4, 25, 31);$

$> f2(15, 24, 18, 24, 10);$

$> f2(17, 27, 26, 23, 30);$

$> f2(26, 28, 14, 17, 19);$

$> f2(21, 30, 6, 14, 1);$

$> f2(12, 30, 5, 20, 8);$

$> f2(14, 16, 2, 24, 21);$

$> f2(16, 1, 19, 8, 8);$

$> f2(32, 2, 26, 12, 1);$

$> f2(21, 8, 6, 9, 27);$

$> f2(17, 15, 2, 12, 13);$

$> f2(14, 8, 32, 23, 21);$

$> f2(13, 4, 11, 29, 3);$

$> f2(9, 17, 17, 24, 2);$

$> f2(9, 29, 32, 25, 6);$

$> f2(16, 7, 29, 29, 32);$

$> f2(6, 25, 28, 30, 13);$

$> f2(2, 32, 5, 19, 23);$

$> f2(25, 17, 23, 15, 25);$

$> f2(30, 30, 24, 21, 21);$

$> f2(17, 9, 25, 24, 21);$

$> f2(11, 21, 18, 23, 18);$

$> f2(12, 21, 7, 32, 26);$

$> f2(26, 22, 23, 15, 13);$

$> f2(19, 3, 32, 5, 23);$

$> f2(20, 8, 2, 22, 3);$

$> f2(28, 11, 22, 18, 13);$

$> f2(26, 26, 25, 25, 23);$

$> f2(11, 29, 4, 21, 19);$

$> f2(12, 22, 32, 23, 9);$

$> f2(30, 8, 16, 14, 2);$

$> f2(1, 10, 7, 2, 11);$

$> f2(18, 2, 24, 5, 15);$

$> f2(13, 15, 13, 18, 23);$

$> f2(18, 7, 13, 10, 4);$

$> f2(27, 21, 18, 32, 24);$

$> f2(17, 10, 18, 20, 14);$

$> f2(2, 16, 23, 2, 29);$

$> f2(22, 1, 23, 26, 30);$

$> f2(27, 3, 19, 25, 13);$

$> f2(18, 3, 30, 1, 18);$

$> f2(9, 1, 6, 10, 3);$

$> f2(23, 16, 9, 25, 17);$

$> f2(25, 28, 24, 9, 32);$

$> f2(23, 23, 7, 32, 8);$

$> f2(6, 13, 20, 18, 23);$

$> f2(8, 27, 25, 25, 12);$

$> f2(29, 5, 30, 1, 30);$

$> f1 := \mathrm{rand}(1..2^{10}) :$

$> \mathrm{seq}(f1(), i = 1..5);$

$> f2 := \mathrm{proc}(a1 :: \mathrm{integer}, a2 :: \mathrm{integer}, a3 :: \mathrm{integer}, a4 :: \mathrm{integer}, a5 :: \mathrm{integer})$

$> \mathrm{local}\ \ b1, b2, b3, b4;$

$> b1 := \mathrm{gcd}(a1, a2);$

$> b2 := \mathrm{gcd}(b1, a3);$

$> b3 := \mathrm{gcd}(b2, a4);$

$> b4 := \mathrm{gcd}(b3, a5);$

$> \mathrm{print}(b4);$

$> \mathrm{end}\ \ \mathrm{proc} :$

$> f2(702, 99, 783, 698, 559);$

$> f2(726, 588, 279, 64, 436);$

$> f2(892, 234, 80, 96, 894);$

$> f2(210, 521, 990, 857, 263);$

$> f2(415, 95, 400, 486, 251);$

$> f2(759, 156, 969, 857, 490);$

$> f2(716, 437, 15, 855, 619);$

$> f2(39, 688, 1, 966, 571);$

$> f2(354, 340, 591, 216, 122);$

$> f2(784, 465, 902, 359, 370);$

$> f2(963, 354, 937, 742, 157);$

$> f2(222, 469, 358, 250, 61);$

$> f2(528, 209, 818, 137, 276);$

$> f2(666, 235, 626, 935, 70);$

$> f2(465, 547, 349, 457, 759);$

```
> f2(395, 476, 608, 155, 845);

> f2(846, 305, 464, 1, 953);

> f2(999, 710, 1004, 500, 943);

> f2(575, 489, 197, 880, 30);

> f2(812, 99, 510, 720, 366);

> f2(22, 534, 189, 819, 234);

> f2(752, 6, 656, 309, 161);

> f2(459, 898, 316, 563, 519);

> f2(16, 559, 524, 428, 170);

> f2(165, 324, 645, 421, 613);

> f2(975, 1005, 388, 667, 743);

> f2(180, 81, 641, 73, 339);

> f2(28, 1, 8, 20, 24);

> f2(8, 3, 11, 31, 26);

> f2(952, 365, 807, 273, 419);

> f2(206, 302, 160, 759, 133);

> f2(649, 329, 445, 306, 801);

> f2(264, 752, 841, 947, 929);

> f2(363, 47, 623, 909, 987);

> f2(3, 843, 60, 741, 363);

> f2(490, 9, 700, 673, 574);

> f2(565, 717, 420, 266, 519);

> f2(149, 308, 1009, 173, 203);

> f2(270, 880, 184, 763, 50);

> f2(894, 817, 606, 129, 433);

> f2(783, 649, 66, 335, 57);

> f2(878, 231, 919, 845, 804);

> f2(124, 178, 53, 385, 40);

> f2(602, 360, 836, 597, 983);
```

