2016

Topics on Register Synthesis Problems

Weihua Liu
University of Kentucky, liuweihua817@gmail.com
Digital Object Identifier: http://dx.doi.org/10.13023/ETD.2016.160

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Weihua Liu, Student

Dr. Andrew Klapper, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

Topics on Register Synthesis Problems

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Weihua Liu
Lexington, Kentucky
Director: Dr. Andrew Klapper, Professor of Computer Science
Lexington, Kentucky
2016

ABSTRACT OF DISSERTATION

Topics on Register Synthesis Problems

Pseudo-random sequences are ubiquitous in modern electronics and information technology. High speed generators of such sequences play essential roles in various engineering applications, such as stream ciphers, radar systems, multiple access systems, and quasi-Monte-Carlo simulation. Given a short prefix of a sequence, it is undesirable to have an efficient algorithm that can synthesize a generator which can predict the whole sequence. Otherwise, a cryptanalytic attack can be launched against the system based on that given sequence.

Linear feedback shift registers (LFSRs) are the most widely studied pseudorandom sequence generators. The LFSR synthesis problem can be solved by the Berlekamp-Massey algorithm [44], by constructing a system of linear equations, by the extended Euclidean algorithm, or by the continued fraction algorithm [52, 53]. It is shown that the linear complexity is an important security measure for pseudorandom sequences design. So we investigate lower bounds of the linear complexity of different kinds of pseudorandom sequences.

Feedback with carry shift registers (FCSRs) were first described by Goresky and Klapper [22, 31]. They have many good algebraic properties similar to those of LFSRs. FCSRs are good candidates as building blocks of stream ciphers. The FCSR synthesis problem has been studied in many literatures [7, 30, 33] but there are no FCSR synthesis algorithms for multi-sequences. Thus one of the main contributions of this dissertation is to adapt an interleaving technique to develop two algorithms to solve the FCSR synthesis problem for multi-sequences.

Algebraic feedback shift registers (AFSRs) are generalizations of LFSRs and FCSRs. Based on a choice of an integral domain R and $\pi \in R$, an AFSR can produce sequences whose elements can be thought of elements of the quotient ring $R/(\pi)$. A

modification of the Berlekamp-Massey algorithm, Xu's algorithm solves the synthesis problem for AFSRs over a pair (R, π) with certain algebraic properties [33]. We propose two register synthesis algorithms for AFSR synthesis problem. One is an extension of lattice approximation approach but based on lattice basis reduction and the other one is based on the extended Euclidean algorithm.

KEYWORDS: FCSRs, AFSRs, Register synthesis problem, Multi-sequences.

Weihua Liu

May 6, 2016

Topics on Register Synthesis Problems

By

Weihua Liu

Dr. Andrew Klapper

Director of Dissertation

Dr. Miroslaw Truszczynski

Director of Graduate Studies

May 6, 2016

Date

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my advisor Dr. Andrew Klapper for his patient guidance and continuous support. I could not have imagined having a better advisor and mentor for my Ph.D. study. His advice on both research as well as on my career have been priceless. I would also like to thank my Ph.D. Committee Members, Dr. Judy Goldsmith, Dr. Miroslaw Truszczynski and Dr. Heide Gluesing-Luerssen for their invaluable advice and direction throughout the years. In addition, I would like to thank Dr. Andrew Klapper, Dr. Judy Goldsmith, and Dr. Debby Keen for their kind assistance with writing recommendation letters and helping me in my job search.

I am very grateful to my research collaborator, Dr. Zhixiong Chen, for his scientific advice and knowledge. Thanks also go to my colleges and all members in the group of Crypto seminar, Dr. Ting Gu, Mr. Virgil Barnard, Dr. Xiaoni Du and Dr. Zhihua Niu for their insightful discussions and suggestions.

My Ph.D. would not have been possible without my family and friend's support and encouragement. Words cannot express how grateful I am to my mother and father for all of the sacrifices that they've made on my behalf. I would also like to thank Dr. Tom Tucker and Mrs. Nancy Tucker for their endless love and supports.

Contents

List of Figures

1 Introduction

This thesis concerns pseudorandom sequence generators and the problem of finding minimal generators of certain types given only partial knowledge of the sequence. This problem has implications for symmetric key cryptography. In the remainder of this chapter we review the definition of pseudorandom sequences, the basics of the type of generator we are concerned with, and the introduction of lattice theory.

1.1 Pseudorandom sequences

Random sequences are useful for a variety of purposes, such as generating encryption keys, gambling, statistical sampling and computer simulation. The randomness means that it is hard to predict the next number using the numbers that we have seen. Truly random sequences can be generated by true random number generators (TRNGs), such as HotBits [69] using radioactive decay and RANDOM.ORG [1] using atmospheric noise. However, TRNGs are nondeterministic and generally are inefficient in most practical environments. In such situations, pseudorandom sequence generators are substituted for TRNGs.

Pseudorandom sequence generators are deterministic algorithms which produce sequences, called pseudorandom sequences, that are apparently random. These sequences are not truly random but apparently random in the sense that it is not efficient for an adversary to distinguish them from the truly random sequences of the same length. To gain confidence in the randomness of pseudorandom sequences, a collection of statistical tests is designed to detect the specific characteristics expected of random sequences. Besides the five basic tests [45] listed blow, other statistical tests have been proposed, such as Golomb's randomness postulates [21], Maurer's universal statistical test [45] and FIPS 140-1 statistical tests [17, 45].

- **Frequency test**: Determine whether the number of 0's and 1's in the sequence are approximately the same.
- **Serial test**: Determine whether the number of occurrences of 00, 01, 10, and 11 as subsequences are approximately the same.
- **Poker test**: Determine whether the sequences of length m each appear approximately the same number of times in the sequence.
- **Runs test**: Determine whether the number of runs (i.e. subsequences that contains consecutive 0's or consecutive 1's which are neither preceded nor suc-

ceeded by the same symbol) of various lengths in the sequence is as expected for a random sequence.

- **Autocorrelation test**: Check for correlations between the sequence and its shifts.

Along with the property of randomness, pseudorandom sequences with particular statistical properties are used in different applications. In frequency hopping spread spectrum, pseudorandom sequences known to both transmitter and receiver are used as spreading codes to lead signals rapidly to switch among many frequency channels. This method has been used in many wireless communication systems, such as bluetooth, cellphone, and GPS systems. Pseudorandom sequences are also used as error correcting codes in satellite and other communications. In stream ciphers and other cryptographic applications, pseudorandom sequences are used as crucial components for generating key streams. In Monte Carlo methods, pseudorandom sequences that are uniformly distributed are used as samples data for simulation.

Considering the security problems, not all pseudorandom sequence generators are suitable for use in cryptography. NIST Special Publication 800-22, A Statistical Test for Random and Pseudorandom Number Generators for Cryptographic Applications [57] discusses the selecting and testing of random and pseudorandom sequence generators in cryptography and offers detailed recommendations on how to use these tests.

1.2 The one-time pad and stream ciphers

The one-time pad is a cipher in which each character in the plaintext is encrypted with a random key. It was first described by Frank Miller in 1882 [47]. In 1917, AT&T research engineer, Gilbert Vernam, re-invented the electrical one-time pad using the XOR operation (addition modulo 2) and got it patented in 1919 (U.S. Patent 1,310,719 [67]). It was proved by Claude Shannon, using information theory, that the one-time pad is mathematically unbreakable [62]. Even adversaries with unbounded computational power and infinite time cannot break it. It is said to have *perfect secrecy*, which means that the ciphertext gives absolutely no additional information about the plaintext. However, it is inconvenient to use a one-time pad in practice due to the constraints that the key must be at least as long as the plaintext and each key can be used only once. A well-designed stream cipher can be a good replacement for one-time pad.

A stream cipher is a private (symmetric) key cryptosystem where encryption and decryption keys are identical. Stream ciphers encrypt plaintext character by character by adding the key stream generated by a pseudorandom sequence generator. The decryption is to subtract the identical copy of the key stream character by character from the ciphertext. Figure 1.1 shows the procedure of encryption and decryption in a stream cipher in the form of binary digits.

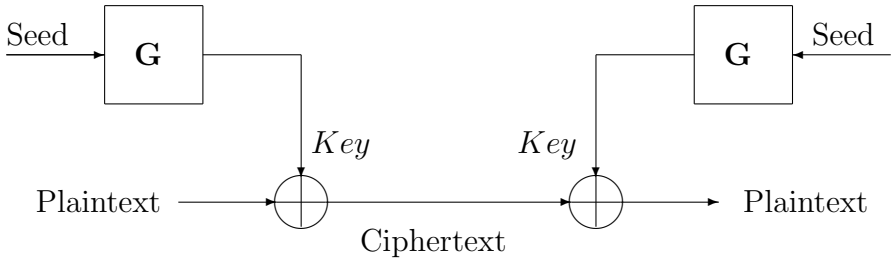

Figure 1.1: Stream Cipher Schematic

Because of their high speed stream ciphers are suitable for transmitting large amounts of data, such as that generated by digital telephones, video on demand, and Voice over Internet Protocol (VoIP). They are often implemented in hardware to add speed. The most widely used stream cipher is RC4, which is used in WEP (security algorithm for IEEE 802.11 wireless networks), SSL (cryptographic protocols to security communication over computer network) and SSH (network protocol for remote login over an unsecured network). In November 2004, ECRYPT (European Network of Excellence in Cryptology) launched a four year project called "eSTREAM" to advance the development of stream cipher designs.

1.3 Sequence generators and their Properties

In this section, we recall three kinds of sequence generators: linear feedback shift registers (LFSRs), feedback with carry shift registers (FCSRs) and algebraic feedback shift registers (AFSRs). The design and analysis of LFSRs and FCSRs are based on similar algebraic structures, which gives rise to a common generalization, AFSRs [32].

LFSRs are widely used in cryptography. There are many important LFSR-based stream ciphers such as A5 used in GSM, and E0 used in Bluetooth. With the right

choices of coefficients, LFSRs produce pseudorandom sequences with desirable randomness properties.

Algebraic attacks [13, 36] on stream ciphers based on LFSRs take advantage of the linear nature of the state change operation. If a sequence is annihilated by a low degree polynomial in the shift operator, then it is also annihilated by the composition of this polynomial with the state change operator. Iteration gives us many more annihilators of the sequence. This allows an adversary to break the stream cipher by solving a system of low degree polynomials. Thus alternatives to LFSRs as building blocks are desirable.

FCSRs, proposed by Klapper and Groresky [22,31], are good alternatives to LFSRs as building blocks to proffer resistance to algebraic attacks. The sequences generated by FCSRs enjoy many useful statistical properties. FCSRs are high speed sequence generators which are suitable for hardware implementation. The stream cipher family Filtered-FCSR (F-FCSR) [3–6] is an example of stream ciphers based on FCSRs.

Generally speaking, a sequence generator is an algorithm for generating sequences of numbers. Different state changes determine different properties of the generated sequences.

Definition 1.3.1. *[25] A sequence generator with output,*

$$F = (U, \Sigma, f, g),$$

consists of a discrete (i.e., finite or countable) set U of states, a discrete alphabet Σ of output values, a state transition function $f : U \to U$ and an output function $g : U \to \Sigma$.
Such a generator is depicted as follows:

$$f \hookrightarrow U \xrightarrow{g} \Sigma.$$

Given an initial state $\boldsymbol{s} \in U$, such a sequence generator outputs an infinite sequence

$$F(\boldsymbol{s}) = g(\boldsymbol{s}), g(f(\boldsymbol{s})), g(f^2(\boldsymbol{s})), \cdots$$

with elements in Σ.

The output sequence $\mathbf{a}=(a_0, a_1, \cdots)$ is *periodic* if there exists an integer $T > 0$ so that

$$a_i = a_{i+T} \tag{1.1}$$

for all $i = 0, 1, 2, \cdots$. We call T a *period* of the sequence $\mathbf{a}$ and the least one is called the *least period* of $\mathbf{a}$. The sequence $\mathbf{a}$ is *eventually periodic* if there exists $N > 0$ and

$T > 0$ so that (1.1) holds for all $i \geq N$. It is well known that every period of $\mathbf{a}$ is a multiple of the least period of $\mathbf{a}$.

Consider the Fibonacci sequence

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \cdots .$$

The sequence generator for the Fibonacci sequence is with states $U = \mathbb{Z}^2$, output alphabet $\mathbb{Z}$, state change function $f(a_0, a_1) = (a_1, a_0+a_1)$, output function $g(a_0, a_1) = a_0$, and initial state $\mathbf{s} = (0, 1)$.

1.3.1 Linear Feedback Shift Registers

Linear feedback shift registers (LFSRs) are high speed generators of linearly recurrent sequences that have many desired properties for applications including cryptography, random number testing and wireless communication systems employing spread spectrum or CDMA techniques. They provide a fast and simple method of generating pseudo-random sequences. Although binary LFSRs are most widely used, we here consider the general case that the alphabet is a finite commutative ring. We assume that R is a finite commutative ring (with identity denoted by 1).

Definition 1.3.2. *[25] A (*Fibonacci mode*) linear feedback shift register of* length *m over* R*, with coefficients $q_1, q_2, \cdots q_m \in$ R is a sequence generator whose state is an element*

$$s = (a_0, a_1, \cdots , a_{m-1}) \in R^m,$$

whose output is $\mathbf{out}(s) = a_0$*, and whose state change operation τ is given by*

$$(a_0, a_1, \cdots , a_{m-1}) \to (a_1, a_2, \cdots , a_{m-1}, \Sigma_{i=1}^{m} q_i a_{m-i}).$$

A circuit presentation of a Fibonacci LFSR is shown in Figure 1.2.

Definition 1.3.3. *[25] A (*Galois mode*) linear feedback shift register of* length *m over* R*, with coefficients $q_1, q_2, \cdots q_m \in$ R is a sequence generator whose state is an element*

$$s = (h_0, h_1, \cdots , h_{m-1}) \in R^m,$$

whose output is $\mathbf{out}(s) = h_0$*, and whose state change operation τ is given by*

$$(h_0, h_1, \cdots , h_{m-1}) \to (a_1 + q_1 h_0, h_2 + q_2 h_0, \cdots , h_{m-1} + q_{m-1} h_0, q_m h_0).$$

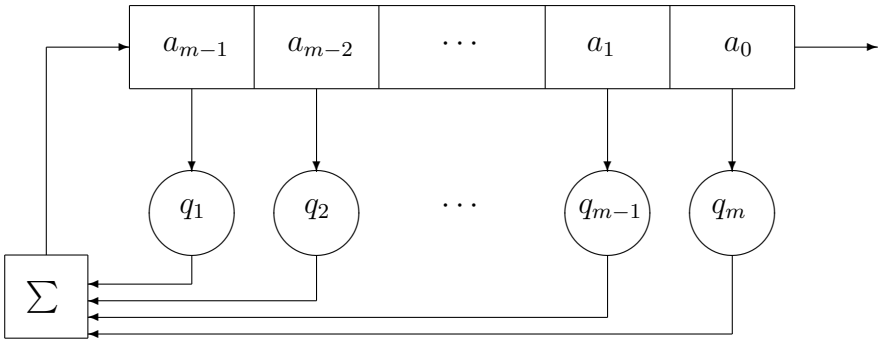

Figure 1.2: A Linear Feedback Shift Register of Length m

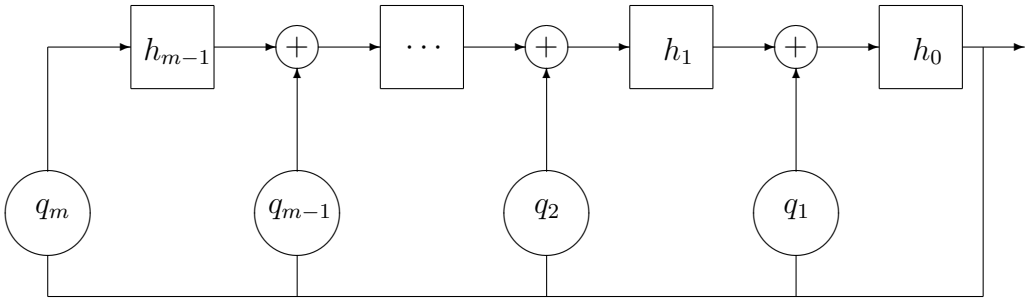

Figure 1.3: Galois LFSR

It is illustrated in Figure 1.3.

Galois LFSRs and Fibonacci LFSRs are equivalent. That is, given identical coefficients, they can produce the same sequence. However, the initial states of the two implementations may be different for the two sequences to be identical. When implemented in hardware, Galois LFSRs are generally faster than Fibonacci LFSR because the additions are performed in parallel by separate adders that will result in a potentially lower clock cycle time. So the Galois form is usually preferred in applications especially in hardware.
There are many useful results about LFSRs and LFSR sequences [25]:

1. The coefficients $q_1, q_2, \ldots, q_m$ can be associated to the *connection polynomial*

$$q(x) = -1 + \sum_{i=1}^{m} q_i x^i \in R[x].$$

Many properties (such as the period) of the output sequence can be determined from this polynomial.

2. Any infinite sequence $\mathbf{a} = (a_0, a_1, a_2 \cdots)$ over R may be identified with its *generating function* $a(x) = \sum_{i=0}^{\infty} a_i x^i$, which is an element of the ring of $R[[x]]$ of formal power series. It is well known that the sequence $\mathbf{a}$ is eventually periodic if and only if its generating function is equal to a quotient of two polynomials,

$$a(x) = \frac{f(x)}{q(x)} \in R[[x]].$$

Now let $q(x)$ be any polynomial with constant term -1. Then $q(x)$ is the connection polynomial for a LFSR which generates $\mathbf{a}$ and $f(x)$ is uniquely determined by the initial loading of this LFSR. The sequence $\mathbf{a}$ is strictly periodic if and only if $\deg(f(x)) < \deg(q(x))$. If the ring R is finite and if $q_m \neq 0$, then every output sequence is strictly periodic.

3. The output sequence $\mathbf{a} = (a_0, a_1, a_2 \cdots)$ of an LFSR can be represented as

$$a_i = x^{-i} f(x) \pmod{q(x)} \pmod{x},$$

where $a(x)$ is the generating function, $q(x)$ is the connection polynomial and $a(x) = -f(x)/q(x)$.

4. The number of cells in the shortest LFSR that can generate $\mathbf{a}$ is called the *linear complexity* or equivalently *linear span* of $\mathbf{a}$; denoted by $\lambda(\mathbf{a})$. It is an important measure of the cryptographic security of the sequence. By the Berlekamp-Massey algorithm, for a given output sequence $\mathbf{a}$, we can reconstruct a generating LFSR with only $2\lambda(\mathbf{a})$ consecutive bits of $\mathbf{a}$. So for the sake of security, we need to use sequences with high linear complexity.

5. A sequence is an *m-sequence* if it is the output sequence of a LFSR that cycles through all possible nonzero states before it repeats. Such sequences have many important statistical properties and have found applications in communications, coding theory, radar system and CDMA. If the ring R is a field, then $\mathbf{a}$ is an m-sequence if and only if the connection polynomial $q(x)$ of its generating LFSR is a primitive polynomial in $R[x]$.

1.3.2 Feedback with Carry Shift Registers

LFSRs are widely used because they can be easily implemented in hardware but using LFSRs alone can not guarantee good security. So many schemes have been proposed

to increase the security of LFSRs. One approach is to combine the outputs of several parallel LFSRs with a non-linear Boolean function or pass the entire state of a single LFSR into a non-linear filtering function. The non-linear function used here should be chosen very carefully according to several criteria in order to avoid correlation attacks and other cryptanalysis. Another approach is to have the LFSR clocked by the output of a second LFSR. However, the appearance of algebraic attacks raised a very challenging problem for stream ciphers based on these LFSR-generators [13, 36]. The main idea behind this method is finding and solving a system of multivariate polynomial equation over a finite field. Generating such a system is usually based on the linear structure of LFSRs. So one good choice is to substitute LFSRs with FCSRs.

FCSRs were first described by Goresky and Klapper [22, 31]. They are similar to LFSRs, so they also can be implemented to be very fast, especially in hardware. The main difference is the fact that the additions are not simple additions but additions with propagation of carries. Since they were introduced, the properties of the sequences generated by FCSRs have been well studied from a mathematical point of view.

Definition 1.3.4. *Fix an integer $N > 1$. Let $S = \{0, 1, \cdots, N-1\}$. Let $q_1, q_2, \cdots, q_m \in S$. An N-ary feedback with carry shift register of length m with multipliers or taps $q_1, q_2, \cdots, q_m$ is a discrete state machine whose state is a collection $(a_0, a_1, \cdots, a_{m-1}; z)$ where $a_i \in S$ and $z \in \mathbb{Z}$ and whose state change operation is described as follows:*

- *Compute the integer sum*
$$\sigma = \sum_{i=1}^{m} q_i a_{m-i} + z.$$
- *Replace $(a_0, a_1, \cdots, a_{m-1}; z)$ by $(a_1, a_2, \cdots, a_{m-1}, \sigma \pmod N; \sigma(divN))$.*

It is convenient to think of a FCSR as a physical circuit, as in Figure 1.4.

As with LFSRs, there is a Galois form of FCSRs. The state of a Galois FCSR is $(a_0, a_1, \cdots, a_{m-1}; c_1, ..., c_m)$, where $a_i, c_j \in S$ and the change of states can be described as follows:

- Calculate $\delta_m = c_m + q_m a_0 \quad \text{and} \quad \delta_j = c_j + a_j + q_j a_0 \ \text{ for } \ 1 \leq j < m.$
- The new values are
$$a'_{j-1} = \delta_j \pmod N, \quad c'_j = \delta_j \ (\text{div}N).$$

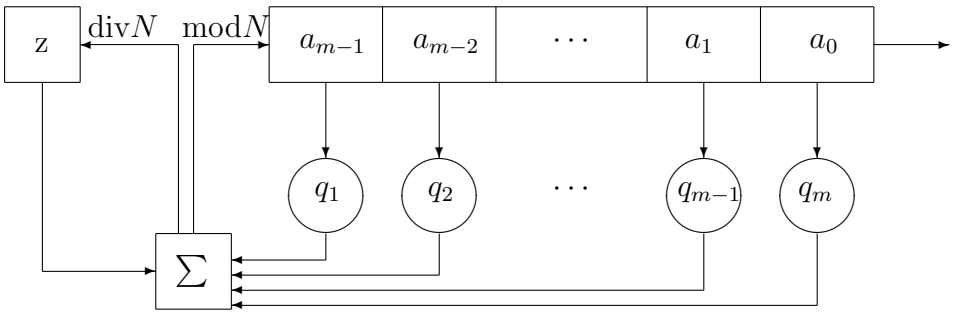

Figure 1.4: A Feedback with Carry Shift Register of Length m

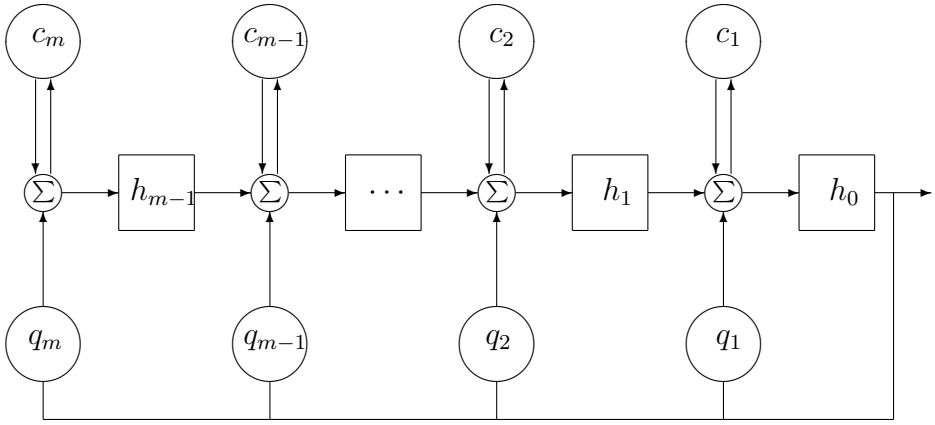

Figure 1.5: Galois FCSR

This procedure is illustrated in Figure 1.5

In a Fibonacci FCSR, all the feedback bits influence a single cell while in Galois mode, a single feedback bit influences all the cells. As noticed, the Fibonacci mode is not suitable for cryptographic applications because most of the cells of a Fibonacci FCSR have a linear transition function [18]. As with the Galois LFSR, the additions are performed in parallel in Galois FCSRs, so Galois FCSRs perform better in applications.

The most studied stream cipher based on FCSRs is F-FCSR. At the FSE 2005, F. Arnault and T. P. Berger proposed several stream ciphers based on FCSRs, called F-FCSR-SF1, F-FCSR-SF8, F-FCSR-DF1 and F-FCSR-DF8 [6]. Later for eSTREAM project, they submitted two new algorithms called F-FCSR-8 and F-FCSR-H. Af-

ter three phases of evaluation, the F-FCSR family (F-FCSR-H v2 and F-FCSR-16) was recommended as one of the eight algorithms selected for ECRYPT eSTREAM portfolio at the time of April 15, 2008 and it tended to lie in the top half of most hardware performance classifications. Unfortunately, Hell and Johansson presented a severe cryptanalytic attack on the F-FCSR stream cipher family, because of which F-FCSR was removed from the final eSTREAM portfolio [26]. At Indocrypt 2007, Arnault, Berger, Lauradoux, and Minier presented two new constructions for software implementations, called X-FCSR-128 and X-FCSR-256 [4]. The main idea was to use two optimal 256 bit FCSRs clocked with different directions and a nonlinear extraction function. The X-FCSR-128 outputs 128 bits at each iteration and runs at 8.2 cycles/byte. The X-FCSR-256 outputs 256 bits at each iteration and has a better performance with 6.5 cycles/byte. They are comparable to the fastest known stream ciphers. The X-FCSR family of stream ciphers was attacked by Stankovski, Hell and Johansson by a state recovery method [64, 65]. They observed that a sufficient amount of consecutive zero feedback bits will eventually make the carry registers contain only zeros so that FCSRs can be treated as LFSRs. Arnault et al. introduced a new representation of FCSR, called ring FCSR, that has better diffusion properties to proffer better resistance to the state recovery attack [5]. Although the F-FCSR stream cipher is no longer in the final eSTREAM portfolio, the design and analysis of stream ciphers based on FCSRs are still interesting problems.

N-adic numbers and FCSR sequences

The analysis of FCSRs is based on N-adic numbers which were discovered by K. Hensel around 1900. There are several books about p-adic numbers and p-adic analysis [34, 55].

Definition 1.3.5. *An N-adic integer is an infinite expression*

$$a = a_0 + a_1 N + a_2 N^2 + \cdots ,$$

where $a_0, a_1, \cdots \in \{0, 1, \cdots, N-1\}$. The set of N-adic integers is denoted by $\mathbb{Z}_N$. The least degree of a nonzero N-adic integer $a = \sum_{i=0}^{\infty} a_i N^i$ is the least index i such that $a_i \neq 0$.

If $a = \sum_{i\geq 0}^{\infty} a_i N^i$, $b = \sum_{i\geq 0}^{\infty} b_i N^i$, we have

$$a = b \Longleftrightarrow \quad \text{for all} \quad i \geq 0, \quad a_i = b_i.$$

Addition in $\mathbb{Z}_N$ is performed by "carrying" overflows to higher terms, i.e.,

$$\underbrace{N^m + \cdots + N^m}_{N} = N^{m+1}, \quad \text{for all } m \in \mathbb{N}.$$

$\mathbb{Z}_N$ is a ring with additive identity 0 and multiplicative identity 1. It can be seen that

$$-1 = (N-1) + (N-1)N + (N-1)N^2 + (N-1)N^3 + \cdots \tag{1.2}$$

since $1 + (-1) = 0$. Calculating the additive inverse can be done as follows. Let $a \in \mathbb{Z}_N$ have least degree d. That is, $a = \sum_{i=d}^{\infty} a_i N^i$ with $1 \le a_d \le N-1$. Then

$$-a = (N - a_d)N^d + \sum_{i=1}^{\infty} (N - a_i - 1)N^i. \tag{1.3}$$

Let $a = \sum_{i \ge 0}^{\infty} a_i N^i \in \mathbb{Z}_N$. Then a is invertible in $\mathbb{Z}_N$ if and only if a_0 is relatively prime to N.

The set of *N-adic numbers*, denoted by $\mathbb{Q}_N$, consists of infinite sums

$$a = a_{-m}N^{-m} + a_{-m+1}N^{-m+1} + \cdots + a_0 + a_1 N + \cdots$$

with coefficients $0 \le a_i \le N-1$. It contains $\mathbb{Z}_N$ as a subring. We have $\mathbb{Q}_N = S^{-1}\mathbb{Z}_N$ where $S = \{N, N^2, N^3, \cdots\}$. Notice that if N is a power of a prime number, then $\mathbb{Z}_p$ is an integral domain and $\mathbb{Q}_p$ is its fraction field. That is, $\mathbb{Q}_p = S^{-1}\mathbb{Z}_p$ where $S = \mathbb{Z}_p^{\times}$ consists of all nonzero elements. For composite N, the ring $\mathbb{Z}_N$ has zero divisors and the ring $\mathbb{Q}_N$ is not a field.

There are many parallels between LFSR sequences and FCSR sequences [25].

1. The m taps $q_1, q_2, \cdots, q_m$ of a FCSR of length m define a *connection integer*

$$q = q_m N^m + q_{m-1}N^{m-1} + \cdots + q_1 N - 1.$$

   The period (and many other properties) of the FCSR sequence may be expressed in terms of number-theoretic properties of this integer.

2. Any infinite sequence $\mathbf{a} = (a_0, a_1, \cdots)$ over $\mathbb{Z}/(N)$ can be identified with the formal power series, $a = \sum_{i \ge 0}^{\infty} a_i N^i$ which is an element of the ring of $\mathbb{Z}_N$. Sequence $\mathbf{a}$ is an eventually periodic N-ary sequence if and only if the associated N-adic number a is a quotient of two integers,

$$a = \frac{f}{q} \in \mathbb{Z}_N.$$

The denominator q or $-q$ is the connection integer of a FCSR which generates the sequence $\mathbf{a}$. The denominator f determine the initial loading of this FCSR. Assume that $q > 0$. The sequence $\mathbf{a}$ is strictly periodic if and only if $-q \leq f \leq 0$. If $\mathbf{a}$ is strictly periodic then for all i,

$$a_i = N^{-i} f \pmod{q} \pmod{N}.$$

3. As in the case of linear span, the *N-adic span* is intended to measure how large an FCSR is required to generate a given eventually periodic sequence $\mathbf{a}$. In the LFSR case, the linear complexity is the number of cells in the smallest LFSR that outputs $\mathbf{a}$ and coincides with degree of the connection polynomial. But in the FCSR case, the number of N-ary coefficients in the connection integer only coincides with the size of the basic register and additional space is required for the memory. So the *N-adic span* of an eventually periodic sequence $\mathbf{a}$ is the number of cells in the register plus the number of elements needed for the memory of an FCSR which outputs the sequence $\mathbf{a}$. The *N-adic complexity* is the real number $\Phi_N(\mathbf{a}) = \log_N(\max(|p|, |q|))$ where p/q is the fraction of sequence $\mathbf{a}$ in lowest terms. It has been proved that the difference between N-adic span and N-adic complexity is bounded by $\log_N(\Phi_N(\mathbf{a})) + 2$ [25]. From a mathematical viewpoint, it is easier to analyze the N-adic complexity.

   The N-adic complexity is a useful measure in the study of the security of pseudorandom sequences for cryptographic application. Based on De Weger and Mahler's rational approximation theory only for $N = 2$ [14], Goresky and Klapper gave an algorithm for the FCSR synthesis problem. This showed that the number of bits we need to know for finding the smallest FCSR that generates a given periodic sequence $\mathbf{a}$ is highly related to N-adic complexity. We discuss N-adic complexity and these algorithms in detail in later sections.

4. An l-sequence is a periodic sequence $\mathbf{a}$ which is obtained from a FCSR with connection integer q such that $q = p^r$ is a power of an odd prime and the period of $\mathbf{a}$ is $\phi(q)$ (ϕ is Euler's totient function). These sequences have been studied since the time of Gauss [20]. They have remarkable distribution and correlation properties which are parallel to those of m-sequences.

1.3.3 Algebraic Feedback Shift Registers and d-FCSRs

A sequence generator based on algebra over complete rings, called an algebraic feedback shift register, was proposed as a generalization of LFSR and FCSR [32]. Here

are some notions we will use.

Let R be an integral domain and π be an element in R. Let S be a complete set of representatives for the quotient ring $R/(\pi)$ (This means that the composition $S \to R \to R/(\pi)$ is a one to one correspondence). For any $u \in R$ denote its image in $R/(\pi)$ by $\widetilde{u} = u \pmod{\pi}$. Having chosen S, every element $a \in R$ has a unique expression $a = a_0 + b\pi$, where $a_0 \in S$. The element a_0 is the representative of $\widetilde{a}$ in S, and $a - a_0$ is divisible by π. We write

$$a_0 = a \pmod{\pi} \text{ and } b = a \,(\text{div } \pi) = \frac{a - a_0}{\pi}.$$

Definition 1.3.6. *Let $q_0, q_1, q_2, \cdots, q_m \in R$ and assume that q_0 is invertible $\pmod{\pi}$. An algebraic feedback shift register (or AFSR) over (R, π, S) of length m with multipliers or taps $q_1, q_2, \cdots, q_m$ is a sequence generator whose states are elements*

$$s = (a_0, a_1, \cdots, a_{m-1}; z) \in S^m \times R$$

*consisting of cell contents a_i and memory z. The output is **out**(s)=a_0. The state change operation is described as follows:*

1. *Compute*
$$\sigma = \sum_{i=1}^{m} q_i a_{m-i} + z.$$
2. *Find $a_m \in S$ such that $-q_0 a_m \equiv \sigma \pmod{\pi}$. That is, $\widetilde{a}_m = -\widetilde{q}_0^{-1}\widetilde{\sigma}$.*
3. *Replace $(a_0, a_1, \cdots, a_{m-1})$ by $(a_1, a_2, \cdots, a_m)$ and replace z by $\sigma \,(\text{div}\pi) = (\sigma + q_0 a_m)/\pi$.*

The procedure is illustrated in Figure 1.6.

An LFSR over a field K is an AFSR where $R = K[x]$ is the ring of all polynomials with coefficients in K, $\pi = x$ and $S = K$ is the set of polynomials of degree 0, which may also be identified with the quotient $R/(\pi) = K[x]/(x)$.

An FCSR is an AFSR with $R = \mathbb{Z}$, $\pi = N$, and $S = \{0, 1, \cdots, N-1\}$. There are many other special cases that have been introduced [22, 31].

For better understanding the analysis of AFSRs, we first recall the basics of algebra. Let R be a commutative ring which is an integral domain (no zero divisors). Let F be its field of fractions. Let $\pi \in R$ be a prime element. The principal ideal generated by π is denoted $I = (\pi)$. Any such π defines a topology on R with respect to which the operations of addition and multiplication are continuous. The set $\{(\pi^i)\}$

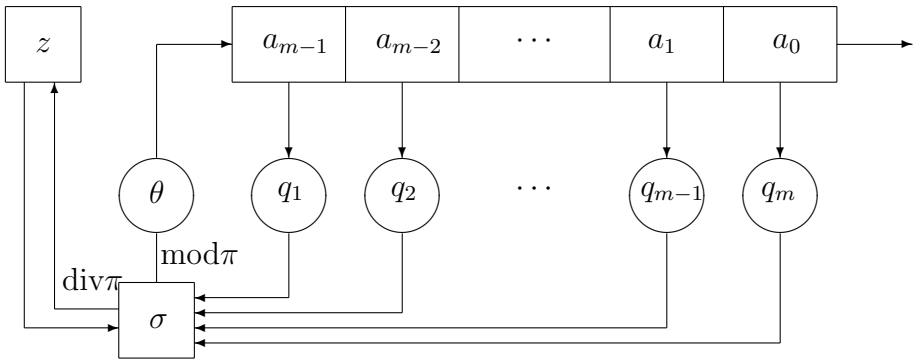

Figure 1.6: An algebraic feedback shift register of length m

forms a basic set of neighborhoods of zero. This topology is known as the $\pi-$adic topology on R and extends to F with the same basic set of neighborhoods of zero.

A *completion* of the π-adic topology on R is a topological ring $\widehat{R}$ containing R that is complete (every Cauchy sequence converges) and is a minimal completion containing R. The same notion of completion applies to F.