$> f2(150, 676, 907, 977, 842);$

$> f2(362, 329, 151, 851, 866);$

$> f2(746, 706, 293, 356, 317);$

$> f2(133, 267, 744, 685, 118);$

$> f2(521, 653, 265, 874, 973);$

$> f2(873, 322, 417, 721, 311);$

$> f2(835, 975, 843, 149, 256);$

$> f2(349, 86, 915, 401, 399);$

$> f2(358, 844, 928, 290, 255);$

$> f2(963, 362, 584, 600, 546);$

$> f2(939, 224, 257, 495, 741);$

$> f2(820, 909, 529, 216, 803);$

$> f2(411, 743, 746, 993, 679);$

$> f2(648, 507, 962, 509, 248);$

$> f2(464, 436, 167, 507, 898);$

$> f2(140, 917, 585, 463, 874);$

$> f2(469, 715, 313, 798, 569);$

$> f2(66, 754, 662, 632, 406);$

$> f2(559, 184, 560, 221, 158);$

$> f2(127, 59, 244, 448, 704);$

$> f2(139, 155, 304, 1005, 994);$

$> f2(299, 7, 980, 500, 283);$

$> f2(587, 707, 496, 246, 845);$

$> f2(174, 492, 333, 391, 871);$

$> f2(660, 255, 969, 4, 696);$

$> f2(926, 150, 466, 143, 955);$

$> f2(287, 215, 645, 455, 161);$

$> f2(985, 673, 90, 540, 689);$

$> f2(798, 150, 165, 675, 718);$

$> f2(76, 205, 390, 957, 560);$

$> f2(889, 576, 310, 562, 60);$

$> f2(929, 418, 167, 382, 943);$

$> f2(516, 104, 260, 252, 44);$

$> f2(962, 912, 111, 742, 638);$

$> f2(672, 914, 183, 698, 351);$

$> f2(995, 235, 397, 208, 718);$

$> f2(468, 935, 1022, 717, 637);$

$> f2(722, 1023, 546, 1008, 705);$

$> f2(841, 416, 745, 475, 133);$

$> f2(265, 670, 707, 328, 771);$

$> f2(793, 42, 363, 168, 234);$

$> f2(580, 624, 114, 326, 805);$

$> f2(567, 597, 199, 638, 734);$

$> f2(585, 957, 498, 405, 59);$

$> f2(655, 294, 3, 22, 379);$

$> f2(948, 346, 640, 102, 887);$

$> f2(910, 192, 11, 878, 825);$

$> f2(1000, 722, 854, 108, 36);$

$> f2(346, 629, 950, 965, 575);$

$> f2(1022, 61, 219, 934, 1024);$

$> f2(773, 462, 585, 292, 476);$

$> f2(938, 142, 822, 437, 1);$

$> f2(371, 217, 476, 223, 311);$

$> f2(256, 27, 631, 298, 46);$

$> f2(238, 533, 629, 176, 561);$

$> f2(632, 1015, 508, 14, 293);$

$> f1 := \mathrm{rand}(1..2^{15}) :$

```
> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> local  b1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5);
> print(b4);
> end  proc :
> f2(4698, 21911, 5666, 29900, 19201);
> f2(11915, 12038, 7977, 25698, 1314);
> f2(18448, 29628, 12442, 4913, 28827);
> f2(6822, 19133, 3782, 13804, 24726);
> f2(13287, 10623, 9163, 15278, 20734);
> f2(12502, 47, 32519, 28770, 4343);
> f2(5688, 18154, 26341, 15172, 23179);
> f2(10962, 19613, 2629, 5574, 8376);
> f2(10044, 28883, 23494, 25217, 5151);
> f2(23886, 4181, 14404, 26628, 2644);
> f2(19278, 9426, 31766, 22602, 16516);
> f2(32611, 7241, 2068, 13208, 6708);
> f2(18600, 10951, 12406, 16917, 10379);
> f2(28986, 22175, 7924, 504, 14563);
> f2(25698, 25008, 23191, 9506, 29213);
> f2(6241, 20070, 30958, 14547, 51);
> f2(19757, 23632, 5455, 28308, 15921);
> f2(3684, 27982, 2206, 23012, 22366);
> f2(14666, 15181, 16756, 15149, 10652);
> f2(1162, 11793, 24988, 22952, 1789);
> f2(1846, 28116, 14751, 31731, 30627);
> f2(30047, 2692, 27562, 7053, 31549);
> f2(10067, 7743, 5528, 31288, 3961);
> f2(27436, 3201, 13376, 15566, 29947);
> f2(21513, 10851, 16845, 8767, 484);
> f2(16605, 27980, 3159, 14828, 30581);
> f2(1843, 32660, 8341, 16687, 13065);
> f2(13501, 9884, 9313, 2312, 14516);
> f2(5272, 28200, 16227, 9771, 31103);
> f2(17722, 28577, 1889, 31195, 31398);
> f2(3698, 27486, 29431, 24829, 18977);
> f2(31968, 25743, 30888, 3644, 4102);
> f2(27069, 2980, 618, 13784, 30790);
> f2(30406, 10310, 23748, 19730, 4264);
> f2(5662, 2746, 30118, 25321, 28173);
> f2(4429, 7764, 15517, 22805, 28296);