The set of power series

$$\sum_{i=0}^{\infty} a_i \pi^i, \quad a_i \in R \tag{1.4}$$

is a completion of R with the π-adic topology if $\cap_n (\pi)^n = (0)$ (we assume this always holds in the following pages). Two such power series $\sum_{i=0}^{\infty} a_i \pi^i$ and $\sum_{i=0}^{\infty} b_i \pi^i$ are identified if for every n,

$$\sum_{i=0}^{n-1} (a_i - b_i)\pi^i \in (\pi)^n$$

Addition and multiplication can be defined naturally. The resulting ring is called the completion of R or the set of π-*adic integers* and is denoted by R_π. If π is irreducible then the ring R_π has a unique prime ideal $\widehat{I}$, the set of such power series with $a_0 = 0$. We have $(\pi) = \widehat{I} \cap R$.

Let S be a complete set of representatives for R modulo π. It can be shown that every element of R_π can be written uniquely in the form of equation (1.4) with every a_i in S. This means that an element of R_π can be expressed as a sequence of elements of S.

Consider the AFSR over (R, π, S) with m multipliers $q_0, q_1, \cdots, q_m$ and the initial state $(a_0, a_1, \cdots, a_m; z)$. As with LFSRs and FCSRs, define the connection element as $q = q_0 + q_1\pi + \cdots + q_m\pi^m \in R_\pi$. The associated π-adic number integer can expressed

in a rational form shown in Theorem 1.3.1.

Theorem 1.3.1. *(Fundamental Theorem on AFSRs [32]) Let the output sequence* $\mathbf{a} = a_0, a_1, \ldots$ *of an AFSR with connection element* q *and initial state* $(a_0, a_1, \cdots, a_{m-1}; z)$ *have associated* π*-adic integer* $\alpha = \sum_{i=0}^{\infty} a_i \pi^i$*. Then*

$$\alpha = \frac{\sum_{n=0}^{m-1} \sum_{i=0}^{n} q_i a_{n-i} \pi^n - z\pi^m}{q} = \frac{u}{q} \in R_\pi. \tag{1.5}$$

The expression u/q *is called a* rational expression *of* α*.*

For more results about AFSRs, please refer to [25] (Chapter 5). Here we are more interested in a special case of AFSRs called d-FCSRs which was first introduced in [31] and described and analyzed in [22–24].

Definition 1.3.7. *Let* $N \geq 2$ *and* $d \geq 1$ *be integers such that the polynomial* $x^d - N$ *is irreducible over the rational number field* $\mathbb{Q}$ *and* $\pi \in \mathbb{C}$ *is a root of this polynomial in an extension field of* $\mathbb{Q}$*. A* d*-FCSR is an AFSR over* $(R = \mathbb{Z}[\pi], \pi, S)$*, where* $\mathbb{Z}[\pi]$ *is the set of polynomials in* π *modulo* $x^d - N$ *with integer coefficients and* $S = \{0, 1, 2, \cdots, N-1\}$*.*

A binary d-FCSR is a special case of d-FCSR with $N = 2$. For an intuitive understanding of the procedure d-FCSR, we look at such binary d-FCSRs in detail. Now, $\pi^d = 2$, $R = \mathbb{Z}[\pi]$, and $S = \{0, 1\}$. Any $z \in \mathbb{Z}[\pi]$ may be uniquely expressed as a polynomial $z = z_0 + z_1\pi + \cdots + z_{d-1}\pi^{d-1}$ with $z_i \in \mathbb{Z}$ by making use of the equation $\pi^d = 2\pi^0$ whenever higher powers of π are encountered. Using the binary expansion of each z_i, any element $z \in \mathbb{Z}[\pi]$ with all $z_i \geq 0$ can be uniquely expressed as a polynomial

$$z = \sum_{i=0}^{e} z_i' \pi^i$$

with coefficients $z_i' \in \{0, 1\}$ and $e \geq 0$. Addition and multiplication are performed as for integers, except that carried bits are advanced d steps because

$$1 + 1 = 2 = 0 + 0\pi + 0\pi^2 + \cdots + 0\pi^{d-1} + 1\pi^d.$$

The operations $(\text{mod } \pi)$ and $(\text{div } \pi)$ are defined as $z \ (\text{mod } \pi) = z_0 \ (\text{mod } 2) \in \mathbb{F}_2$ and $z \ (\text{div } \pi) = z_1\pi + \cdots + z_{d-1}\pi^{d-1}$.

The circuit of a binary d-FCSR with $d = 2$ is illustrated in Figure 1.7.

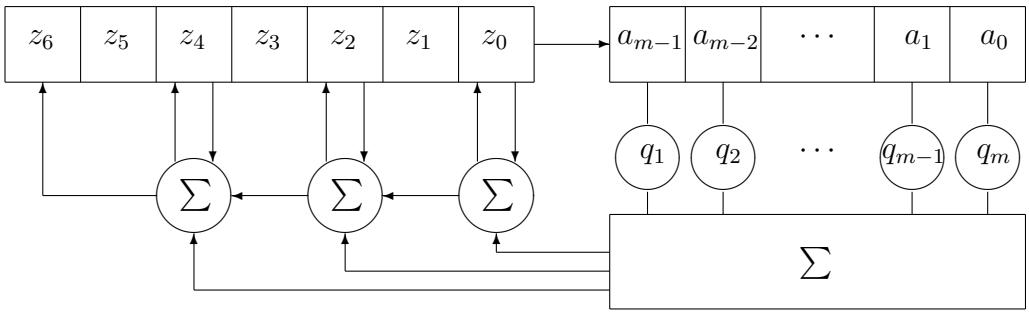

Figure 1.7: A binary d-FCSR with $d = 2$

1.4 Register Synthesis Problem

Given a short prefix of a sequence, it is undesirable to have an efficient algorithm that can synthesize a generator which can generate the whole sequence. Otherwise, a cryptanalytic attack can be launched against the system based on that given sequence. So finding such synthesis algorithm is an interesting problem in cryptanalysis. For a class of generators $\mathcal{F}$ and a sequence $\mathbf{a}$, a *register synthesis algorithm* is an algorithm that finds the smallest size generator in $\mathcal{F}$ that outputs sequence $\mathbf{a}$ given only a prefix of $\mathbf{a}$. We consider sequence generator classes $\mathcal{F}$ to be the set of LFSRs over a particular ring R, the set of FCSRs for a particular N, or the set of AFSRs over a particular integral domain.

For the class of LFSRs over a particular ring R, the size of an LFSR is measured by the number of cells used to represent the states. Given a sequence $\mathbf{a}$, the size of the smallest LFSR that can generate $\mathbf{a}$ is defined as the linear complexity (or linear span) of $\mathbf{a}$, denoted by $\lambda(\mathbf{a})$. The most famous LFSR synthesis algorithm, Berlekamp-Massey algorithm, can find the smallest LFSR that generates a given sequence $\mathbf{a}$ with only $2\lambda(\mathbf{a})$ consecutive bits of $\mathbf{a}$ [44]. The LFSR synthesis problem can also be solved by constructing a system of linear equations, by the extended Euclidean algorithm, or by the continued fraction algorithm [52, 53]. On the one hand, these algorithms provide a way to predict the whole sequence by using part of the information. On the other hand, they illustrate that linear complexity is an important security measure for pseudorandom sequences design. So in Chapter 2, we investigate lower bounds of the linear complexity of different kinds of pseudorandom sequences.

FCSRs share many good algebraic properties with LFSRs. Klapper and Goresky gave an lattice approximation approach to the FCSR synthesis problem [30] in terms of integer approximation lattice that was proposed by Mahler [43] and de Weger [14]. In the case of binary FCSRs, the algorithm can construct the smallest FCSR which generates the sequence $\mathbf{a}$, and it does so using only a knowledge of the first $2\lambda_2(\mathbf{a})+\lceil 2\log_2(\lambda_2(\mathbf{a}))\rceil+2$ bits of $\mathbf{a}$, where $\lambda_2(\mathbf{a})$ is the 2-adic complexity of $\mathbf{a}$ [25]. For arbitrary N, Arnault, Berger, and Necer proposed an algorithm based on the extended Euclidean algorithm [7]. The register synthesis problem for FCSRs was also solved by Xu's algorithm which is a modified version of Berlekamp-Massey algorithm [33].

As an extension of single sequences, multi-sequences have been introduced for applications of word-oriented stream ciphers. For positive integers M and N, an M-fold N-ary multi-sequence is denoted by

$$\mathcal{S} = \left(\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}\right),$$

which consists of M parallel streams of N-ary sequences $\mathbf{S}^{(h)} = \left(s_0^{(h)}, s_1^{(h)}, s_2^{(h)}, \ldots\right)$, where $s_i^{(h)} \in \{0, 1, ..., N-1\}$ for $i \in \mathbb{N}$ and $h = 0, 1, \ldots, M-1$. We say $\mathcal{S}$ is eventually periodic if $\mathbf{S}^{(h)}$, $h = 0, 1, \ldots, M-1$, are all eventually periodic sequences.

The register synthesis problem for single sequences can be extended to multi-sequences. That is, given a prefix of each sequence $\mathbf{S}^{(h)}$, find a common generator of the smallest size that can generate all M sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$ (with a different initial state for each sequence). The LFSR synthesis problem for multi-sequences has been solved by Feng and Tzeng's generalized Euclidean algorithm [16], by a modification of the fundamental iterative algorithm [60], by the Sakata algorithm using Gröbner basis theory [59] and by an F$[x]$-lattice basis reduction algorithm [70]. However, multi-sequence synthesis with FCSRs is more complicated than with LFSRs. To the best of our knowledge, there are no FCSR synthesis algorithms for multi-sequences in the literature. Thus the main contributions of Chapter 3 is to adapt an interleaving technique to develop two algorithms to solve the FCSR synthesis problem for M-fold N-ary multi-sequences under the restriction that $x^M - N$ is irreducible over the rational field $\mathbb{Q}$ for $M \geq 2$ and $N \geq 2$.

AFSRs are generalizations of LFSRs and FCSRs. An AFSR based on a chosen R, π and S can produce sequences whose elements can be thought of as elements of the quotient ring $R/(\pi)$. As a modification of the Berlekamp-Massey algorithm, Xu's algorithm can solve the synthesis problem for many AFSRs over the pair (R, π) that has certain algebraic properties [33]. In Chapter 4, we propose two register synthesis algorithms for AFSRs. The first one can be seen as an extension of lattice

approximation approach but based on lattice basis reduction. For AFSRs over (R, π), where $R = \mathbb{Z}[\pi]$ with $\pi^2 = D \in \mathbb{Z}$, the algorithm can find the smallest AFSR that generates the sequence $\mathbf{a}$ given at least $2\varphi_\pi(\mathbf{a}) + 2 + \lceil \log_{|D|}(4D^2 + 2|1 + D|) \rceil$ terms of sequence $\mathbf{a}$, where $\varphi_\pi(\mathbf{a})$ is the π-adic complexity of $\mathbf{a}$. It has quadratic time complexity. The second algorithm applies the extended Euclidean algorithm on a norm-Euclidean imaginary quadratic field to find a smallest AFSR for a given sequence $\mathbf{a}$. It is more efficient than the lattice rational approximation algorithm in that only $2\phi_\pi(\alpha) + 1$ terms of sequence $\mathbf{a}$ are needed. $\phi_\pi(\alpha)$ is a complexity measure that reflects the size of AFSRs.

1.5 Lattices and basis reduction

An integer *lattice* L of rank d is a discrete additive subgroup of $\mathbb{R}^n$ of the form

$$L(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_d) := \sum_{i=1}^{d} \mathbf{b}_i \mathbb{Z},$$

where $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d \in \mathbb{R}^n$ are linearly independent vectors over $\mathbb{R}$ [15]. We call $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d)$ a basis of lattice L. Usually, the basis of a lattice is not unique. When $d = n$, we call L a full lattice. We always suppose the lattice we discuss is full. For arbitrary vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d \in \mathbb{R}^d$, let

$$span(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_d) := \sum_{i=0}^{d} \mathbf{b}_i \mathbb{R}$$

be the space spanned by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$.

Let $\langle \cdot, \cdot \rangle$ be the inner product of $\mathbb{R}^d$. That is, for two vectors $\mathbf{u} = (u_1, u_2, \ldots, u_d) \in \mathbb{R}^d$ and $\mathbf{v} = (v_1, v_2, \ldots, v_d) \in \mathbb{R}^d$,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{d} u_i v_i.$$

Let $|| \cdot ||$ and $|| \cdot ||_\infty$ be the Euclidean norm and the sup (or L_∞) norm on lattices respectively. So for any vector $\mathbf{u} = (u_1, u_2, \ldots, u_d) \in \mathbb{R}^d$, we have

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle} = \sqrt{u_1^2 + u_2^2 + \cdots + u_d^2},$$

and

$$||\mathbf{u}||_\infty = \max(|u_0|, |u_1|, \ldots, |u_d|).$$

The notation $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]_\leq$ means $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \leq \cdots \leq \|\mathbf{b}_d\|$ which is to say the $\mathbf{b}_i$s are ordered. The Gram matrix, denoted by $G(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d)$, is a $d \times d$ symmetric matrix with entries given by $G_{ij} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle$.

Definition 1.5.1. *[15] The determinant,* $\det(L)$*, of lattice* $L(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_d)$ *is defined by*

$$\det(L) = \det(G(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d))^{1/2}.$$

When L *is a full lattice, we have* $\det(L)$ *is the determinant of the matrix whose rows are the* $\mathbf{b}_i$*.*

Proposition 1.5.1. *[15] The determinant of a lattice is independent of the choice of basis* $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d \in \mathbb{R}^d$*.*

Definition 1.5.2. *[15] (***Successive Minima** $\zeta_1, \zeta_2, \ldots, \zeta_d$*) For every lattice* $L \in \mathbb{R}^d$ *of rank* d *the successive minima* $\zeta_1, \zeta_2, \ldots, \zeta_d$ *are defined as:*

$$\zeta_i = \zeta_i(L) := \min \left\{ r > 0 \,\middle|\, \begin{array}{l} \exists \text{ linearly independent} \\ c_1, c_2, \ldots, c_i \in L \text{ with} \\ \|c_j\| \leq r \text{ for } j = 1, 2 \ldots, i \end{array} \right\}, \text{ for } i = 1, 2, \ldots, d.$$

The successive minima depend on the underlying norm. The first successive minimum with respect to the Euclidean norm is

$$\zeta_1(L) = \min(||\mathbf{b}|| : \mathbf{b} \in L \backslash \{0\}).$$

The first successive minimum with respect to the sup norm is

$$\zeta_{1,\infty}(L) = \min(||\mathbf{b}||_\infty : \mathbf{b} \in L \backslash \{0\}).$$

The definition of successive minima is due to H. Minkowski [48]. The values of successive minima remain unchanged under isometric transformations of the lattice [15], so they are geometric lattice invariants. According to Proposition 1.5.1, the determinant of a lattice is also a geometric lattice invariant. However, the value $\zeta_{1,\infty}(L)$ is not a geometric invariant but with a bound [48],

$$\zeta_{1,\infty}(L) \leq (\det(L))^{1/d}.$$

Suppose we have two vectors $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$, where $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_d) \in \mathbb{R}^d$ has the smallest Euclidean norm in L and $\hat{\mathbf{u}} = (\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_d) \in \mathbb{R}^d$ has the smallest sup norm. That is, $||\tilde{\mathbf{u}}|| = \zeta_1(L)$ and $||\hat{\mathbf{u}}||_\infty = \zeta_{1,\infty}(L)$. Normally, $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$ are not the same vectors, but they are related by the inequalities

$$\begin{aligned} ||\tilde{\mathbf{u}}||_\infty &= \max(|\tilde{u}_1|, \ldots, |\tilde{u}_d|) \\ &\leq \sqrt{|\tilde{u}_1|^2 + \cdots + |\tilde{u}_d|^2} = ||\tilde{\mathbf{u}}|| \\ &\leq ||\hat{\mathbf{u}}|| = \sqrt{|\hat{u}_1|^2 + |\hat{u}_2|^2 + \cdots + |\hat{u}_d|^2} \\ &\leq \sqrt{d \cdot \max(|\hat{u}_1|^2, |\hat{u}_2|^2, \ldots, |\hat{u}_d|^2)} \\ &\leq \sqrt{d}||\hat{\mathbf{u}}||_\infty \end{aligned} \tag{1.6}$$

and

$$\begin{aligned} ||\hat{\mathbf{u}}|| &= \sqrt{|\hat{u}_1|^2 + |\hat{u}_2|^2 + \cdots + |\hat{u}_d|^2} \\ &\leq \sqrt{d \cdot \max(|\hat{u}_1|^2, |\hat{u}_2|^2, \ldots, |\hat{u}_d|^2)} \\ &\leq \sqrt{d}||\tilde{\mathbf{u}}||_\infty \leq \sqrt{d}||\tilde{\mathbf{u}}||. \end{aligned} \tag{1.7}$$

Given a basis of a lattice L, finding a vector of the smallest norm (the shortest vector problem or SVP) is a computationally hard problem in lattice theory. Although the SVP has been proved to be NP-hard if the dimension is unrestricted [2], there are some efficient algorithms based on lattice basis reduction that can solve the SVP under certain conditions. Loosely speaking, the lattice reduction problem is: given an arbitrary lattice basis, obtain a basis of shortest possible vectors which are mutually orthogonal. Finding a good reduced basis has many important applications in mathematics, computer science, and cryptography. They were used to break Merkle-Hellman public key cryptosystem based on the knapsack problem [46] or based on rational numbers [66], Blum's protocol for exchanging secrets [19], truncated linear congruential generators [19], RSA with exponent 3 [10,28], and NTRU (a lattice-based cryptosystem proposed by Hoffstein, Pisher, and Silverman) [11]. Nguyen and Stern surveyed the applications of lattices to cryptology and explained the developments of lattice reduction both in cryptography and cryptanalysis [51]. There are many different kinds of lattice reduction, such as Hermite [27], Minkowski [48], Venkov [58], Hermite-Korkine-Zolotarev(HKZ) [35], and Lenstra-Lenstra-Lovász (LLL) [37]. For two dimensional lattices, Gauss's basis reduction algorithm, which is a generalization of the Euclidean algorithm, can be used. For higher dimensions, the lattice reduction problem is more complicated..

In this work we utilize two algorithms, the lattice reduction greedy algorithm [50] and the LLL Algorithm [37]. The lattice reduction greedy algorithm is a generalization of the Lagrange's algorithm on arbitrary dimensions. Up to dimension four, it can compute a Minkowski reduced basis, which includes the shortest vector as its first vector, in quadratic time. But it becomes extremely complicated as the dimension increases and may not output a basis that is Minkowski reduced. The LLL algorithm can find a LLL reduced basis in polynomial time. It is an approximation algorithm for the Shortest Vector Problem.

1.5.1 The LLL lattice basis reduction

Definition 1.5.3. *Given d linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d \in \mathbb{R}^d$, the Gram-Schmidt orthogonalization of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$ is $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_d^*$. The $\mathbf{b}_i^*$s, $i =$*

$1, 2, \ldots, d$ *are defined by*

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*,$$

where $\mu_{i,j} = \frac{<\mathbf{b}_i, \mathbf{b}_j^*>}{<\mathbf{b}_j^*, \mathbf{b}_j^*>}$.

Definition 1.5.4. *A basis* $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]_{\leq}$ *is called LLL reduced with parameter* δ *(or a* δ*-LLL reduced basis),* $1/4 < \delta \leq 1$*, when:*

1. $|\mu_{i,j}| \leq 1/2$*, for* $1 \leq j < i \leq d$*;*

2. $\delta \cdot ||\mathbf{b}_{i-1}^*||^2 \leq ||\mathbf{b}_i^*||^2 + \mu_{i-1,i}^2||\mathbf{b}_{i-1}^*||^2$ *for* $i = 2, 3, \ldots, d$.

The first property is a criterion for being length reduced. The parameter δ describes how well reduced the basis is. A larger value for δ implies a more strongly reduced basis [15]. The LLL reduction was originally defined by A.K. Lenstra, H.W. Lenstra and L. Lovász with $\delta = 3/4$ [37]. The LLL algorithm, given in Figure 1.8, takes an arbitrary basis of L as inputs and outputs a δ-LLL reduced basis with polynomial time complexity. Given a d-dimensional integer lattice basis with vectors of Euclidean norm less than B in a d-dimensional space, the LLL algorithm outputs a δ-LLL reduced basis in $O(d^4 \log B \cdot \mathcal{M}(d \log B))$ bit operations, where $\mathcal{M}(d \log B)$ denote the time required to multiply $d \log B$-bit integers [49]. The first vector of the output of the LLL algorithm, say $\mathbf{b}_1$, has the property that $||\mathbf{b}_1|| \leq (\frac{1}{\delta - 1/4})^{(d-1)/2}\zeta_1(L)$. This means that $\mathbf{b}_1$ can be used to approximate the smallest nonzero vector in the lattice.

1.5.2 Low-dimensional lattice basis reduction

Definition 1.5.5. *[50] (***Minkowski reduction***) A basis* $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]_{\leq}$ *of a lattice* L *is Minkowski-reduced if for all* $1 \leq i \leq d$*,* $\mathbf{b}_i$ *has minimal norm among all lattice vectors* $\mathbf{b}_i$ *such that* $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_i]_{\leq}$ *can be extended to a basis of* L.

Notice that the first vector in a Minkowski-reduced basis is the shortest nonzero vector in lattice L. It has been proved that the shortest vector problem (SVP) is NP-hard if the dimension is unrestricted [2]. Nguyen and Stehlé [50] proposed a greedy algorithm that generalizes Lagrange's algorithm for lattice reduction to arbitrary dimension. They showed that up to dimension four, their algorithm computes a Minkowski-reduced basis in quadratic time without fast arithmetic but as the dimension increases, the analysis becomes more complex. Figure 1.9 is an iterative description of Nguyen and Stehl's greedy algorithm [50].

1: **procedure** LLL($\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$)
2: **Input:** A basis $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d$ of lattice L.
3: **Output:** A δ-LLL reduced basis of $L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d)$.
4: /* Compute the Gram-Schmidt orthonormalization $\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_d^*$.*/
5: **for** $i = 1$ to d **do**
6: $\mathbf{b}_i^* := \mathbf{b}_i$
7: **for** $j = 1$ to i-1 **do**
8: $\mu_{i,j} := \frac{<\mathbf{b}_i, \mathbf{b}_j^*>}{<\mathbf{b}_j^*, \mathbf{b}_j^*>}$
9: $\mathbf{b}_i^* := \mathbf{b}_i - \mu_{i,j}\mathbf{b}_j^*$
10: **end for**
11: **end for**
12: /* Reduction step */
13: $k := 2$ /* k is the stage */
14: **while** $k \leq d$ **do**
15: **for** $j := k - 1$ to M **do**
16: $\mathbf{b}_k := \mathbf{b}_k - \lfloor \mu_{k,j} \rceil \mathbf{b}_j$ where $\lfloor \cdot \rceil$ is rounding to the nearest integer.
17: Update the Gram-Schmidt orthogonalization accordingly.
18: **end for**
19: **if** $\delta \cdot ||\mathbf{b}_{k-1}^*||^2 > ||\mathbf{b}_k^*||^2 + \mu_{k,k-1}^2 ||\mathbf{b}_{k-1}^*||^2$ **then**
20: $\mathbf{b}_{k-1} \leftrightarrow \mathbf{b}_k$ i.e. exchange $\mathbf{b}_{k-1}$ and $\mathbf{b}_k$
21: Update the Gram-Schmidt orthogonalization accordingly.
22: $k := \max(k - 1, 2)$
23: **else**
24: $k := k + 1$
25: **end if**
26: **end while**
27: **end procedure**

Figure 1.8: The LLL Algorithm

Theorem 1.5.1. *[50] Let $d \leq 4$. Given as input an ordered basis $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]_{\leq}$ and its Gram matrix, the greedy algorithm of Figure 1.9 outputs a Minkowski-reduced basis of $L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d)$, with bit complexity in $O(\log \|\mathbf{b}_d\| \cdot [1 + \log \|\mathbf{b}_d\| - \log \zeta_1(L)])$, where the $O()$ constant is independent of the lattice. Moreover, in dimension five, the output basis may not be Minkowski-reduced.*

We use the greedy algorithm in four dimensions, i.e., $d = 4$, to find the shortest vector in L in our Rational Approximation algorithm. More exactly, the closest vector in step 6 of GreedyLatticeReduction can be found as follows.

1. Let $\mathbf{h} = \sum_{i=1}^{m-1} y_i \mathbf{b}_i$ be the orthogonal projection of $\mathbf{b}_m$ on $span(\mathbf{b}_1, \mathbf{b}_2, \ldots,$

procedure GreedyLatticeReduction($b_1, b_2, \ldots, b_d$)
Input: A basis $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d]_{\leq}$ with its Gram matrix
Output: An ordered basis of $L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d)$ with its Gram matrix
    $m := 2$
    while $m \leq d$ **do**
        Compute a vector $\mathbf{c} \in L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1})$ closest to $\mathbf{b}_m$
    end while
    $\mathbf{b}_m := \mathbf{b}_m - \mathbf{c}$ and update the Gram matrix
    if $\|\mathbf{b}_m\| \geq \|\mathbf{b}_{m-1}\|$ **then**
        $m := m + 1$
    else
        insert $\mathbf{b}_m$ between $\mathbf{b}_{m'-1}$ and $\mathbf{b}_{m'}$ such that $\|\mathbf{b}_{m'-1}\| \leq \|\mathbf{b}_m\| < \|\mathbf{b}'_m\|$.
        update the Gram matrix and set $m := m' + 1$.
    end if
end procedure

Figure 1.9: The lattice reduction greedy algorithm

$\mathbf{b}_{m-1}$). Then

$$G(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1}) \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{m-1} \end{pmatrix} = \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_m \rangle \\ \langle \mathbf{b}_2, \mathbf{b}_m \rangle \\ \vdots \\ \langle \mathbf{b}_{k-1}, \mathbf{b}_m \rangle \end{pmatrix}.$$

2. Let $\mathbf{c}$ be the closest vector to $\mathbf{h}$ in $L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1})$. Then $\mathbf{h}-\mathbf{c} \in \mathrm{Vor}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1})$, where $\mathrm{Vor}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1}) = \{\mathbf{x} \mid \|\mathbf{x}-\mathbf{v}\| \geq \|\mathbf{x}\|, \ \forall\, \mathbf{v} \in L(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{m-1})\}$ is the Voronoï cell. With Theorem 1.5.2, $\mathbf{c}$ can be found by a suitable exhaustive search when $j \leq 4$.

Theorem 1.5.2. *[50]*

1. *Let* $[\mathbf{b}_1, \mathbf{b}_2]_{\leq}$ *be a Minkowski-reduced basis and* $\mathbf{u} \in \mathrm{Vor}(\mathbf{b}_1, \mathbf{b}_2)$. *Write* $\mathbf{u} = x\mathbf{b}_1 + y\mathbf{b}_2$. *Then* $|x| < 3/4$ *and* $|y| \leq 2/3$.

2. *Let* $[\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]_{\leq}$ *be a Minkowski-reduced basis and* $\mathbf{u} \in \mathrm{Vor}(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$. *Write* $\mathbf{u} = x\mathbf{b}_1 + y\mathbf{b}_2 + z\mathbf{b}_3$. *Then* $|x| < 3/4$, $|y| \leq 2/3$ *and* $|z| \leq 1$.

2 LFSR synthesis and linear complexity

2.1 Previous works on LFSR synthesis algorithms

Linear complexity is an important security measure of pseudorandom sequences. We consider the register synthesis problem for LFSRs over a field $\mathbb{F}$. Recall the properties of LFSRs sequence in Section 1.3.1. Let

$$a(x) = \sum_{i=0}^{\infty} a_i x^i$$

be the generating function associated with sequence $\mathbf{a}$. Suppose eventually periodic sequence $\mathbf{a}$ can be generated by an LFSR with connection polynomial $q(x)$. Then there is a polynomial $f(x)$ in $\mathbb{F}[x]$ so that $a(x) = f(x)/q(x)$. Equivalently,

$$q(x)a(x) = f(x). \tag{2.1}$$

The size of the LFSR [25] is defined as

$$\Phi(f, g) = \max(\deg(f(x)) + 1, \deg(q(x))).$$

Thus the linear complexity $\lambda(\mathbf{a})$ is the minimum over all f, g with $a(x) = f(x)/g(x)$ of $\Phi(f, g)$. In other words, $\lambda(\mathbf{a})$ is the size of the smallest LFSR that can generate the sequence $\mathbf{a}$. Notice that if $\mathbb{F}$ is a finite field, then $\deg(f(x)) < \deg(q(x))$ because sequence $\mathbf{a}$ is strictly periodic. In this case, $\lambda(\mathbf{a}) = \deg(q(x))$, where $q(x)$ is the connection polynomial of the smallest LFSR that generates sequence $\mathbf{a}$.

The LFSR synthesis problem can be rephrased as follows:

- **Given** a prefix $a_0, a_1, \cdots, a_{k-1}$ of $\mathbf{a}$.
- **Find** a pair (f, g) that minimizes $\Phi(f, g)$ among all polynomials f, g that satisfy equation (2.1).

The most famous synthesis algorithm for LFSRs is Berlekamp-Massey algorithm which is given in Figure 2.1.

We say a pair $(f(x), q(x))$ of polynomials form a *degree i approximation* to $a(x)$, if

$$q(x)a(x) \equiv f(x) \pmod{x^i}.$$

A natural number i is a *turning point* if

$$\Phi(f_{i+1}, q_{i+1}) > \Phi(f_i, q_i)$$

1: **procedure** B-M($a_0, \cdots, a_{n-1}$)
2: **if** all $a_i = 0$ **then**
3: **return** (0,1)
4: **else**
5: $a(x) = \sum_{i=0}^{n-1} a_i x^i$
6: Let m be minimal with $a_m \neq 0$
7: $f_m(x) = 0$
8: $q_m(x) = 1$
9: $f_{m+1}(x) = a_m x^m$
10: $q_{m+1}(x) = \begin{cases} 1 + x^m & \text{if } m > 0 \\ 1 & \text{else} \end{cases}$
11: $c = a_m$
12: **for** $i = m + 1$ to $n - 1$ **do**
13: Let $a(x)q_i(x) - f_i(x) \equiv bx^i \pmod{x}^{i+1}$
14: **if** $b = 0$ **then**
15: $f_{i+1}(x) = f_i(x)$
16: $q_{i+1}(x) = q_i(x)$
17: **else**
18: $f_{i+1}(x) = f_i(x) - (b/c)x^{i-m} f_m(x)$
19: $q_{i+1}(x) = q_i(x) - (b/c)x^{i-m} q_m(x)$
20: **if** $\Phi(f_{i+1}, q_{i+1}) > \Phi(f_i, q_i)$ **then**
21: $m = i$
22: $c = b$
23: **end if**
24: **end if**
25: $i = i + 1$
26: **end for**
27: **return** (f_n, q_n)
28: **end if**
29: **end procedure**

Figure 2.1: The Berlekamp-Massey algorithm

Theorem 2.1.1. *[25] Let* $\mathbf{a} = a_0, a_1, \cdots$ *and let* $a(x) \in \mathbb{F}[[x]]$ *be its generating function. Let* (f_i, q_i) *be the values computed at stage* $i \geq 1$ *in the Berlekamp-Massey algorithm. Then* (f_i, q_i) *is a degree* i *approximation to* $a(x)$*. Suppose* (f, q) *is another degree* i *approximation to* $a(x)$*. Then*

$$\Phi(f_i, q_i) \leq \Phi(f, q)$$

If $\Phi(f_i, q_i) = \Phi(f, q)$ *and if* i *is a turning point, then* $f_i/q_i = f/q$*. If* $i \geq 2\lambda(a)$*, then* $\Phi(f_i, q_i) = \lambda(a)$ *and* $f_i(x)/q_i(x) = a(x)$*.*

Theorem 2.1.1 says that at each stage the Berlekamp-Massey algorithm generates a Φ-minimizing approximation. If i is a turning point then there is a unique such approximation. If $i \geq 2\lambda(\mathbf{a})$ then this approximation is exact: it generates the whole sequence $\mathbf{a}$. The overall time complexity of the Berlekamp-Massey algorithm is $O(n^2)$ where n is the number of known symbols of the sequence. Furthermore, the algorithm is adaptive: each time a new bit is determined, it can be used to update the determined LFSR in linear worst case time [25].

2.2 Linear complexity of FCSR sequences

Linear complexity has been extensively studied [52, 56]. The linear complexity test has been selected by NIST (National Institute of Standards and Technology) as one of the randomness tests in the statistical test suite for random and pseudorandom number generators for cryptographic applications [57]. One of the important tools to study linear complexity is the characteristic polynomial.

Let $\mathbb{F}$ denote a field and $\mathbb{F}_q$ denote the finite field or Galois field with q elements. It is known that q must be a prime or a power of a prime. Suppose $q = p^e$ where p is a prime and $e \geq 1$. Then p is the characteristic of the field $\mathbb{F}_q$. Let $\mathbb{Z}/(n)$ be the quotient ring of integers modulo n. When q is not a prime, $\mathbb{Z}/(q)$ and $\mathbb{F}_q$ are different. We denote by $\mathbb{Z}/(n)^\times$ the multiplicative group of nonzero elements of $\mathbb{Z}/(n)$, which consists of all the invertible elements in $\mathbb{Z}/(n)$. It has been proved that $\mathbb{F}_q^\times$, the multiplicative group of $\mathbb{F}_q$, is a cyclic group (a group that can be generated by one element). A generator of $\mathbb{F}_q^\times$ is called a primitive element of $\mathbb{F}_q$.

Definition 2.2.1. *[25] Let* $\mathbf{a} = a_0, a_1, a_2, \cdots$ *be an arbitrary sequence of elements of the field* $\mathbb{F}$*. We say* $\mathbf{a}$ *satisfies a linear recurrence of order* m *if there exist coefficients* $q_0, q_1, \ldots, q_m \in \mathbb{F}$ *with* $q_0 \neq 0$ *such that*

$$q_0 a_i + q_1 a_{i-1} + \cdots + q_m a_{i-m} = 0 \text{ for } i = m, m+1, m+2, \cdots .$$

The polynomial

$$q^*(x) = q_0x^m + q_1x^{m-1} + \cdots + q_{m-1}x + q_m \in \mathbb{F}[x]$$

is called a characteristic polynomial of sequence $\mathbf{a}$. *The polynomial* $q^*(x)$ *is also called the reciprocal polynomial of* $q(x) = q_0 + q_1x + \cdots + q_mx^m \in \mathbb{F}[x]$.

Suppose that an LFSR with coefficients $q_1, q_2, \cdots, q_m$ generates the eventually periodic sequence $\mathbf{a}$. That is, the connection polynomial of the LFSR is $q(x) = -1 + \sum_{i=1}^{m} q_ix^i$. Then the reciprocal polynomial of $q(x)$, $q^*(x) = -x^m + q_1x^{m-1} + \cdots + q_{m-1}x + q_m$, is a characteristic polynomial of a.

Theorem 2.2.1. *[39] Let* $\mathbf{a} = a_0, a_1, a_2, \cdots$ *satisfies the linear recurrence. Then there exists a unique monic polynomial* $m(x) \in \mathbb{F}_q[x]$ *having the following properties:*

- $m(x)$ *is a characteristic polynomial of* $\mathbf{a}$*;*
- *a monic polynomial* $f(x) \in \mathbb{F}_q[x]$ *of positive degree is a characteristic polynomial of* $\mathbf{a}$ *if and only if* $m(x)$ *divides* $f(x)$.

The polynomial $m(x)$ *is called the minimal polynomial of the sequence.*

In fact, $m(x)$ is the characteristic polynomial of sequence $\mathbf{a}$ that has the least possible degree. It can be shown that $\lambda(\mathbf{a}) = \deg(m(x))$.

Definition 2.2.2. *[39] Let* $f(x) \in \mathbb{F}_q[x]$ *be a nonzero polynomial. If* $f(0) \neq 0$, *then the least positive integer* e *for which* $f(x)$ *divides* $x^e - 1$ *is called the order of* $f(x)$ *and is denoted by* $ord(f(x))$.

Theorem 2.2.2. *[39] Suppose* $\mathbf{a} = a_0, a_1, \cdots$ *satisfies the linear recurrence with minimal polynomial* $m(x) \in \mathbb{F}_q[x]$. *Then the least period of the sequence is equal to* $ord(m(x))$.

There is more discussion about characteristic polynomials and minimal polynomials [39, 52, 53].