> f2(6875, 32655, 21134, 24817, 10002);
> f2(15030, 14446, 22447, 18899, 6197);
> f2(15220, 14100, 31754, 1417, 8421);
> f2(24875, 21092, 8432, 10415, 26364);
> f2(9042, 23834, 31760, 21871, 12476);
> f2(4229, 32284, 32615, 3087, 14344);
> f2(21797, 14560, 18035, 29271, 20069);
> f2(1983, 24213, 2874, 14033, 25122);
> f2(9647, 31675, 15176, 31242, 30310);
> f2(11915, 12038, 7977, 25698, 1314);
> f2(18448, 29628, 12442, 4913, 28827);
> f2(6822, 19133, 3782, 13804, 24726);
> f2(13287, 16023, 9163, 15278, 20734);
> f2(12502, 47, 32519, 28770, 4343);

> f2(5688, 18154, 26341, 15172, 23179);

> f2(10962, 19613, 2629, 5574, 8376);

> f2(10044, 28883, 23494, 25217, 5151);

> f2(3145, 7814, 622, 8918, 15568);

> f2(23886, 4181, 14404, 26628, 2644);

> f2(19278, 9426, 31766, 22602, 16516);

> f2(32611, 7241, 2068, 13208, 6708);

> f2(18600, 20951, 12406, 16917, 20379);

> f2(28986, 22175, 7924, 504, 14563);

> f2(25698, 25008, 23191, 9506, 29213);

> f2(6241, 30070, 30958, 14547, 51);

> f2(19757, 23632, 5455, 28308, 15921);

> f2(3684, 27982, 2206, 23012, 22366);

> f2(14666, 15181, 16756, 15149, 10652);

> f2(1162, 11793, 24988, 22952, 1789);

> f2(1846, 28116, 14715, 31731, 30627);

> f2(30027, 2692, 27562, 7053, 31549);

> f2(10067, 7743, 5528, 31288, 3961);

> f2(27436, 3201, 13376, 15566, 29947);

> f2(21513, 10851, 16845, 8767, 484);

> f2(16005, 27980, 3159, 14828, 30581);

> f2(1843, 32660, 8341, 16687, 13065);

> f2(13501, 9884, 9313, 2312, 14516);

> f2(17722, 28577, 1889, 31195, 31398);

> f2(3698, 27586, 29431, 24829, 18977);

> f2(31968, 25743, 30888, 3644, 4102);

> f2(27069, 2980, 618, 13784, 30790);

> f2(10406, 10310, 23748, 19730, 4264);
> f2(5662, 2746, 30118, 25321, 28173);
> f2(4429, 7764, 15517, 22805, 28296);
> f2(6875, 32655, 21134, 24817, 10002);
> f2(15030, 14446, 22447, 18899, 6197);
> f2(15220, 14100, 31754, 1417, 8421);
> f2(9042, 23834, 31760, 21871, 12476);
> f2(4229, 32284, 32615, 3087, 14344);
> f2(1902, 26770, 21061, 23600, 470);
> f2(21797, 14560, 18035, 29271, 20069);
> f2(1983, 24213, 2874, 14033, 25122);
> f2(9647, 31675, 15176, 31242, 30310);
> f2(10174, 22135, 11118, 25968, 10231);
> f1 := rand(1..2^15) :

> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> local  b1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5);
> print(b4);
> end  proc :

> f2(26812, 20216, 3950, 14252, 11958);
> f2(17212, 27945, 2411, 6222, 27391);
> f2(10073, 10907, 18638, 23556, 24537);
> f2(19583, 28943, 18040, 23186, 9016);
> f2(21290, 29905, 12251, 1274, 9399);
> f2(19454, 17306, 22780, 2830, 12689);
> f2(30451, 29653, 20382, 24134, 2446);
> f2(27169, 2444, 7998, 22309, 1716);
> f2(24008, 9486, 15344, 25058, 27032);
> f2(20373, 18352, 4929, 8051, 29320);

> f1 := rand(1..2^20) :
> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> local  b1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5); > print(b4);
> end  proc :

> f2(168538, 21911, 38434, 947404, 772865);
> f2(994955, 962310, 892713, 779362, 558370);
> f2(1034256, 29628, 635034, 37681, 1011867);
> f2(72358, 379581, 921286, 210412, 385174);
> f2(340967, 239999, 959435, 801710, 708862);
> f2(209110, 589871, 589575, 389218, 856311);
> f2(595512, 1033962, 255717, 80708, 613003);
> f2(76498, 904349, 2629, 136646, 8376);
> f2(272188, 880851, 383942, 418433, 726047);
> f2(396361, 990854, 197230, 172758, 605392);
> f2(122190, 823381, 407620, 583684, 363092);
> f2(248654, 795858, 228374, 383050, 508036);
> f2(196451, 793673, 198676, 897944, 662068);
```