In 1999, Seo, etc. proved a lower bound on the linear complexity of binary FCSRs with special connection integers using cyclotomic polynomials [61]. Qi and Xu extended the results to binary l-sequences (defined in Section 1.3.2) [54]. In the following parts of this section, we discuss the lower bounds defined for the linear complexities of FCSRs with more general settings.

2.2.1 Cyclotomic Polynomials

Before introducing the main results about the linear complexity, we summarize some known results related to cyclotomic polynomials that are important tools needed later. These results can be found in most books about number theory and finite fields [39, 40].

For every integer $n \geq 1$, Euler's function, $\phi(n)$, is defined to be the number of integers a such that $\gcd(a, n) = 1$, where $0 \leq a < n$. It satisfies the following:

1. For any prime p and positive integer k, $\phi(p^k) = p^{k-1}(p-1)$.

2. If $m, n \geq 1$ and $\gcd(m, n) = 1$, then $\phi(mn) = \phi(m)\phi(n)$.

For any integers n and $a > 0$ with $\gcd(a, n) = 1$, the order of a modulo n, denoted by $ord_n a$, is defined to be the least positive integer d such that $a^d \equiv 1 (\text{mod } n)$.

Theorem 2.2.3. *If $ord_n a = d$ and $a^m \equiv 1 \,(\text{mod } n)$ for some $m > 0$, then $d|m$.*

Theorem 2.2.4. *For any $n > 1$ and $\gcd(a, n) = 1$, we have*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

From the two theorems above, we can conclude that $ord_n a | \phi(n)$ for any $n > 1$ if $\gcd(a, n) = 1$. If $ord_n a = \phi(n)$, then a is called a primitive root modulo n. Here are two theorems that are used a lot for computing the order of integers.

Theorem 2.2.5. *Let the prime factorization of the integer n be $p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$. If $ord_{p_i^{e_i}} a = d_i$ for $i = 1, 2, \cdots, k$, then $ord_n a = \text{lcm}(d_1, d_2, \cdots, d_k)$.*

Theorem 2.2.6. *Let p be a prime. Then $ord_{p^{j+1}} a = ord_{p^j} a$ or $ord_{p^{j+1}} a = p \cdot ord_{p^j} a$ for $j \geq 2$.*

Definition 2.2.3. *[39] A polynomial $f(x) \in \mathbb{F}_q[x]$ is irreducible over $\mathbb{F}_q$ if $f(x)$ has positive degree and f(x)=g(x)h(x) with $g(x), h(x) \in \mathbb{F}_q[x]$ implies that either $g(x)$ or $h(x)$ is a constant polynomial.*

If K is a subfield of $\mathbb{F}_q$, then the polynomial $x^q - x$ in $K[x]$ factors in $\mathbb{F}_q[x]$ and $\mathbb{F}_q$ is a splitting field of $x^q - x$ over K. Let n be a positive integer. The splitting field of $x^n - 1$ over a field K is called the nth cyclotomic field over K and the roots of $x^n - 1$ in it are called the nth roots of unity over K. If K has characteristic p and n is not divisible by p, then all the nth roots of unity over K form a cyclic group. A generator of the cyclic group is called a primitive nth root of unity over K. In fact, let ξ be a primitive nth root of unity over K. There are exactly $\phi(n)$ different primitive nth roots of unity over K given by ξ^s where $1 \leq s < n$ and gcd(s, n) = 1.

Definition 2.2.4. *[40] Let K be a field of characteristic p. Let n be a positive integer not divisible by p, and let ξ be a primitive nth root of unity over K, then the polynomial*

$$\Phi_n(x) = \prod_{s \in (\mathbb{Z}/(n))^\times}^{n} (x - \xi^s)$$

is called the nth cyclotomic polynomial over K.

Proposition 2.2.1. *[40] Let K be a field of characteristic p with q elements. Let n a positive integer not divisible by p. Then*

1. $$x^n - 1 = \prod_{d|n} \Phi_d(x),$$

2. $$\Phi_n(x) = \prod_{i=1}^{\phi(n)/d} t_i(x),$$

where $t_i(x)$ is an irreducible polynomial of degree d, and $d = ord_n q$.

2.2.2 Complementary properties and special cases

Theorem 2.2.7. *[31] An N-ary FCSR sequence with connection integer q is eventually periodic with period dividing the order of N modulo q. If q is the least connection integer, then the period equals the order of N modulo q.*

Lemma 2.2.1. *Let $q = p^e$ where p is an odd prime and $e \geq 1$. If $\gamma \in \mathbb{Z}/(q)^\times$ has even order T modulo q, then* $\gcd(\gamma^{T/2} - 1, q) = 1$.

Proof: We have $\phi(q) = p^{e-1}(p-1)$, so $T | p^{e-1}(p-1)$. Suppose $T = p^s t$, where $0 \leq s \leq e-1$ and $t | p-1$. We claim that $ord_p \gamma = t$.

If the order of γ modulo p is w, then $w | p-1$. So $w \not| p$. We have

$$\gamma^T \equiv 1 \pmod{p^e}, \text{ so } \gamma^T \equiv 1 \pmod{p}.$$

According to Theorem 2.2.3, $w | T$. This implies that $w | t$. We have

$$\gamma^w \equiv 1 \pmod{p}, \quad so \quad \gamma^w = kp + 1 \quad \text{for some } k\ .$$

We have

$$\gamma^{wp^e} = (kp+1)^{p^e} \equiv 1 \pmod{p^e}.$$

So $T|wp^e$ which means that $t|w$. Thus, $t = w$. We have

$$\gamma^{T/2} = \gamma^{p^s t/2} = \gamma^{t/2} \not\equiv 1 \pmod{p}.$$

$\gcd(\gamma^{T/2} - 1, p) = 1$, so $\gcd(\gamma^{T/2} - 1, q) = 1$ □

Lemma 2.2.2. *Let $q = p_1 p_2 \cdots p_k$, where p_i's $(1 \leq i \leq k)$ are distinct odd primes. Let $\gamma \in \mathbb{Z}/(q)^\times$ and let the order of γ modulo p_i be T_i for $1 \leq i \leq k$. Suppose $\delta = \gcd(T_1, T_2, \ldots, T_k)$ is an even number and $T_i' = T_i/\delta$ is odd. Then $\gcd(\gamma^{T/2} - 1, q) = 1$, where $T = ord_q \gamma$.*

Proof:

$$T = \operatorname{lcm}(T_1, T_2, \ldots, T_k) = T_1' T_2' \ldots T_k' \delta.$$

For every i, we have

$$\gamma^{T/2} = \gamma^{T_i \prod_{j \neq i} T_j'/2} \not\equiv 1 \pmod{p_i},$$

so $p_i \not| (\gamma^{T/2} - 1)$. Further, $\gcd(\gamma^{T/2} - 1, q) = 1$. □

Lemma 2.2.3. *Let $q = q_1 q_2 \ldots q_k$ where $q_i = p_i^{e_i}$ such that p_i's are distinct odd primes and $e_i \geq 1$ $(1 \leq i \leq k)$. Let $\gamma \in \mathbb{Z}/(q)^\times$ and $ord_{q_i} \gamma = T_i$ for $1 \leq i \leq k$. Suppose $\delta = \gcd(T_1, T_2, \ldots, T_k)$ is an even number and $T_i' = T_i/\delta$ is odd. Then $\gcd(\gamma^{T/2} - 1, q) = 1$, where T is the order of γ modulo q.*

Proof: As shown in Lemma 2.2.1, $\gcd(\gamma^{T_i/2} - 1, q) = 1$ and the order of γ modulo p_i is t_i where $T_i = p_i^{s_i} t_i$ with $s_i \leq e_i$ and $t_i | p - 1$.

For $1 \leq i \leq k$, we have

$$\gamma^{T/2} = \gamma^{T_i \prod_{j \neq i} T_j'/2} = \gamma^{t_i p_i^{s_i} \prod_{j \neq i} T_j'/2} \not\equiv 1 \pmod{p_i}.$$

So $\gcd(\gamma^{T/2} - 1, q) = 1$.

□

Theorem 2.2.8. *Let q be the connection integer of an N-ary FCSR. Suppose $\gamma \equiv N^{-1} \pmod{q}$. Then $\gamma \in \mathbb{Z}/(q)^\times$. Let q and γ have the properties in Lemma 2.2.1, Lemma 2.2.2 or Lemma 2.2.3. If it is the smallest FCSR to generate the strictly periodic sequence $\mathbf{a} = (a_0, a_1, a_2 \ldots)$, then*

$$a_i + a_{i+T/2} = N - 1 \equiv -1 \pmod{N},$$

where T is the period of sequence $\mathbf{a}$.

Proof: Suppose **a** is associated with N-adic number a with the rational expression $a = -h/q$. Then

$$a_i \equiv \gamma^{-i} h \pmod{q} \pmod{N}.$$

Let $f_i \equiv \gamma^i h (\text{mod } q)$. So

$$\begin{aligned} f_i + f_{i+T/2} &\equiv \gamma^i h + \gamma^{i+T/2} h \\ &\equiv \gamma^i h (1 + \gamma^{T/2}) \\ &\equiv \gamma^i h \frac{1 - \gamma^T}{1 - \gamma^{T/2}} \pmod{q}. \end{aligned}$$

According to Theorem 2.2.7, $T = ord_q N$. It is also true that $ord_q \gamma = T$ because $\gamma \equiv N^{-1} (\text{mod } q)$. According to Lemma 2.2.1, Lemma 2.2.2 and Lemma 2.2.3, $\gcd(\gamma^{T/2} - 1, q) = 1$. So $f_i + f_{i+T/2} \equiv 0 (\text{mod } q)$.

For every $i \geq 0$, $0 < f_i \leq q - 1$ and $0 < f_{i+T/2} \leq q - 1$. So $f_i + f_{i+T/2} = q$

$$\begin{aligned} a_i + a_{i+T/2} &\equiv f_i + f_{i+T/2} \pmod{N} \\ &\equiv q \pmod{N} \\ &\equiv q_0 \pmod{N} \\ &\equiv -1 \pmod{N} \end{aligned}$$

□

Theorem 2.2.8 means that the FCSR sequence $\mathbf{a} = (a_0, a_1, a_2 \dots)$ described above satisfies a linear recurrence of order $T/2 + 1$, because

$$a_i - a_{i+1} + a_{i+T/2} - a_{i+T/2+1} = 0 \quad \text{holds for all} \quad i \geq 0.$$

A characteristic polynomial of sequence **a** is

$$f(x) = -1 + x - x^{T/2} + x^{T/2+1}$$

So the linear complexity of **a** is less than or equal to $T/2 + 1$.

Case 1: 2-adic FCSRs with connection integer $q = p_1 p_2$

Let $N = 2$, and connection integer $q = p_1 p_2$ where p_1 and p_2 are distinct odd primes and 2 is primitive module p_i. Suppose p_i is of the form $2r_i + 1$, where r_i is an odd prime. We have

$$T_i = ord_{p_i} 2 = p_i - 1 = 2r_i, \quad \text{for } i = 1, 2,$$

and

$$T = ord_q 2 = (p_1 - 1)(p_2 - 1)/2 = 2r_1r_2.$$

A characteristic polynomial is $f(x) = (x-1)(x^{r_1r_2} - 1)$. $2 \not| r_1r_2$, so

$$\begin{aligned} f(x) &= (x-1)(x^{r_1r_2} - 1) = (x-1)\prod_{d|r_1r_2} \Phi_d(x) \\ &= (x-1)(x-1)\Phi_{r_1}(x)\Phi_{r_2}(x)\Phi_{r_1r_2}(x) \end{aligned}$$

Let $q(x)$ be the minimal polynomial of a sequence generated by the 2-adic FCSRs with connection integer q. Then $q(x)|f(x)$ and $ord(q(x)) = T = 2r_1r_2$. So $q(x)$ can only have the following forms:

1. $q(x) = (x-1)^2 q_1(x)q_2(x)$ where $q_i(x) \neq 1$, $q_1(x)|\Phi_{r_1}(x)$ and $q_2(x)|\Phi_{r_2}(x)$.

2. $q(x) = (x-1)^2 q_3(x)q_4(x)$ where $q_4(x) \neq 1$, $q_4(x)|\Phi_{r_1r_2}(x)$ and $q_3(x)|\Phi_{r_1}(x)\Phi_{r_2}(x)$.

If $ord_{r_i} 2 = m_i$, then $ord_{r_1r_2} 2 = lcm(m_1, m_2)$ because of Theorem 2.2.5.

According to Proposition 2.2.1, $\Phi_{r_1}(x)$ factors into irreducible polynomials of degree m_1. Similarly, $\Phi_{r_2}(x)$ factorized into irreducible polynomials of degree m_2 and $\Phi_{r_1r_2}(x)$ factorized into irreducible polynomials of degree $\text{lcm}(m_1, m_2)$. So $deg(q(x)) \geq \min(2 + m_1 + m_2, 2 + lcm(m_1, m_2))$. The linear complexity of $\mathbf{a}$ generated by such FCSR is greater than or equal to $\min(2 + m_1 + m_2, 2 + lcm(m_1, m_2))$.

Case 2: N-adic l-sequence, where N is an odd prime

Because N is an odd prime, $\mathbb{Z}/(N)$ is a field with characteristic N. An l-sequence has a connection integer $q = p^e$ where p is an odd prime and $T = ord_q N = \phi(q) = p^{e-1}(p-1)$. Additionally,

$$q \equiv -1 \pmod{N}.$$

These conditions satisfy Lemma 2.2.1. One characteristic polynomial is

$$f(x) = (x-1)(x^{p^{e-1}(p-1)/2} - 1).$$

Let $p = 2r + 1$, where r is prime. We know that $N \neq r$, otherwise

$$q = p^e = (2N+1)^e \not\equiv -1 \pmod{N}.$$

So $N \not| p^{e-1}r$ and $N \not| 2p^{e-1}r$. From Theorem 2.2.1, we have

$$\begin{aligned}
& (x-1)(x^{p^{e-1}r}+1) \\
= \; & (x-1)\frac{x^{2p^{e-1}r}-1}{x^{p^{e-1}r}-1} \\
= \; & (x-1)\frac{\prod_{d|2p^{e-1}r}\Phi_d(x)}{\prod_{d|p^{e-1}r}\Phi_d(x)} \\
= \; & (x-1)\prod_{d|p^{e-1}r}\Phi_{2d}(x) \\
= \; & (x-1)\Phi_2(x)\Phi_{2p}(x)\Phi_{2p^2}(x)\cdots\Phi_{2p^{e-1}}(x)\Phi_{2r}(x)\Phi_{rp}(x)\Phi_{2rp^2}(x)\cdots\Phi_{2rp^{e-1}}(x).
\end{aligned}$$

Suppose $q(x)$ is the minimal polynomial. Then there are only two cases that can happen for $q(x)$, otherwise the period T cannot reach $2p^{e-1}r$.

1. There exist two irreducible polynomials $q_1(x)$ and $q_2(x)$ such that $q_1(x)q_2(x)|q(x)$. They satisfy $q_1(x)|\Phi_{2p^{e-1}}(x)$ and $q_2(x)|\Phi_{2rp^j}(x)$ for some $1 \le j < e-1$
2. There exists an irreducible polynomial $q_3(x)$ such that $q_3(x)|q(x)$ and $q_3(x)|\Phi_{2rp^{e-1}}(x)$.

According to Theorem 2.2.5 and Theorem 2.2.6, we have

$$ord_{2p^{e-1}}N = p^{e-2}(p-1),$$

because $ord_{p^{e-1}}N = p^{e-2}(p-1)$ and $ord_2 N = 1$. Similarly, for every $1 \le j < e$, we have

$$ord_{2p^j}N = p^{j-1}(p-1).$$

Let $ord_r N = m$. Then

$$ord_{2rp^j} = \mathrm{lcm}(m, p^{j-1}(p-1)) \text{ for any } 1 \le j \le e.$$

Based on the properties of cyclotomic polynomials, we have

$$\deg(q_1(x)) \ge ord_{2p^{e-1}}N = p^{e-2}(p-1),$$

$$\deg(q_2(x)) \ge ord_{2rp^j}N = \mathrm{lcm}(m, p^{j-1}(p-1)), 1 \le j < e,$$

and

$$\deg(q_3(x)) \ge ord_{2rp^{e-1}}N = \mathrm{lcm}(m, p^{e-2}(p-1)).$$

Let $\mathbf{a}$ be the generated sequence. So the linear complexity of $\mathbf{a}$ is

$$\lambda(\mathbf{a}) \ge \min\{p^{e-2}(p-1) + \mathrm{lcm}(m, p-1), \mathrm{lcm}(m, p^{e-2}(p-1))\}.$$

3 FCSR synthesis

3.1 Previous work on FCSR synthesis algorithms

We recall some basic facts about FCSR sequences. Let $\mathbf{a} = (a_0, a_1, \cdots)$ be an eventually periodic N-ary sequence. Then the associated N-adic number

$$a = \sum_{i=0}^{\infty} a_i N^i$$

is a quotient of two integers. That is, $a = f/q$ with $\gcd(f, q) = 1$. As we have discussed in Section 1.3.2, N-adic span can be used to measure the size of an FCSR. Instead we usually use the N-adic complexity, which is very close to the N-adic span. Let $\Phi_N(f, q) = \log_N(\max(|f|, |q|))$. The N-adic complexity $\lambda_N(\mathbf{a})$ is the minimum over all f, q with $a = f/q$ of $\Phi_N(f, q)$. So the FCSR synthesis problem can be rephrased as follows:

- **Given** a prefix $a_0, a_1, \cdots, a_{k-1}$ of an eventually periodic N-ary sequence $\mathbf{a}$.
- **Find** an integer pair (f, q) satisfying $a = f/q$ and minimizing $\Phi_N(f, q)$.

A useful description for FCSR synthesis algorithms is in terms of integer approximation lattices [30]. This notion is due to Mahler [43] and de Weger [14].

Definition 3.1.1. *[25] Let $a = a_0 + a_1 N + \cdots \in \mathbb{Z}_N$ be an N-adic integer. Its k-th approximation lattice is the set*

$$L_k = L_k(\mathbf{a}) = \{(h_1, h_2) \in \mathbb{Z} \times \mathbb{Z} : ah_2 - h_1 \equiv 0 \pmod{N^k}\}$$

An element $(f, q) \in L_k$ with q relatively prime to N represents a fraction f/q as a N-adic number agrees with that of a in the first k places. It will be shown that when k is large enough, f/q will equal a as an N-adic number. We introduce two rational approximation algorithms. One is based on the extended Euclidean algorithm and the other is based on lattice approximation. There is another algorithm, proposed by Xu and Klapper, which is a modified version of the Berlekamp-Massy algorithm [33]. It is even applicable to many more general AFSRs, so we introduce it in the next Chapter.

3.1.1 Rational approximation based on the extended Euclidean algorithm

The Euclidean algorithm is well-known for efficiently computing the greatest common divisor (GCD) of two integers. This algorithm can also be defined for more general rings. An *integral domain* R is said to be *Euclidean* if there exists a map $\psi : R\backslash\{0\} \to \mathbb{N}$ such that given any $a, b \in R$, there exist $q, r \in R$ such that $a = bq + r$ with either $r = 0$ or $\psi(r) < \psi(b)$. Any such ring is a principal ideal domain (PID). ψ is called a "Euclidean function", "degree function", "valuation function", or "norm function". Moreover, there are principal ideal domain which are not Euclidean but where the equivalent of the Euclidean algorithm can be defined [9].

The set of integers $\mathbb{Z}$ is an Euclidean domain with the Euclidean function defined as the absolute value, that is, $\psi(a) = |a|$ for all $a \in \mathbb{Z}$.

Theorem 3.1.1. *[63] (*Division with remainder property*) Let $a, b \in \mathbb{Z}$ with $b > 0$. Then there exist unique $q, r \in \mathbb{Z}$ such that $a = bq + r$ and $0 \leq r < b$.*

This procedure can be iterated :

$$\begin{aligned} a &= bq_1 + r_1 \\ b &= r_1 q_2 + r_2 \\ r_1 &= r_2 q_3 + r_3 \\ &\cdots \\ r_{n-1} &= q_{n+1} r_n \end{aligned} \tag{3.1}$$

At last, we get the greatest common divisor of a and b by the relation that

$$\gcd(a, b) = \gcd(b, r_1) = \cdots = \gcd(r_n, 0) = r_n.$$

The Euclidean algorithm is based on the division with remainder property and it can be implemented as in Figure 3.1.

```
1: procedure EA(a, b)
2:     if b = 0 then
3:         return a
4:     else
5:         return EA(b, a (mod b))
6:     end if
7: end procedure
```

Figure 3.1: The Euclidean algorithm

The overall running time of the Euclidean algorithm is proportional to the number of recursive calls it makes, times the time needed for division. The number of recursive calls is controlled by the Fibonacci numbers as shown in the following theorem.

Theorem 3.1.2. *[12] Let F_k be the Fibonacci numbers defined by $F_0 = 0$, $F_1 = 1$, and $F_i = F_{i-1} + F_{i-2}$ for $i \geq 2$. For any integer $k \geq 1$, if $a > b \geq 1$ and $b < F_k$, then the call EA(a,b) makes fewer than k recursive calls.*

Bézout's identity says that there are integers x and y such that

$$xa + yb = \gcd(a, b).$$

Such x and y can be computed through the extended Euclidean algorithm which is described in Figure 3.2. It takes as input a pair of nonnegative integers and returns a triple of the form $(\gcd(a, b), x, y)$ that satisfies Bézout's identity. The time complexity is asymptotically the same as the Euclidean algorithm.

```
1:  procedure EEA(a, b)
2:      (r_0, x_0, y_0) = (a, 1, 0)
3:      (r_1, x_1, y_1) = (b, 0, 1)
4:      while r_1 ≠ 0 do
5:          (q, r) = (r_0 (div r_1), r_0 (mod r_1))
6:          (x_3, y_3) = (x_0 − q x_1, y_0 − q y_1)
7:          (r_0, x_0, y_0) = (r_1, x_1, y_1)
8:          (r_1, x_1, y_1) = (r, x_3, y_3)
9:      end while
10:     return (r_0, x_0, y_0)
11: end procedure
```

Figure 3.2: The extended Euclidean algorithm

The Euclidean algorithm and the extended Euclidean algorithm can also be applied to the ring of polynomials over a field in the same manner because the ring of polynomials over a field is also a Euclidean domain with its Euclidean function defined as the degree of polynomials.

The algorithm given in Figure 3.3 is the rational approximation algorithm for FCSR synthesis based on the extended Euclidean algorithm. Suppose the first k symbols $a_0, a_1, \cdots, a_{k-1}$ of an N-ary sequence are available. We execute the extended Euclidean algorithm with $a = N^k$ and $b = a_0 + a_1 N + \cdots + a_{k-1} N^{k-1}$. Then we can obtain sequences of integers r_i, x_i, and y_i with $r_i = x_i a + y_i b$. That is,

$$y_i b - r_i \equiv 0 \pmod{N^k},$$

so $(r_i, y_i) \in L_k$. When $|r_i|$ first becomes less than $N^{(k-1)/2}$, $\Phi(r_i, y_i)$ is minimized [25]. Theorem 3.1.3 shows the number of terms needed. If $N = 2$ then given $2\lambda_2(\mathbf{a}) + 3$ bits, the EEAapprox outputs a description of the smallest FCSR that generates $\mathbf{a}$. For $N \neq 2$, suppose $p = r_0$ and $q = y_0$, where (r_0, x_0, y_0) is the output of EEAapprox. Then (p, q) is a pair of coprime integers but it may happen that $q \not\equiv -1 \pmod N$. We can multiply q with $u \equiv q^{-1} \pmod N$ and $1 \leq |u| < N/2$. Then up/uq is the rational number corresponding to an FCSR that outputs $\mathbf{a}$. Moreover, $\Phi_N(up, uq) = \log_N(\max(|up|, |uq|)) < (1 - \log_N 2) + \Phi_N(p, q)$, where $\Phi_N(p, q)$ is minimal among all the elements in $L_k(\mathbf{a})$

One problem for EEAAPPROX is that it is not adaptive. If a better approximation is needed, then the previous approximation is no longer useful and the entire algorithm must be started from the beginning. Arnault, Berger and Necer discussed some possible solutions. For more details please refer to [7].

procedure EEAAPPROX($a_0, \cdots, a_{k-1}$)
  if k is not odd **then**
    $k = k - 1$
  end if
  $(r_0, x_0, y_0) = (N^k, 1, 0)$
  $(r_1, x_1, y_1) = (\sum_{i=0}^{k-1} a_i N^i, 0, 1)$
  while $r_1 > N^{k/2}$ **do**
    Let $r_0 = qr_1 + r$
    $(x_3, y_3) = (x_0 - qx_1, y_0 - qy_1)$
    $(r_0, x_0, y_0) = (r_1, x_1, y_1)$
    $(r_1, x_1, y_1) = (r, x_3, y_3)$
  end while
  if $|y_1| \leq N^{k/2}$ **then**
    return (r_1, y_1)
  else
    return FALSE
  end if
end procedure

Figure 3.3: The extended Euclidean rational approximation algorithm

Theorem 3.1.3. *[25] Suppose that N is not a square and the N-adic complexity of the infinite sequence $a_0, a_1, \cdots$ is less than or equal to n. Suppose algorithm EEAapprox is executed with $k \geq 2n + 3$ and the algorithm outputs a pair of integers (r_1, y_1). Then*

$$\sum_{i=0}^{\infty} a_i N^i = \frac{r_1}{y_1},$$

r_1 and y_1 are relatively prime, and gcd(N, y_1)=1.

Theorem 3.1.4. *[25] The* EEAAPPROX *algorithm runs in time $O(k^2)$ if k elements of $\boldsymbol{a}$ are used.*

3.1.2 Rational approximation based on lattice approximation

The algorithm give in Figure 3.4 is the rational approximation algorithm based on lattice approximation, called LATTICEAPPROX. It has the same adaptive features as Berlekamp-Massey algorithm (Figure 2.1). For each k, the algorithm tries to find the smallest basis of the kth approximation lattice $L_k(\mathbf{a})$. As k grows, the minimal vector in $L_k(\mathbf{a})$ will give the rational expression of a. LATTICEAPPROX is for FCSR synthesis with $N = 2$. Theorem 3.1.5 shows that the smallest FCSR for a sequence $\mathbf{a}$ can be found with at most $2\lambda_2(\mathbf{a}) + \lceil 2\log_2(\lambda_2\mathbf{a})\rceil + 2$ bits. It is shown in [25] that the time complexity for LATTICEAPPROX is $O(T^2 \log T \log\log T)$.

Theorem 3.1.5. *[25] Suppose $\mathbf{a} = a_0, a_1, \ldots,$ is an eventually periodic sequence with associalted 2-adic integer $a = \sum_{i=0}^{\infty} a_i 2^i = f/q$, with $f, q \in \mathbb{Z}$, and $\gcd(f, q) = 1$. If $T \geq 2\lambda_2(\mathbf{a}) + \lceil 2\log_2(\lambda_2\mathbf{a})\rceil + 2$, then* LATTICEAPPROX *outputs $g = (f, q)$.*

3.2 Multi-sequences and joint N-adic complexity

The FCSR synthesis problem for multi-sequences is: given a prefix of each sequence $\mathbf{S}^{(h)}$, find a common generator of the smallest size that can generate all M sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$ (with a different initial state for each sequence). Let

$$\mathcal{S} = \left(\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}\right)$$

be an M-fold N-ary eventually periodic multi-sequence, as defined in Section 1.4. If $U(x) \in \mathbb{F}_q[x]$ is a characteristic polynomial for each of the M sequence, then it specifies the connection polynomial of an LFSR that generates each of the M sequences. However, we cannot solve the FCSR synthesis problem in the same way. This makes multi-sequence synthesis with FCSRs more complicated than with LFSRs. We proposed a new idea of adopting interleaving technique with π-adic numbers to study the problem. We derive two algorithms based on this method. One is based on the lattice reduction greedy algorithm proposed by Nguyen and Stehlé (Figure 1.9). The other is based on the LLL algorithm (Figure 1.8) which is a polynomial time lattice reduction algorithm. Both of these rational approximation algorithms can find the smallest common FCSR for a given multi-sequence but with different

```
1: procedure LatticeApprox(a_0, ..., a_{T-1})
2:     a = Σ_{i=0}^{T-1} a_i 2^i
3:     Let t be minimal with a_{t-1} = 1
4:     f = (0, 2)
5:     g = (2^{t-1}, 1)
6:     for (k = t, ..., T - 1) do
7:         if (a · g_2 - g_1 ≡ 0(mod 2^{k+1})) then
8:             if Φ_2(f) < Φ_2(g) then
9:                 f = 2f
10:            else
11:                Let d minimize Φ_2(f + dg)
12:                ⟨g, f⟩ = ⟨g, 2(f + dg)⟩
13:            end if
14:        else
15:            if Φ_2(f) < Φ_2(g) then
16:                Let d minimize Φ_2(f + dg) with d odd
17:                ⟨g, f⟩ = ⟨f + dg, 2g⟩
18:            else
19:                Let d minimize Φ_2(g + df) with d odd
20:                ⟨g, f⟩ = ⟨g + df, 2f⟩
21:            end if
22:        end if
23:    end for
24: end procedure
```

Figure 3.4: The rational approximation algorithm based on lattice approximation

numbers of known terms. If the number of sequences within the multi-sequence is less than or equal to 3, the one based on Nguyen and Stehlé's algorithm is suggested because it has better time complexity and fewer terms are needed. Otherwise, the one based on the LLL algorithm will be much better according to its time complexity.

We suppose that $x^M - N$ is irreducible over the rational field $\mathbb{Q}$ for $M \geq 2$ and $N \geq 2$. For eventually periodic multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$, each eventually periodic sequence $\mathbf{S}^{(h)}$ can be identified with an N-adic integer. That is, for $h = 0, 1, \cdots, M-1$, $a^{(h)} = \sum_{i \geq 0}^{\infty} s_i^{(h)} N^i \in \mathbb{Z}_N$ and $a^{(h)} = p^{(h)}/q^{(h)}$ for some $p^{(h)}, q^{(h)} \in \mathbb{Z}$.

Definition 3.2.1. *The joint N-adic complexity of multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$, $\lambda_{N,M}(\mathcal{S})$, is the size of the smallest FCSR that can generate all M sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots,$ and $\mathbf{S}^{(M-1)}$.*

To find a common FCSR that can generate all M sequences, we look into the

interleaved sequence of $\mathcal{S}$ which is defined as

$$(s_0^{(0)}, s_0^{(1)}, s_0^{(2)}, \ldots s_0^{(M-1)}, s_1^{(0)}, s_1^{(1)}, \ldots s_1^{(M-1)}, s_2^{(0)}, s_2^{(1)}, \ldots).$$

Take $\pi^M = N$, $R = \mathbb{Z}[\pi]$, and $S = \{0, 1, ..., N-1\}$. Because $x^M - N$ is irreducible over the rational field $\mathbb{Q}$ and $\pi \in \mathbb{C}$ is a root of this polynomial, it is true that R is an integral domain. The ring of π-adic numbers, R_π, contains the subring consisting of quotients γ/b with $\gamma \in R$ and $b \in \mathbb{Z}$ such that $\gcd(b, N) = 1$. We also have $\mathbb{Z}_N \subset R_\pi$.

The carries in each $\mathbf{S}^{(h)}$ $(h = 0, 1, \cdots M-1)$ are independent and will be out of order if the interleaved sequence is treated as an N-adic integer. To make the carries work properly within a single sequence, we associate the interleaved sequence with a π-adic number $\varsigma \in R_\pi$ as follows:

$$\varsigma = s_0^{(0)} + s_0^{(1)}\pi + s_0^{(2)}\pi^2 + \cdots + s_0^{(M-1)}\pi^{M-1} + s_1^{(0)}\pi^M + s_1^{(1)}\pi^{M+1} + \cdots . \tag{3.2}$$

The following theorem demonstrates how to find $\lambda_{N,M}(\mathcal{S})$ in terms of the π-adic number ς.

Theorem 3.2.1. *Let* $\varphi(\gamma) = \max(|r_0|, |r_1|, \ldots, |r_{M-1}|)$ *for any* $\gamma = r_0 + r_1\pi + r_2\pi^2 \cdots + r_{M-1}\pi^{M-1} \in \mathbb{Z}[\pi]$ *with* $r_i \in \mathbb{Z}$ $(i = 0, 1, \ldots, M-1)$. *The joint* N*-adic complexity of* $\mathcal{S}$, $\lambda_{N,M}(\mathcal{S})$, *is the minimal value of* $\log_N(\max(\varphi(\gamma), |q|))$, *where* $\gamma/q = \varsigma$ *(*ς *is the* π*-adic number associated with the interleaved sequence of* $\mathcal{S}$*),* $\gamma \in \mathbb{Z}[\pi]$ *and* $q \in \mathbb{Z}$.

Proof: Consider each sequence $\mathbf{S}^{(h)}$ $(h = 0, 1, \ldots, M-1)$. We have $a^{(h)} = \sum_{i \geq 0}^{\infty} s_i^{(h)} N^i = p^{(h)}/q^{(h)}$. Because $\pi^M = N$, we have

$$\begin{aligned}
\varsigma &= a^{(0)} + a^{(1)}\pi + a^{(2)}\pi^2 + \cdots + a^{(M-1)}\pi^{M-1} \\
&= \frac{p^{(0)}}{q^{(0)}} + \frac{p^{(1)}}{q^{(1)}}\pi + \frac{p^{(2)}}{q^{(2)}}\pi^2 + \cdots + \frac{p^{(M-1)}}{q^{(M-1)}}\pi^{M-1} \\
&= \frac{\gamma}{\mathrm{lcm}(q^{(0)}, q^{(1)}, \cdots, q^{(M-1)})} \quad \text{for some } \gamma \in \mathbb{Z}[\pi].
\end{aligned}$$

Here, $\mathrm{lcm}(q^{(0)}, q^{(1)}, \cdots, q^{(M-1)})$ denotes the least common multiple of $q^{(0)}$, $q^{(1)}$, $\cdots$, $q^{(M-1)}$. The FCSR with connection integer $\mathrm{lcm}(q^{(0)}, q^{(1)}, \cdots, q^{(M-1)})$ can generate all these M sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$ with proper initial settings determined by γ. So ς has a rational expression of the form γ/q, where $\gamma \in \mathbb{Z}[\pi]$ and $q \in \mathbb{Z}$.

On the other hand, if $\varsigma = \gamma/q$ for some $\gamma = r_0 + r_1\pi + r_2\pi^2 \cdots + r_{M-1}\pi^{M-1} \in \mathbb{Z}[\pi]$ and $q \in \mathbb{Z}$, then $r_h/q = a^{(h)}$, $h = 0, 1, \cdots, M-1$. It means that the FCSR with connection integer q and the particular initial setting related to r_h can generate

sequence $\mathbf{S}^{(h)}$. Considering all the M initial settings together, the size of the FCSR is

$$\log_N(\max(\varphi(\gamma), |q|)) = \log_N(\max(|r^0|, |r^1|, \ldots, |r^{M-1}|, |q|)).$$

So according to the definition, $\lambda_{N,M}(\mathcal{S})$ is the minimal value of $\log_N(\max(\varphi(\gamma), |q|))$. □

Thanks to Theorem 3.2.1, the problem of FCSR synthesis for multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$, where $\mathbf{S}^{(h)}$ is an eventually periodic N-ary sequence, can be rephrased as follows:

- **Given** A prefix of the interleaved sequence of $\mathcal{S}$
- **Find** $\gamma \in \mathbb{Z}[\pi]$ and $q \in \mathbb{Z}$ satisfying $\varsigma = \gamma/q$ and minimizing $\max(\varphi(\gamma), |q|)$.

The joint N-adic complexity and joint 2-adic complexity have been discussed under the assumption that sequences $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$ are all strictly periodic [29,71,72]. That is, $|p^{(h)}| \leq |q^{(h)}|$ for $h = 0, 1, \ldots, M-1$. So,

$$\lambda_{N,M}(\mathcal{S}) = \log_N(\mathrm{lcm}(q^{(0)}, q^{(1)}, \cdots, q^{(M-1)})).$$

Results about the expected value, upper bound, and lower bound of the joint complexity $\lambda_{N,M}(\mathcal{S})$ in this case were proved by Hu, et al. [29], and Yang, et al. [71].