$> f2(772264, 53719, 274550, 246293, 708507);$

$> f2(323898, 1037983, 597748, 164344, 702691);$

$> f2(287842, 942512, 744087, 501026, 61981);$

$> f2(287842, 942512, 744087, 501026, 61981);$

$> f2(923745, 587126, 194798, 768211, 426035);$

$> f2(707885, 842832, 595279, 585364, 867889);$

$> f2(495204, 126286, 2206, 481764, 120670);$

$> f2(833866, 605005, 409972, 768813, 731548);$

$> f2(33930, 470545, 876956, 219560, 231165);$

$> f2(722742, 224724, 244091, 457715, 1013667);$

$> f2(1013087, 592516, 551850, 498573, 686909);$

$> f2(960339, 400959, 628120, 621112, 200569);$

$> f2(420652, 855169, 209984, 15566, 128251);$

$> f2(742409, 43619, 803277, 598591, 557540);$

$> f2(933509, 552268, 953431, 473580, 63349);$

$> f2(984883, 556948, 41109, 311599, 1028873);$

$> f2(242877, 370332, 828513, 1018120, 604340);$

$> f2(201880, 814632, 1032035, 632363, 227711);$

$> f2(17722, 192417, 1889, 227803, 260774);$

$> f2(462450, 486238, 586487, 90365, 412193);$

$> f2(589024, 746639, 129192, 396860, 430086);$

$> f2(911805, 166820, 1016426, 210392, 653382);$

$> f2(95942, 960582, 449732, 118034, 69800);$

$> f2(398878, 625338, 980390, 680681, 126477);$

$> f2(233805, 499284, 1031325, 710933, 585352);$

$> f2(924379, 196495, 316046, 909553, 10002);$

$> f2(539318, 702574, 415663, 477651, 38965);$

$> f2(572276, 931604, 130058, 296329, 860389);$

$> f2(942379, 283236, 336112, 174255, 976636);$

$> f2(893778, 417050, 654352, 1004911, 340156);$

$> f2(888965, 130588, 524135, 166927, 538632);$

$> f2(296814, 977042, 1004101, 23600, 360918);$

$> f2(840997, 1030368, 706163, 881239, 544357);$

$> f2(67519, 614037, 625466, 603857, 778786);$

$> f2(697775, 129979, 80712, 522762, 554598);$

$> f2(665534, 611959, 961390, 714096, 108535);$

$> f2(125116, 20216, 397166, 964524, 44726);$

$> f2(475964, 716073, 493931, 104526, 125695);$

$> f2(894809, 797339, 542926, 875524, 679897);$

$> f2(806015, 946447, 50808, 416402, 566072);$

$> f2(512810, 881873, 634843, 34042, 894135);$

$> f2(740350, 574362, 22780, 625422, 242065);$

$> f2(1046259, 95189, 282526, 1039942, 723342);$

$> f2(616993, 657804, 696126, 120613, 624308);$

$> f2(679368, 304398, 179184, 614882, 879000);$

$> f2(247464, 689261, 461243, 516672, 454146);$

$> f2(419482, 431916, 987169, 260533, 390184);$

$> f2(169926, 804585, 409019, 264561, 807663);$

$> f2(875138, 585804, 846861, 389358, 112648);$

$> f2(989792, 679127, 281173, 325517, 657764);$

$> f2(48555, 7261, 934115, 429609, 715313);$

$> f2(863441, 689240, 165730, 745097, 842749);$

$> f2(398112, 471673, 698502, 227984, 416839);$

$> f2(633277, 221181, 245536, 250177, 102726);$

$> f2(989946, 717327, 949160, 832006, 103001);$

$> f2(401628, 445054, 877389, 888610, 1005888);$

$> f2(559877, 124531, 577719, 1004636, 601381);$

$> f2(323890, 761709, 35120, 853369, 1024625);$

```
> f2(697367, 42095, 400601, 232542, 839070);
> f2(675864, 546293, 528917, 369137, 326505);
> f2(994041, 890040, 97685, 347211, 1026677);
> f2(497298, 827447, 951026, 758956, 546357);
> f2(407431, 750080, 487162, 385551, 201178);
> f2(751222, 278295, 1039279, 487949, 634355);
> f2(346083, 156032, 908357, 790999, 614228);
> f2(198024, 603106, 727734, 676675, 648252);
> f2(504939, 1021974, 35666, 149005, 708218);
> f2(475034, 555193, 795481, 718551, 608907);
> f2(999805, 338084, 247061, 723059, 420972);
> f2(109526, 91456, 46871, 16393, 840716);
> f2(381175, 861746, 538956, 166748, 778202);
> f2(16700, 221549, 684260, 741280, 573694);
> f2(142756, 546787, 930427, 960095, 499497);
> f2(21797, 14560, 18035, 29271, 20069);
> f2(222120, 655632, 782286, 1005410, 314561);
> f2(338090, 1025927, 501826, 856171, 778514);
> f2(1039202, 127480, 967973, 98543, 465581);
> f2(911503, 1018637, 381714, 62231, 72946);
> f2(453351, 212973, 342794, 462180, 514779);
> f2(69973, 484146, 394624, 289688, 440209);
> f2(194634, 230098, 607220, 392174, 1020086);
> f2(143553, 375063, 888730, 193566, 430811);
> f2(7939, 626643, 720665, 321679, 336530);
> f2(989635, 145982, 833121, 19090, 341705);
> f2(641921, 490278, 451466, 587235, 759351);
> f2(978608, 753385, 272793, 436529, 368185);
> f2(619708, 911032, 117289, 411232, 527991);
> f1 := rand(1..2^25) :
> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> local  b1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5);
> print(b4);
> end  proc :
> f2(3314266, 28333463, 2135586, 31356108, 12307201);
> f2(33500811, 17739526, 10329897, 23848034, 558370);
> f2(33540112, 32535484, 26849434, 11572017, 14643355);
> f2(3218086, 30788285, 23989958, 21181932, 25550998);
> f2(20263911, 21211519, 32416715, 30161838, 33214718);
> f2(4403414, 12124207, 28901127, 2486370, 13439223);
> f2(18421304, 7325418, 26470117, 23149380, 12147339);
> f2(24193746, 20827293, 2629, 3282374, 13639864);
> f2(10757948, 4026579, 28695494, 5661313, 31134751);
> f2(8784969, 3088006, 29557358, 11707094, 28916944);
> f2(20045134, 32280661, 31864900, 15263748, 6654548);
> f2(30657358, 28058834, 8616982, 383050, 4702340);
> f2(6487907, 11279433, 7538708, 4043672, 5904948);
> f2(20695208, 25219543, 24391798, 25412117, 11194267);
> f2(22343994, 17815199, 28909300, 11698680, 702691);
> f2(7627874, 25059760, 29055639, 3646754, 29422109);
> f2(28186721, 29947254, 14874862, 33274067, 28737587);
> f2(14339373, 19717200, 31003983, 5828244, 29179441);
> f2(8883812, 25292110, 5245086, 21453284, 13752158);

> f2(10271050, 22625101, 409972, 29080365, 16460188);

> f2(18908298, 8859153, 2974108, 32725416, 3376893);

> f2(10159926, 1273300, 25409915, 28769267, 17790883);

> f2(6255967, 6883972, 14183338, 31955853, 4881213);

> f2(18786131, 27663935, 2725272, 4815416, 20123513);

> f2(22440748, 23923841, 5452864, 24132814, 29488379);

> f2(1790985, 18917987, 13386189, 24715839, 3703268);

> f2(19807877, 16280908, 7244887, 2570732, 18937717);

> f2(2033459, 26771348, 41109, 23380271, 4174601);

> f2(19117245, 18196124, 13411425, 2066696, 24721588);

> f2(4396184, 8154664, 33537891, 24749611, 12810623);

> f2(12600634, 8581025, 9439073, 24345051, 2357926);

> f2(4656754, 7826270, 24703735, 18964733, 32918049);

> f2(6880480, 33252495, 129192, 396860, 3575814);

> f2(7203261, 23235492, 8356458, 8599000, 4847686);

> f2(20018886, 22980678, 24566980, 2215186, 26284200);

> f2(22418974, 10062522, 5174694, 23749353, 23195149);

> f2(32739661, 30907988, 17808541, 17488149, 16313992);

> f2(15604443, 14876559, 17093262, 3006705, 20981522);

> f2(21510838, 24819822, 18241455, 24594899, 18913333);

> f2(32029556, 22951700, 20053002, 5539209, 11346149);

> f2(14573867, 28594788, 27599088, 32680111, 15656700);

> f2(19768146, 29777178, 10091536, 8344943, 22360252);

> f2(27103365, 26344988, 1572711, 2264079, 1587208);

> f2(24414062, 977042, 23024197, 17849392, 16089558);

> f2(10278181, 31439072, 27969139, 21852759, 33050213);

> f2(28379071, 16342677, 22645562, 11089617, 25944610);

> f2(32155055, 7470011, 17906504, 14154250, 18380390);
```