3.3 Rational approximation for multi-sequences

To solve the problem of FCSR synthesis for the multi-sequence $\mathcal{S}$, we define an integer lattice using the first k consecutive terms of the interleaved sequence of $\mathcal{S}$. We assume that M divides k, which means that the known prefixes of each $\mathbf{S}^{(h)}$ are of the same length, k/M. The elements in the formed integer lattice will determine approximations of the rational expression of the interleaved sequence. Finding a minimal vector in this integer lattice will result in a best rational expression of the associated π-adic number. In other words, the best common FCSR that can generate the multi-sequence $\mathcal{S}$ will have been found.

Definition 3.3.1. *Let $\pi^M = N$ where M, N are positive integers such that $x^M - N$ is an irreducible polynomial over the rational numbers $\mathbb{Q}$. Let $R = \mathbb{Z}[\pi]$ and $\varsigma \in R_\pi$, the ring of π-adic numbers. The* kth integer approximation lattice *of ς is defined as*

$$L_k(\varsigma) := \{(u_0, \ldots, u_{M-1}, v) \in \mathbb{Z}^{M+1} : \varsigma v - (u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}) \equiv 0 \pmod{\pi^k}\}.$$

It can be shown that $L_k(\varsigma)$ is a lattice, because it is closed under addition and scalar multiplication.

Denote the interleaved sequence of $\mathcal{S}$ by $\mathbf{S}$, where $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$. Let $\varsigma \in R_\pi$ be the π-adic number that is associated with $\mathbf{S}$. Then, for any vector $(u_0, \ldots, u_{M-1}, v) \in L_k(\varsigma)$, we have

$$\varsigma \equiv \frac{u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}}{v} \pmod{\pi^k} \quad \text{if} \quad \gcd(v, N) = 1.$$

That is, $(u_0+u_1\pi+\cdots+u_{M-1}\pi^{M-1})/v$ is a rational approximation of ς up to k terms.

Let $\varsigma_k \in \mathbb{Z}[\pi]$ be associated with the first k terms of $\mathbf{S}$ as follows.

$$\begin{aligned} \varsigma_k &= s_0^{(0)} + s_0^{(1)}\pi + s_0^{(2)}\pi^2 + \cdots + s_{\frac{k}{M}-1}^{(0)}\pi^{k-1} + \cdots + s_{\frac{k}{M}-1}^{(M-2)}\pi^{k-1} + s_{\frac{k}{M}-1}^{(M-1)}\pi^k \\ &= s_0 + s_1\pi + \cdots + s_{M-1}\pi^{M-1}, \text{ for some } s_0, s_1, \ldots, s_{M-1} \in \mathbb{Z}. \end{aligned}$$

It can be verified that the vectors $\mathbf{u}_1 = (N^{k/M}, \ldots, 0)$, $\mathbf{u}_2 = (0, N^{k/M}, \ldots, 0)$, $\cdots$, $\mathbf{u}_{M-1} = (0, \ldots, N^{k/M}, 0)$, $\mathbf{u}_M = (0, \ldots, N^{k/M})$ and $\mathbf{u}_{M+1} = (s_0, s_1, \ldots, s_{M-1}, 1)$ are all in $L_k(\varsigma)$.

Theorem 3.3.1. *$L_k(\varsigma)$ is a full lattice of rank $M+1$ and $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1}$ form a basis of $L_k(\varsigma)$.*

Proof: It is true that $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1}$ are linearly independent vectors in $\mathbb{R}^{M+1}$. We will show that $L_k(\varsigma) = L(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$.
Let $\mathbf{u} = \sum_{i=1}^{M+1} c_i\mathbf{u}_i$, where $c_i \in \mathbb{Z}$. We have $\mathbf{u} \in L_k(\varsigma)$, because $L_k(\varsigma)$ is closed under addition and scalar multiplication. So $L(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1}) \subseteq L_k(\varsigma)$.
Suppose $\mathbf{v} = (v_0, v_1, \ldots, v_M)$ is an arbitratry vector in $L_k(\varsigma)$. Then we have

$$\varsigma v_M - (v_0 + v_1\pi + v_1\pi^2 + \cdots + v_{M-1}\pi^{M-1}) \equiv 0 \pmod{\pi^k}.$$

So there exists $\gamma \in \mathbb{Z}[\pi]$ such that

$$\varsigma_k v_M - (v_0 + v_1\pi + v_2\pi^2 + \cdots + v_{M-1}\pi^{M-1}) = \gamma\pi^k = \gamma N^{k/M}.$$

Let $\gamma = r_0 + r_1\pi + r_2\pi^2 + \cdots + r_{M-1}\pi^{M-1}$, where $r_i \in \mathbb{Z}$. Making corresponding terms equal, we have

$$\begin{aligned} &s_0 v_M - v_0 = r_0 N^{k/M}, \\ &s_1 v_M - v_1 = r_1 N^{k/M}, \\ &s_2 v_M - v_2 = r_2 N^{k/M}, \\ &\cdots \\ &s_{M-1} v_M - v_{M-1} = r_{M-1} N^{k/M}. \end{aligned}$$

So $\mathbf{v} = v_M\mathbf{u_{M+1}} - r_0\mathbf{u_1} - r_1\mathbf{u_2} - \cdots - r_{M-1}\mathbf{u_M}$ which is a linear combination of $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$. That is, $\mathbf{v} \in L(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$. $L_k(\varsigma) \in \mathbb{R}^{M+1}$ is a lattice with the same dimension of the space $\mathbb{R}^{M+1}$, so it is full.

□

Recalling the definition of the determinant of a lattice L, we have

$$\det(L_k(\varsigma)) = (\det(G(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})))^{\frac{1}{2}} = N^k.$$

Suppose we have two vectors $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$, where $\tilde{\mathbf{u}} = (\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_{M-1}, \tilde{v}) \in L_k(\varsigma)$ has the smallest Euclidean norm and $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{M-1}, \hat{v}) \in L_k(\varsigma)$ has the smallest super norm. Then, according to the inequalities (1.6) and (1.7),

$$||\tilde{\mathbf{u}}||_\infty \leq \sqrt{M+1}||\hat{\mathbf{u}}||_\infty. \tag{3.3}$$

Also, we have $||\hat{\mathbf{u}}||_\infty \leq \det(L_k(\varsigma))^{\frac{1}{M+1}} = N^{\frac{k}{M+1}}$ owing to Minkowski's bound on lattices [15] and

$$||\tilde{\mathbf{u}}||_\infty \leq \sqrt{M+1}||\hat{\mathbf{u}}||_\infty \leq \sqrt{M+1}N^{\frac{k}{M+1}}. \tag{3.4}$$

We want to use $\tilde{\mathbf{u}}$ or $\hat{\mathbf{u}}$ in $L_k(\varsigma)$ to approximate the best rational expression of the interleaved sequence, but finding them is a hard problem in lattice theory.

3.4 Multi-sequences FCSR synthesis via lattice approximation

In this section, we introduce two approximation algorithms. One, called ApproxGreedy (Figure 3.5), is based on the lattice reduction greedy algorithm (Figure 1.9) and the other, called ApproxLLL (Figure 3.6), is based on the LLL algorithm (Figure 1.8). Given a multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$, the interleaved sequence $\mathbf{S}$ is associated with a π-adic number ς. Suppose the first k consecutive terms of $\mathbf{S}$ are known and $M|k$. Then each of the rational approximation algorithms can compute a vector in the kth integer approximation lattice, $L_k(\varsigma)$, that almost has the minimal super norm. When k is sufficiently large, both of their outputs are exactly a rational expression of the smallest common FCSR that can generate the multi-sequence $\mathcal{S}$. But the minimum k that suffices for the two algorithms differs. In other words, they both solve the problem of FCSR synthesis for multi-sequences.

3.4.1 Rational approximation algorithm based on the lattice reduction greedy algorithm

The lattice reduction greedy algorithm was proposed by Nguyen and Stehlé [50] in 2009. Figure 1.9 is an iterative description of it. ApproxGreedy, based on the lattice reduction greedy algorithm, is given in Figure 3.5. The inputs are the first k/M consecutive terms of each sequence $\mathbf{S}^{(i)}$ for $(0 \leq i \leq M-1)$, so the total number of input terms is k. The output is a pair (β, q) where $\beta \in \mathbb{Z}[\pi]$ and $q \in \mathbb{Z}$. Theorem 3.4.1 shows that when k is large enough, q will be the connection integer of the smallest FCSR that can generate the multi-sequence $\mathcal{S}$. The output β determines the initial state for each $\mathbf{S}^{(i)}$.

procedure ApproxGreedy($\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$)
Input: $s_0^{(i)}, s_1^{(i)}, \ldots, s_{\frac{k}{M}-1}^{(i)}$, the first k/M terms of sequence $\mathbf{S}^{(i)}$ $(0 \leq i \leq M-1)$.
Output: (β, q) where $\beta \in \mathbb{Z}[\pi], q \in \mathbb{Z}$.
    for $j = 1$ to $M-1$ **do**
        $s_j := \sum_{i=0}^{\frac{k}{M}-1} s_i^{(j)} N^i$
    end for
    $\mathbf{u}_{M+1} := (s_0, s_1, \ldots, s_{M-1}, 1)$
    for $i = 1$ to M **do**
        $\mathbf{u}_i := (0, 0, \ldots, \underbrace{N^{k/M}}_{ith\ postion}, \ldots, 0)$
    end for
    Sort $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1}$ by their norm $\|\cdot\|$ so $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$ is an ordered basis
    Compute the Gram matrix G so that $G_{ij} = \langle \mathbf{u}_i, \mathbf{u}_j \rangle$
    $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$:=GreedyLatticeReduction$(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$
    Suppose $\mathbf{u}_1 = (u_0, u_1, \ldots, u_{M-1}, v)$
    return $(u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}, v)$
end procedure

Figure 3.5: The multi-FCSR Rational Approximation with GreedyLatticeReduction

Theorem 3.4.1. *Let $\pi^M = N$ where $x^M - N$ be irreducible over the rationals. Suppose $M \leq 3$. Let ς be the π-adic number identified with the interleaved sequence of $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$. So $L_k(\varsigma)$ is of dimension at most four. Let the joint N-adic complexity of $\mathcal{S}$ be less than or equal to n. Suppose* ApproxGreedy *is executed with $k \geq \max(\lfloor 2M \cdot n + M \cdot \log_N(2\sqrt{M+1}) + 1 \rfloor, \lfloor M(M+1)\log_N(\sqrt{M+1}) + 1 \rfloor)$ and outputs (β, q), where $\beta = b_0 + b_1\pi + + \cdots + b_{M-1}\pi^{M-1} \in \mathbb{Z}[\pi], q \in \mathbb{Z}$. Then for*

$0 \le i \le M-1$,

$$\sum_{j=0}^{\infty} s_j^{(i)} N^j = \frac{b_i}{q},$$

where $(s_0^{(i)}, s_1^{(i)}, s_2^{(i)}, \dots)$ *is the sequence* $\mathbf{S}^{(i)}$. *This means that* b_i/q *is a rational expression of sequence* $\mathbf{S}^{(i)}$. *The value of* $\max(\varphi(\beta), |q|)$ *is equal to* $\lambda_{N,M}(\mathcal{S})$, *which implies that* $|q|$ *is the connection integer of the smallest common FCSR that can generate* $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \dots, \mathbf{S}^{(M-1)}$. *For simplicity, we always suppose the denominators of all the rational forms are positive.*

Proof: (β, q) is the output of ApproxGreedy, so $\mathbf{v} = (b_0, b_1 \dots, b_{M-1}, q)$ is the first vector of the Minkowski reduced basis. It is true that $\gcd(b_0, b_1, \dots, b_{M-1}, q) = 1$. Otherwise, $(b_0, b_1 \dots, b_{M-1}, q)$ cannot be the smallest vector. For any other $\mathbf{v}' \in L_k(\varsigma)$, we have $||\mathbf{v}|| \le ||\mathbf{v}'||$.

First, we show that $q \neq 0$. If $q = 0$, then $\beta \equiv 0 \pmod{\pi^k}$. So if $\beta \neq 0$ then $\varphi(\beta) \ge N^{k/M}$. Hence,

$$||\mathbf{v}|| \ge ||\mathbf{v}||_\infty = \max(\varphi(\beta), |q|) \ge N^{k/M}.$$

But according to equation (3.4), $||\mathbf{v}|| \le \sqrt{M+1} N^{\frac{k}{M+1}}$. This is impossible because $N^{k/M} > \sqrt{M+1} N^{\frac{k}{M+1}}$, when $k > M(M+1)\log_N(\sqrt{M+1})$. So $\beta = q = 0$, which is false.

Suppose $\mathbf{u} = (u_0, u_1, \dots, u_{M-1}, p) \in \mathbb{Z}^{M+1}, \gcd(p, N) = 1$ such that

$$\varsigma = \frac{u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}}{p},$$

and $||\mathbf{u}||_\infty = \lambda_{N,M}(\mathcal{S})$. That is, $(u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1})/p$ is a best rational expression of ς. Then $\gcd(u_0, u_1, \dots, u_{M-1}, p) = 1$, since otherwise the joint N-adic complexity would be smaller. So we have

$$||\mathbf{u}||_\infty = \max(|u_0|, |u_1|, \dots, |u_{M-1}|, |p|) \le N^n.$$

Suppose $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \hat{u}_2, \dots, \hat{u}_{M-1}, \hat{v})$ has the smallest sup norm in $L_k(\varsigma)$. Then $||\hat{\mathbf{u}}||_\infty \le ||\mathbf{u}||_\infty \le N^n$ because $\mathbf{u} \in L_k(\varsigma)$. Also, $\mathbf{v}$ has the smallest Euclidean norm, so

$$||\mathbf{v}||_\infty \le \sqrt{M+1}||\hat{\mathbf{u}}||_\infty \le \sqrt{M+1} \cdot N^n.$$

We have

$$q \cdot \frac{u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}}{p} \equiv b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1} \pmod{\pi^k},$$

so there exists $\gamma \in \mathbb{Z}[\pi]$ such that

$$q(u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}) - p(b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1}) = pqN^{k/M}\gamma.$$

Suppose $\gamma = r_0 + r_1\pi + \cdots + r_{M-1}\pi^{M-1}$ with $r_i \in \mathbb{Z}$, so

$$\begin{aligned} qu_0 - pb_0 &= pqr_0N^{k/M}, \\ qu_1 - pb_1 &= pqr_1N^{k/M}, \\ &\cdots, \\ qu_{M-1} - pb_{M-1} &= pqr_{M-1}N^{k/M}. \end{aligned}$$

If $\gamma \neq 0$, then $r_i \neq 0$ for some $0 \leq i \leq M-1$. WLOG, let $i = 0$.

$$|qu_0 - pb_0| \leq 2 \cdot ||\mathbf{u}||_\infty \cdot ||\mathbf{v}||_\infty \leq 2\sqrt{M+1} \cdot N^{2n}.$$

But

$$|pqr_0N^{k/M}| \geq |N^{k/M}| > 2\sqrt{M+1} \cdot N^{2n},$$

because $k > 2Mn + M \cdot \log_N(2\sqrt{M+1})$. This is a contradiction. So $\gamma = 0$. This means that

$$q(u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}) = p(b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1})$$

Because $\gcd(b_0, b_1, \ldots, b_{M-1}, q) = 1$, we have $q|p$. Similarly, $p|q$. So $p = q$ and $b_i = u_i$ for $0 \leq i \leq M-1$. Thus

$$\varsigma = \beta/q = \frac{b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1}}{q}.$$

In addition, $\gcd(q, N) = 1$ and $\max(\varphi(\beta), |q|) = ||\mathbf{v}||_\infty = \lambda_{N,M}(\mathcal{S})$. □

Consider the case of 3-fold binary multi-sequences. That is, let $M = 3$ and $N = 2$. Theorem 3.4.1 shows that the number of known terms required for ApproxGreedy to output the smallest common FCSR is at most $\max(6n + 7, 13)$, where n is the 2-adic joint complexity of the given multi-sequence.

Theorem 3.4.2. *Let $M \leq 3$. The algorithm* ApproxGreedy *runs in time $O(k^2)$ if k elements of $\mathbf{S}$ are used, where $\mathbf{S}$ is the interleaved sequence of the multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$.*

Proof: The time complexity of the multiplication of two integers that are no more than $N^{k/M}$ is $O(k \log k \log\log k)$ if Fast Fourier Transforms are used. So the time

complexity of obtaining $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1}$ from step 4 to step 10 in Figure 3.5 is $O(k \log^2 k \log\log k)$. We have $||\mathbf{u}_i|| = N^{k/M}, (1 \le i \le M)$ and

$$||\mathbf{u}_{M+1}|| = \sqrt{s_0^2 + s_1^2 + \cdots + s_{M-1}^2 + 1} < \sqrt{M \cdot N^{2k/M} + 1}.$$

The sorting and computation of the Gram matrix G both have time complexity of $O(k^2)$ because the dimension of the $L_k(\varsigma)$ is fixed. According to Theorem 1.5.1, the time complexity of the lattice basis reduction step is bounded by $O\big(\log ||\mathbf{u}_{M+1}||[1 + ||\mathbf{u}_{M+1}|| - \log \zeta_1(L_k(\varsigma)]\big) = O(k^2)$. So the time complexity of the algorithm ApproxGreedy is $O(k^2)$.

□

3.4.2 Rational approximation algorithm based on the LLL algorithm

The rational approximation algorithm based on the LLL algorithm, ApproxLLL, is given in Figure 3.6. Notice that it takes the same inputs as ApproxGreedy which are also the first k/M consecutive terms of each sequence $\mathbf{S}^{(i)}$ for $(0 \le i \le M-1)$. The output pair (β', q'), where $\beta' \in \mathbb{Z}[\pi]$ and $q' \in \mathbb{Z}$, is formed from the first vector of the LLL-reduced basis obtained from Step 11. We assume that the LLL algorithm runs with $\delta = 3/4$. Actually, other values of δ will also work and the analysis is similar. Theorem 3.4.3 shows that when k is large enough, q' will be the connection integer of the smallest FCSR that can generate $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$. Compared with Theorem 3.4.1, the number of inputs needed for ApproxLLL is about $\frac{M^2}{2} \log_N 2$ more additively. This is because the first vector in the LLL reduced basis is not the smallest nonzero vector but its Euclidean norm is less than or equal to $2^{M/2} \cdot \zeta_1(L_k(\varsigma))$ when $\delta = 3/4$.