$> f2(10102718, 25777783, 21932910, 23782768, 2205687);$

$> f2(18999484, 27283192, 18222958, 22984620, 8433334);$

$> f2(6767420, 17493289, 493931, 9541710, 125695);$

$> f2(13477721, 22817435, 1591502, 28138500, 18505689);$

$> f2(18631807, 29257999, 2147960, 18242194, 19440440);$

$> f2(19387178, 15561937, 32092123, 7374074, 25011383);$

$> f2(1788926, 27837338, 1071356, 11111182, 242065);$

$> f2(27260659, 10580949, 31739806, 7331398, 9111950);$

$> f2(14248481, 6949260, 31104830, 17946405, 11110068);$

$> f2(30039496, 32810254, 5422064, 23683554, 15559064);$

$> f2(11620245, 3884976, 10490689, 15769459, 18608776);$

$> f2(22267560, 10126445, 24578491, 28828224, 21425666);$

$> f2(20342426, 3577644, 25104417, 12843445, 30798888);$

$> f2(30578630, 24921833, 2506171, 20187505, 807663);$

$> f2(20798082, 7925836, 7138317, 29749486, 32618504);$

$> f2(31398496, 20602071, 1329749, 13957005, 14289252);$

$> f2(26262955, 17833053, 15614179, 9866793, 26929713);$

$> f2(5057745, 5932120, 27428706, 25910921, 18668541);$

$> f2(29758240, 8860281, 19572870, 4422288, 15096903);$

$> f2(18459069, 22241277, 7585568, 6541633, 5345606);$

$> f2(9378554, 14348815, 11434920, 19706374, 24220249);$

$> f2(27664604, 22465150, 33383245, 10325794, 4151616);$

$> f2(23628549, 12707443, 18403511, 25121884, 29961509);$

$> f2(16052530, 10198893, 21006640, 13436281, 9413233);$

$> f2(14328855, 30450799, 400601, 8621150, 31247774);$

$> f2(4139769, 16618680, 23166357, 11881547, 7318133);$

$> f2(22517394, 1876023, 32408306, 11244716, 27809333);$

$> f2(2504583, 750080, 21458682, 4579855, 15929818);$

$> f2(407431, 750080, 487162, 385551, 201178);$

```
> f2(10188406, 25444119, 10476463, 2585101, 7974387);
> f2(27609059, 22176128, 14539845, 2888151, 4808532);
> f2(22218120, 13186018, 2824886, 30036803, 4842556);
> f2(19379307, 9410582, 10521426, 14829069, 5951098);
> f2(9912218, 23623865, 18621273, 1767127, 3754635);
> f2(4145533, 6629540, 23315733, 5965939, 7761004);
> f2(16886742, 14771520, 13678359, 6307849, 26006540);
> f2(21352695, 2958898, 23607628, 14846812, 16506842);
> f2(142756, 5789667, 7221883, 9348703, 6790953);
> f2(13648188, 25387373, 8024292, 13324192, 13156606);
> f2(10707880, 11141392, 15462350, 5199714, 22334657);
> f2(10707880, 11141392, 15462350, 5199714, 22334657);
> f2(33545058, 2224632, 9356581, 7438575, 11999917);
> f2(6154383, 27233037, 17158930, 16839447, 2170098);
> f2(22473447, 12795885, 24460042, 29822308, 2611931);
> f2(28381525, 29844274, 14026112, 6581144, 26654609);
> f2(4388938, 28541650, 31015924, 5635054, 20943030);
> f2(17969345, 10860823, 23957402, 10679326, 19305179);
> f2(9445123, 10063827, 16449305, 23390351, 10822290);
> f2(25106883, 4340286, 24950369, 19090, 25507529);
> f2(26856321, 21461798, 23520138, 24704483, 3905079);
> f2(27193008, 5996265, 30681497, 27699505, 1416761);
```

```
> f1 := rand(1..2^30) :
> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> local  b1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5);
> print(b4);
> end  proc :

> f2(70423130, 61887895, 35690018, 735999180, 112970497);
> f2(1040173072, 837841852, 463057050, 514888497, 383742107);
> f2(573643430, 97897149, 627969734, 860042732, 428204182);
> f2(792015847, 457419135, 971940811, 969685934, 368759038);
> f2(943927510, 716767279, 599326471, 438693986, 13439223);
> f2(18421304, 74434282, 664004325, 224475972, 213473931);
> f2(493955794, 725470365, 671091269, 909252038, 986718392);
> f2(245638972, 574451923, 565566406, 911630977, 31134751);
> f2(411438153, 607067782, 834863726, 917676758, 431570128);
> f2(825351502, 267161685, 937834564, 15263748, 73763412);
> f2(1070844750, 1068246226, 981695510, 235264074, 172474500);
> f2(610467683, 1017912393, 309528596, 775795608, 173677108);
> f2(658229416, 796971479, 91500662, 58966549, 850055067);
> f2(861204794, 957339295, 834215668, 414351864, 336247011);
> f2(779379810, 327049648, 431708823, 137864482, 1002500637);
> f2(833493089, 1070134646, 249755886, 167491795, 397836339);
> f2(81448237, 623696976, 970528079, 978906772, 29179441);
> f2(42438244, 763489614, 642779294, 759650788, 886167390);
> f2(1016904010, 962149197, 906379636, 29080365, 150677916);
> f2(1059095690, 881274385, 2974108, 200497576, 573802237);
> f2(513476406, 1041460692, 293845371, 297204723, 655325091);
> f2(845116767, 141101700, 785935274, 870816653, 373979965);
> f2(488548179, 61218367, 237606296, 843676216, 255004537);
> f2(223767340, 1064111233, 374551616, 795884750, 667022587);
> f2(1008423945, 287353443, 214712781, 427369023, 37257700);
> f2(858668677, 955805004, 510561367, 271006188, 656471925);
> f2(1042220851, 429424532, 201367701, 694468911, 641708809);
> f2(757314749, 857056924, 1020044385, 35621128, 595146932);
> f2(138613912, 847015464, 503299939, 293185067, 918780287);
> f2(650134842, 579006369, 210765665, 259226075, 1008990886);
> f2(575082098, 645360478, 427356919, 320954621, 167135777);
> f2(141098208, 268133519, 906098856, 872812092, 775327750);
> f2(174975421, 459443108, 981434986, 243480024, 105510982);
> f2(254899910, 22980678, 293002436, 975293714, 126947496);
> f2(89527838, 278497978, 877589926, 426402537, 560066061);
> f2(1072927053, 1071095380, 722451613, 285923605, 586739336);
> f2(250485467, 954400655, 117756558, 103670001, 926951186);
> f2(424164022, 763017326, 957765551, 494356947, 555784245);
> f2(736672628, 291387156, 187825162, 575964553, 850206949);
> f2(886989099, 28594788, 832905456, 938649775, 921626364);
> f2(925737810, 163994906, 614071312, 712988015, 693448892);
> f2(496865413, 93453852, 68681575, 1008897039, 370685960);
> f2(91522926, 504293522, 828330565, 554720304, 720732630);
> f2(513594661, 1071626464, 699057779, 357397079, 871911013);

> f1 := rand(1..2^35) :
> seq(f1(), i = 1..5);
> f2 := proc(a1 :: integer, a2 :: integer, a3 :: integer, a4 :: integer, a5 :: integer)
> local  b1, b2, b3, b4;
> b1 := gcd(a1, a2);
> b2 := gcd(b1, a3);
> b3 := gcd(b2, a4);
> b4 := gcd(b3, a5);
> print(b4);
> end  proc :

> f2(3269307980, 26793946391, 19850198080, 30240777652, 25081555836);
> f2(18632741098, 10965063760, 12909431904, 26874001278, 29572849874);
> f2(26267534857, 32746199006, 641808217, 30149802247, 21673854367);
> f2(14894775391, 9180683664, 29275641318, 24016795899, 24959761143);
> f2(3063288988, 31035908041, 24781263705, 4435643882, 29784965836);
> f2(2865148341, 26473123855, 3699650391, 20093271659, 23541926951);
> f2(20951051952, 21953437697, 23722028998, 11445285435, 12784642402);
> f2(5305374036, 9079888463, 12634014936, 13486065786, 16990189328);
> f2(25125729745, 23062484870, 23397440871, 29702303090, 30225099715);
> f2(24181355874, 27786263465, 13953772262, 3264343197, 12556586206);
> f2(5847633365, 603472230, 30690735354, 28998726717, 18270558736)
> f2(27945147601, 8591323954, 14267311241, 24920459540, 700275354);
> f2(3898368235, 31400791666, 24259688359, 22895394886, 23927768529);
> f2(18755534371, 32312406365, 13101683145, 33050561271, 1433291707);
> f2(19321809291, 9320147420, 27122689632, 5541357723, 16869703501);
> f2(9091948366, 24556383538, 32340464080, 26839546881, 28274964409);
> f2(11699736551, 15894932166, 19139950582, 13633454580, 22482572207);
> f2(30764548671, 17526964713, 23697719493, 9524631408, 30370932766);
> f2(18119046956, 23097876579, 26098072062, 15396294352, 9541609838);
> f2(28634401814, 16984389142, 21492627645, 27836629811, 22527509738);
> f2(4305530608, 6773867526, 24829655696, 9104967989, 20189846689);
> f2(1004309963, 10811232130, 26068311356, 24187440691, 5297946119);
> f2(24088028176, 29109983791, 24001932812, 3662077356, 24934261930);
> f2(32639102117, 3020599620, 5961831045, 20648136101, 23733757541);
> f2(25100263375, 29269428215, 17595617668, 6200302235, 24720196327);
> f2(11240756404, 17877512273, 32863039105, 31978979401, 33734611283);
```

$> f2(2985375672, 29140255085, 22961776423, 17230783761, 1635540387);$

$> f2(22959523022, 29130026286, 26514514080, 10953229047, 14225594501);$

$> f2(28117084809, 30035665225, 4606595517, 19439152434, 22208681761);$

$> f2(25856752904, 30992412400, 13678373705, 3029686195, 7034407841);$

$> f2(5617638763, 18456243247, 29693163119, 3193465741, 5342271451);$

$> f2(11933015043, 14735131467, 935278652, 19362184933, 23870488939);$

$> f2(30953628138, 33754267657, 7234980540, 24125690529, 6704694846);$

$> f2(26807768629, 2421005005, 11672689060, 12612471050, 7416416775);$

$> f2(23608188174, 22875178864, 2648814876, 13258662651, 22760640562);$

$> f2(20375705470, 16293580593, 28694829662, 9220003969, 24415115697);$

$> f2(32586107663, 20085364361, 23081248834, 26185797967, 3796994105);$

$> f2(20308914030, 14656979175, 31391122327, 29325542221, 9210796836);$

$> f2(34291562620, 986965170, 7636251701, 6647492993, 30645392424);$

$> f2(18040349274, 21234088296, 22584876868, 9428614741, 7461815255);$

$> f2(2780173462, 26047958692, 5521615755, 2176809937, 29201733450);$

$> f2(3292464781, 18678730090, 25290642761, 3042008215, 5469500243);$

$> f2(30541529954, 22788890346, 11750623938, 15347859749, 10563080548);$

$> f2(11711038781, 18835622021, 8664311051, 33853698792, 1243533997);$

$> f2(16665122934, 20519242249, 23234264717, 33787616521, 19952629610);$

$> f2(2455956988, 33972022083, 1095123919, 24094188363, 2260470933);$

$> f2(33149864192, 27794528605, 1689770070, 16420636563, 6684092817);$

$> f2(2431407503, 19763365222, 17298504524, 16592445344, 14394948898);$

$> f2(14544543999, 21011809219, 24425993578, 14503759432, 15607628376);$

$> f2(33228228130, 10169135019, 25141985504, 20917754113, 7714560495);$

$> f2(21537548005, 8197187380, 3201944461, 14549234193, 29457199320);$

$> f2(2512901923, 8311606683, 25596925671, 26254093034, 31385691105);$

$> f2(6840129191, 9017241224, 28264740347, 20979919810, 7001033213);$

$> f2(8970494200, 21310908880, 15647713716, 12446451879, 1336521211);$

$> f2(29217060738, 18164881929, 24305595646, 9974773427, 8018231646);$

$> f2(33569626379, 21432003093, 10757996978, 30799562994, 8255806475);$

$> f2(18280008498, 14502911116, 6060492693, 26200280649, 29865133519);$

$> f2(32124563306, 11171312208, 9264063957, 30564625099, 21606772025);$

$> f2(33062209310, 4072382009, 5586216002, 16455365362, 26825104022);$

$> f2(13843517048, 29961786774, 19057697327, 29743316152, 27809552944);$

$> f2(15071613149, 15398067358, 28873282687, 8314103867, 33850389748);$

$> f2(9575937472, 30941667008, 10249192587, 31832268955, 12283176240);$

$> f2(23120720877, 29932431330, 33684991275, 9440924679, 30483983316);$

$> f2(20251796980, 27070074139, 29343696459, 10950651587, 7379285488);$

$> f2(33118513398, 9323679565, 18073590958, 31142641132, 16596049229);$

$> f2(33330473351, 26263793511, 20102378132, 34214985983, 32905762761);$

$> f2(705914884, 13286760120, 24666617758, 31988142230, 3968560594);$

$> f2(15715763343, 771398587, 24045650207, 29987093719, 29746811525);$

$> f2(26480266695, 3720668321, 5675156441, 6578750113, 31112386650);$

$> f2(16214193692, 23582090929, 31778762526, 867975318, 3789581477);$

$> f2(23137399459, 29191170766, 28164708428, 29114304717, 9990260102);$

$> f2(8654612413, 23649214000, 17179601785, 13070172736, 20777388342);$

$> f2(30902805042, 667619388, 10291922849, 12630524322, 26262022311);$

$> f2(20994982270, 7515526063, 21684999684, 8713331816, 19981948164);$

$> f2(23645766908, 26512155692, 14251741122, 26121740176, 25781354607);$

$> f2(187151467, 512727062, 882936658, 216155661, 5951098);$

$> f2(2249578214, 26845403774, 33622067872, 27686342546, 8029638839);$

$> f2(25174990522, 5368486239, 2627987427, 10193058027, 770790797);$

$> f2(19573575888, 29792636622, 6655475156, 30078385063, 16317485054);$

$> f2(32724121293, 18454795901, 9510851282, 22768552959, 9117542946);$

$> f2(4486070256, 5728827071, 8158899017, 15580972448, 14465476329);$

$> f2(21399323099, 8008361093, 26648734985, 1001407134, 19320173251);$

$> f2(11981742408, 1629299459, 20242405145, 9096981546, 8044000619);$

$> f2(32871884968, 21644170474, 28554451524, 34102138480, 18282717298);$

```
> f2(21219732806, 3175054117, 11749075511, 25253431893, 28604715207);

> f2(26209915518, 31087246046, 17505804873, 21904851901, 31747916274);

> f2(28365071765, 33345678395, 32539975519, 6524736265, 5723339307);

> f2(22557578244, 4227377804, 20737243791, 2579132710, 33478459395);

> f2(7283409942, 10232857979, 10949707700, 25857023322, 18139184768);

> f2(4952580198, 15447694199, 18049303438, 29312112832, 9458392075);

> f2(20600259438, 7551257401, 22886844577, 27430335018, 964625889);

> f2(8834213897, 19669100953, 23015067624, 1211626194, 26379904854);

> f2(25639898220, 12013283364, 20756223322, 5389655669, 8479080374);

> f2(10101185477, 25286386239, 12392723454, 29692953661, 18354897115);

> f2(30518451110, 10345515008, 9488029445, 14071173582, 23184165449);

> f2(33816140068, 14083858908, 20553458602, 8820575374, 13425277750);

> f2(20068886965, 21277583361, 8979758451, 5856372953, 20839718364);

> f2(8746784991, 1354879287, 23238294784, 16079331355, 11122871927);

> f2(15279354154, 27798263854, 33833868526, 22652481045, 2514005621);

> f2(7547192496, 20087001649, 11465886328, 26146677751, 11361743356);
```

Experiment 4

$k = 3, \text{length} = 5$

(1)

```
> su:=proc()

> local u1, u2, u3, u4, u5, w1, w2, w3, w4;

> for u1 from 1 to 7 do

> for u2 from 1 to 7 do

> for u3 from 1 to 7 do

> for u4 from 1 to 7 do

> for u5 from 1 to 7 do

> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 2) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(2)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 3) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(3)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 4) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
```

```
> su();
```

(4)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 5) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(5)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 6) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(6)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 7 do
> for u2 from 1 to 7 do
> for u3 from 1 to 7 do
> for u4 from 1 to 7 do
> for u5 from 1 to 7 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 7) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

Experiment 5

$k = 4, \text{length} = 5$

(1)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 2) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(2)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 3) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(3)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 4) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(4)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 5) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(5)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 6) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(6)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 7) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(7)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 8) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(8)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 9) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(9)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 10) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

(10)

```
> su:=proc()
> local u1, u2, u3, u4, u5, w1, w2, w3, w4;
> for u1 from 1 to 15 do
> for u2 from 1 to 15 do
> for u3 from 1 to 15 do
> for u4 from 1 to 15 do
> for u5 from 1 to 15 do
> w1 := gcd(u1, u2);
> w2 := gcd(w1, u3);
> w3 := gcd(w2, u4);
> w4 := gcd(w3, u5);
> if (w4 = 11) then
> print(u1, u2, u3, u4, u5, w4);
> fi;
> od;
> od;
> od;
> od;
> od;
> end:
> su();
```

Bibliography

[1] S. Blackburn, and S. Galbraith, *Cryptanalysis of two cryptosystems based of group actions*, Lecture Notes in Comput. Sci., 1716, 52–61, 1999.

[2] J.A. Buchmann, *Introduction to cryptography*, Springer, 2001.

[3] N. Ferguson, and B. Schneier, *Practical cryptography*, Wiley, 2003.

[4] J.B. Fraleigh, *A first course in abstract algebra*, Addison-Wesley, 1989.

[5] M.I. Gonzalez Vasco, and R. Steinwandt, *A reaction attack on a public-key encryption based on the word problem*, Appl. Algebra Eng. Com. Comput. 14(5), 335-340, 2004.

[6] D. Grigoriev, and I. Ponomarenko, *Homomorphic public-key cryptosystems and encrypting boolean circuits*, arXiv:math.cs.CR/0301022, 2003.

[7] D. Grigoriev, and I. Ponomarenko, *Homomorphic public-key cryptosystems over groups and rings*, Quaderni di Mathematica, vol. 13, 305-325, 2004.

[8] D. Grigoriev, and I. Ponomarenko, *Constructions in public-key cryptography over matrix groups*, to appear in Contemporary Math. AMS.

[9] D.F. Holt, *Handbook of computational group theory*, Chapman & Hall/CRC, 2005.

[10] http://en.wikipedia.org.

[11] http://eprint.iacr.org/2005/070.pdf.

[12] http://mathworld.wolfram.com.

[13] http://www.ms.unimelb.edu.au/ cfm/notes/cgt-notes.pdf.

[14] R.C. Lyndon, and P.E. Schupp, *Combinatorial group theory*, Springer-Verlag, 1977.

[15] R.A. Mollin, *An introduction to cryptography*, Chapman & Hall/CRC, 2001.

[16] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997.

[17] M. Newman, *Integeral matrices*, Academic Press, 1972.

[18] B. Schneier, *Applied cryptography*, John Wiley & Sons, Inc, 1994.

[19] G.J. Simmons, *Contemporary cryptology the science of information security*, IEEE Press, 1992.

[20] R. Steinwandt, *Loopholes in two public-key cryptosystems using the modular groups*, Preprint Universitaet von Karlsruhe, 2000.

[21] D.R. Stinson, *Cryptography theory and practice*, Chapman & Hall/CRC, 2006 .

[22] N. Wagner, and M. Magyarik, *A public-key cryptosystem based on the word problem*, Lecture Notes in Compt. Sci., 196, 19-36, 1985.

[23] A. Yamamura, *Public-key cryptosystems using the modular groups*, Lecture Notes in Compt. Sci., 1431, 203-216, 1998.

[24] A. Yamamura, *A functional cryptosystem using a group action*, Lecture Notes in Compt. Sci., 1587, 314-325, 1999.