Theorem 3.4.3. *Let $\pi^M = N$ where $x^M - N$ is irreducible over the rational numbers. Let ς be the π-adic number identified with the interleaved sequence of $\mathcal{S}$. So $L_k(\varsigma)$ is a $(M+1)$-dimensional full lattice. Let the joint N-adic complexity of $\mathcal{S}$ be less than or equal to n. Suppose* ApproxLLL *is executed with $k \ge \max(\lfloor 2M \cdot n + M \cdot \log_N(\sqrt{M+1}) + \frac{M^2+2M}{2} \log_N 2 + 1 \rfloor, \lfloor M(M+1) \log_N(\sqrt{M+1}) + \frac{M^3+M^2}{2} \log_N 2 + 1 \rfloor)$ and outputs (β', q'), where $\beta' = b'_0 + b'_1\pi + + \cdots + b'_{M-1}\pi^{M-1} \in \mathbb{Z}[\pi], q' \in \mathbb{Z}$. Then for $0 \le i \le M-1$,*

$$\sum_{j=0}^{\infty} s_j^{(i)} N^j = \frac{b'_i}{q'},$$

where $(s_0^{(i)}, s_1^{(i)}, s_2^{(i)}, \ldots)$ is the sequence $\mathbf{S}^{(i)}$. This means that b'_i/q' is a rational expression of sequence $\mathbf{S}^{(i)}$. The value of $\max(\varphi(\beta'), |q'|)$ is equal to $\lambda_{N,M}(\mathcal{S})$, which

procedure APPROXLLL($\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$)
Input: $s_0^{(i)}, s_1^{(i)}, \ldots, s_{\frac{k}{M}-1}^{(i)}$. The first k/M terms of sequence $\mathbf{S}^{(i)}$ $(0 \le i \le M-1)$.
Output: (β', q') where $\beta' \in \mathbb{Z}[\pi], q' \in \mathbb{Z}$.
    for $j = 1$ to $M-1$ **do**
        $s_j := \sum_{i=0}^{\frac{k}{M}-1} s_i^{(j)} N^i$
    end for
    $\mathbf{u}_{M+1} := (s_0, s_1, \ldots, s_{M-1}, 1)$
    for $i = 1$ to M **do**
        $\mathbf{u}_i := (0, 0, \ldots, \underbrace{N^{k/M}}_{ith\ postion}, \ldots, 0)$
    end for
    $(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$:=LLL$(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{M+1})$
    Suppose $\mathbf{u}_1 = (u_0, u_1, \ldots, u_{M-1}, v)$
    return $(u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}, v)$
end procedure

Figure 3.6: The multi-FCSR Rational Approximation with LLL

implies that $|q'|$ is the connection integer of the smallest common FCSR that can generate $\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)}$. For simplicity, we always suppose the denominators of all the rational forms are positive.

Proof: Let $\tilde{\mathbf{u}} = (\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_{M-1}, \tilde{v}) \in L_k(\varsigma)$ be a vector with the smallest Euclidean norm. (β', q') is the output of APPROXLLL, so $\mathbf{v} = (b'_0, b'_1 \ldots, b_{M-1}, q')$ is the first vector of the LLL reduced basis. So

$$||\mathbf{v}|| \le 2^{M/2}||\tilde{\mathbf{u}}||.$$

First, we show that $q \neq 0$. If $q = 0$, then $\beta \equiv 0 \pmod{\pi^k}$. So $\varphi(\beta) \ge N^{k/M}$, and thus

$$||\mathbf{v}|| \ge ||\mathbf{v}||_\infty = \max(\varphi(\beta), |q|) \ge N^{k/M}.$$

But according to inequality (3.4), $||\tilde{\mathbf{u}}|| \le \sqrt{M+1}N^{\frac{k}{M+1}}$, so

$$||\mathbf{v}|| \le 2^{M/2}\sqrt{M+1}N^{\frac{k}{M+1}}.$$

This is impossible because $N^{k/M} > 2^{M/2}\sqrt{M+1}N^{\frac{k}{M+1}}$ when

$$k > M(M+1)\log_N(\sqrt{M+1}) + \frac{1}{2}M^2(M+1)\log_N 2.$$

Suppose $\mathbf{u} = (u_0, u_1, \ldots, u_{M-1}, p) \in \mathbb{Z}^{M+1}, p \neq 0$ such that

$$\varsigma = \frac{u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}}{p},$$

and $||\mathbf{u}||_\infty = \Phi_N(\mathbf{S}^{(0)}, \mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(M-1)})$. Then $\gcd(u_0, u_1, \ldots, u_{M-1}, p) = 1$, otherwise the joint N adic-complexity would be smaller. So we have

$$||\mathbf{u}||_\infty = \max(|u_0|, |u_1|, \ldots, |u_{M-1}|, |p|) \leq N^n.$$

Suppose $\hat{\mathbf{u}} = (\hat{u}_0, \hat{u}_1, \hat{u}_2, \ldots, \hat{u}_{M-1}, \hat{v}) \in L_k(\varsigma)$ has the smallest L_∞ norm. Then $||\hat{\mathbf{u}}||_\infty \leq ||\mathbf{u}||_\infty \leq N^n$ because $\mathbf{u} \in L_k(\varsigma)$. So we have

$$||\mathbf{v}||_\infty \leq ||\mathbf{v}|| \leq 2^{M/2}||\tilde{\mathbf{u}}|| \leq 2^{M/2}\sqrt{M+1}||\hat{\mathbf{u}}||_\infty \leq 2^{M/2}\sqrt{M+1} \cdot N^n.$$

We have

$$\frac{u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}}{p} = \frac{b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1}}{q} \pmod{\pi^k},$$

so there exists $\gamma \in \mathbb{Z}[\pi]$ such that

$$q(u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}) - p(b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1}) = pqN^{k/M}\gamma.$$

Suppose $\gamma = r_0 + r_1\pi + \cdots + r_{M-1}\pi^{M-1}$ so

$$\begin{cases} qu_0 - pb_0 = pqr_0N^{k/M} \\ qu_1 - pb_1 = pqr_1N^{k/M} \\ \cdots \\ qu_{M-1} - pb_{M-1} = pqr_{M-1}N^{k/M}. \end{cases}$$

If $\gamma \neq 0$, then $r_i \neq 0$ for some $0 \leq i \leq M-1$. WLOG, let $i = 0$.

$$|qu_0 - pb_0| \leq 2 \cdot ||\mathbf{u}||_\infty \cdot ||\mathbf{v}||_\infty \leq 2^{M/2+1}\sqrt{M+1} \cdot N^{2n}.$$

But

$$|pqr_0N^{k/M}| \geq |N^{k/M}| > 2^{M/2+1}\sqrt{M+1} \cdot N^{2n},$$

because $k > 2Mn + M \cdot \log_N(2\sqrt{M+1}) + \frac{M^2+2M}{2}\log_N 2$. This is a contradiction. So $\gamma = 0$. This means that

$$\frac{b_0 + b_1\pi + \cdots + b_{M-1}\pi^{M-1}}{q} = \frac{u_0 + u_1\pi + \cdots + u_{M-1}\pi^{M-1}}{p} = \varsigma.$$

So it is proved that $\varsigma = \frac{\beta}{q}$ and $\max(\varphi(\beta), |q|) = ||\mathbf{v}||_\infty = \lambda_{N,M}(\mathcal{S})$. □

Theorem 3.4.4. *The algorithm* ApproxLLL *runs in time* $O(k^2 \log k \log\log k)$ *if* k *elements of* $\mathbf{a}$ *are used.*

Proof: From theorem 3.4.2, we know that $||\mathbf{u}_i|| = N^{k/M}, (1 \leq i \leq M)$ and

$$||\mathbf{u}_{M+1}|| = \sqrt{s_0^2 + s_1^2 + \cdots + s_{M-1}^2 + 1} < \sqrt{M \cdot N^{2k/M} + 1}.$$

Given a d-dimensional integer lattice basis with vectors of Euclidean norm less than B in a d-dimensional space, the time complexity of the LLL algorithm is $O(d^4 \log B \cdot \mathcal{M}(d \log B))$ bit operations, where $\mathcal{M}(d \log B)$ denotes the time required to multiply $d \log B$-bit integers [49]. In ApproxLLL, $d = M + 1$ and B can be chosen as $\sqrt{M+1}N^{k/M}$. We have $\mathcal{M}(d \log B) = O(k \log k \log \log k)$, if Fast Fourier Transforms are used and if M is fixed. So the time complexity of step 11 in Figure 3.4.3 is $O(k^2 \log k \log \log k)$. The other steps don't cost more according to the discussion in Theorem 3.4.2. So the time complexity of ApproxLLL is $O(k^2 \log k \log \log k)$.

□

3.4.3 Comparison of ApproxGreedy and ApproxLLL

ApproxGreedy and ApproxLLL are both rational approximation algorithms that can solve the problem of FCSR synthesis for multi-sequences. When $M = 1$, the multi-sequence N-adic FCSR synthesis problem is reduced to the single sequence synthesis problem. ApproxGreedy and ApproxLLL still work, so ApproxGreedy and ApproxLLL can be thought of as a generalization of lattice approximation algorithm (Figure 3.4) to multi-sequences cases but they are not adaptive. According to the time complexity and number of terms needed, ApproxGreedy is better than ApproxLLL when $M \leq 3$. When $M > 3$, ApproxGreedy may not output the right rational expression due to Theorem 1.5.1, so ApproxLLL should be used.

4 AFSR Synthesis

We recall the definition of AFSRs over (R, π, S) in Section 1.3.3, where R is an integral domain, $\pi \in R$ and S is a complete set of representative of $R/(\pi)$. Fix an eventually periodic sequence $\mathbf{a} = a_0, a_1, a_2, \cdots$ of S and denote its corresponding π-adic number by α. That is,

$$\alpha = a_0 + a_1\pi + a_2\pi^2 + \cdots .$$

According to Theorem 1.3.1, α has a rational expression u/q. If (u, q) is found, then the AFSR that generates sequence $\mathbf{a}$ can be constructed by Theorem 1.3.1. So our goal is to find a rational expression u/q using as few terms of sequence $\mathbf{a}$ as we can.

The AFSR synthesis problem is :

- **Given** A prefix of the eventually periodic sequence $\mathbf{a}$=$a_0, a_1, \cdots$ over $R/(\pi)$.
- **Find** $f, q \in R$ such that $\alpha = u/q$.

Xu's rational approximation algorithm [25, 33], proposed by Xu and Klapper, is a modificaion of the Berlekamp-Massey algorithm (Figure 2.1) that solves the LFSR synthesis problem. It solves the synthesis problem for AFSRs over (R, π, S) with certain algebraic properties, which we introduce in Section 4.1. We approach the AFSR synthesis problem with two different methods. One can be seen as an extension of the lattice approximation approach (Figure 3.4) and is introduced in Section 4.3. The other one, in Section 4.4, is an approximation algorithm based on the extended Euclidean algorithm on norm-Euclidean imaginary quadratic fields.

4.1 Xu's rational approximation algorithm

Xu's rational approximation algorithm is a modificaion of the Berlekamp-Massey algorithm in the sense that at each stage i, it maintains a rational element whose π-adic expansion is coincident with the given π-adic number up to i terms. As in the Berlekamp-Massey algorithm, the discrepancy is controlled at each stage. But they do so by forming a more general linear combination between two previous rational approximations. When constructing these linear combinations, several new terms are considered together to compensate for the increase in size due to the carry. Given R, π, and S, two structures are needed to make Xu's rational approximation algorithm work: size function and interpolation set.

Size function [25]

To measure the "size" of the elements of R, Xu and Klapper introduced a function $\psi_{R,\pi} : R \to \mathbb{Z} \bigcup \{-\infty\}$ satisfying the following properties for some constants b and c:

1. $\psi_{R,\pi}(0) = -\infty$ and $\psi_{R,\pi}(x) \geq 0$ if $x \neq 0$;
2. for all $x, y \in R$ we have $\psi_{R,\pi}(xy) \leq \psi_{R,\pi}(x) + \psi_{R,\pi}(y) + b$;
3. for all $x, y \in R$ we have $\psi_{R,\pi}(x \pm y) \leq \max\{\psi_{R,\pi}(x), \psi_{R,\pi}(y)\} + c$;
4. for all $x \in R$ and $k \geq 0 \in \mathbb{Z}$, we have $\psi_{R,\pi}(\pi^k x) = k + \psi_{R,\pi}(x)$.

Define a "height" function

$$\Gamma_{R,\pi}(x, y) = \max\{\psi_{R,\pi}(x), \psi_{R,\pi}(y)\},$$

and the "π-adic complexity" of a sequence,

$$\lambda_\pi(\mathbf{a}) = \inf\{\Gamma_{R,\pi}(u, q) : a = u/q\}.$$

The size function gives a description of the sizes of AFSRs. They claim that in many cases $\lambda_\pi(\mathbf{a})$ grows at most linearly with the actual size of the AFSR that is needed to generate $\mathbf{a}$.

Interpolation set [25]

To control the growth of the size of a new approximation, they assume there exists a subset of R, denoted by $P_{R,\pi}$, from which the coefficients are selected. The subset will restrict the elements that can be used to multiply the previous approximations. The subset $P_{R,\pi}$ of R must have the following properties.

There is an integer $B > 0$ such that

1. $0 \in P_{R,\pi}$, and if $s \in P_{R,\pi}$ with $\pi^B | s$, then $s = 0$;
2. for every $h_1, h_2 \in R$ and $s, t \in P_{R,\pi}$, we have

$$\psi_{R,\pi}(sh_1 + th_2) < \max(\psi_{R,\pi}(h_1)), \psi_{R,\pi}(h_2) + B;$$

3. for every $h_1, h_2 \neq 0 \in R$, there exist $s, t \in P_{R,\pi}$ such that $(s, t) \neq (0, 0)$ and $\pi^B | (sh_1 + th_2)$.

With these definition, Xu's rational approximation algorithm is given in Figure 4.1. The constant B is from the definition of $P_{R,\pi}$. It was shown that after a finite number of steps the algorithm outputs a description of the AFSR for a given eventually periodic sequence.

Theorem 4.1.1. *[25] Let* $\mathbf{a} = a_0, a_1, \cdots$ *be an eventually periodic sequence. The associated* π*-adic number* α *has the rational expression* u/q *where* $\Gamma_{R,\pi}(u,q)$ *has the minimum value,* $\lambda_\pi(\mathbf{a})$*. In Xu's algorithm for the sequence* $\mathbf{a}$*,*

1. *For every* j*,* $r_j \neq 0$*.*

2. *Suppose*

$$i > B(2b + 2c + B + c\lceil \log(B) \rceil + 2f_1) + 2B\lambda_\pi(\mathbf{a}),$$

where $f_1 = \max(\psi_{R,\pi}(a) : a \in S) \cup \psi_{R,\pi}(1)$*. Then the algorithm is convergent at* i*. That is,* $h_i/r_i = u/q$*.*

1: **procedure** XU($a_0, a_1, \cdots, a_k$)
2: $a = 1 + \pi \sum_{i=0}^{k} a_i \pi^i$
3: $(h_0, r_0) = (0, 1)$
4: Let $r_1 = b_0 + b_1\pi + \cdots + b_{B-1}\pi^{B-1}$ satisfy $r_1 a \equiv 1 \pmod{\pi^B}$
5: $h_1 = 1$
6: $m = 0$
7: **for** $i = 1$ to $k - 1$ **do**
8: **if** $(h_i - r_i a) \not\equiv 0 \pmod{\pi^{i+1}}$ **then**
9: **if** $\exists s \neq 0 \in P_{R,\pi}$ with $\pi^{i+B} | s(h_i - r_i a)$ **then**
10: $(h_{i+1}, r_{i+1}) = s(j_i, r_i)$
11: **else**
12: Find $s, t \in P_{R,\pi}$, not both zero, with $\pi^{i+B} | s(h_i - r_i a) + t\pi^{i-m}(h_m - r_m a)$
13: $(h_{i+1}, r_{i+1}) = s(j_i, r_i) + t\pi^{i-m}(h_m, r_m)$
14: **end if**
15: **if** $\Gamma_{R,\pi}(h_{i+1}, r_{i+1}) > \Gamma_{R,\pi}(h_i, r_i)$ and $\Gamma_{R,\pi}(h_i, r_i) \leq i - m + \Gamma_{R,\pi}(h_m, r_m)$ and $t \neq 0$ **then**
16: $m = i$
17: **end if**
18: **end if**
19: **end for**
20: Let $1 + \pi(u/q) = h_k/r_k$
21: Find the largest t so that π^t that divides both u and q
22: **return** $(u/\pi^t, q/\pi^t)$
23: **end procedure**

Figure 4.1: Xu's rational approximation algorithm

Theorem 4.1.2. *[25] The worst case time complexity of the Xu's rational approximation algorithm is in*

$$O(\sum_{m=1}^{\lambda_\pi(\mathbf{a})} \sigma(m)),$$

where $\sigma(m)$ is the time required to add two elements $a, b \in R$ with $\psi_{R,\pi}(a), \psi_{R,\pi}(b) \leq m$.

The worst case space complexity is in

$$O(\lambda_\pi(\mathbf{a}) \log(|S|)),$$

where $|S|$ is the cardinality of set S.

Notice that if $u, q \in R$ is the output pair of Xu's algorithm when k is large enough, then q is the connection element for an AFSR over R that outputs sequence $\mathbf{a}$. However, q may not the the smallest.

4.2 Algebraic number fields

In this section, we review some definitions and results of basic algebraic number theory. An algebraic number field K is a finite field extension of the rational numbers $\mathbb{Q}$. That is, K is a field that contains $\mathbb{Q}$ and can be considered as a vector space over $\mathbb{Q}$ of finite dimension. The dimension of this vector space is called the degree of the extension and is denoted by $[K : \mathbb{Q}]$. When $[K : \mathbb{Q}] = 2$, we say K is a quadratic extension of $\mathbb{Q}$ or K is a quadratic number field. An algebraic integer in a number field K is an element $\alpha \in K$ which is a root of a monic polynomial with coefficients in $\mathbb{Z}$.

Theorem 4.2.1. *[8] Any quadratic extension of $\mathbb{Q}$ is of the form $\mathbb{Q}(\sqrt{d})$, where d is a square free integer, not 0 or 1. The set of all algebraic integers in $\mathbb{Q}$ forms a ring*

$$E = \begin{cases} \mathbb{Z} + \mathbb{Z} \cdot \sqrt{d} & if \quad d \equiv 2 \text{ or } 3 \pmod 4, \\ \mathbb{Z} + \mathbb{Z} \cdot (\frac{1+\sqrt{d}}{2}) & if \quad d \equiv 1 \pmod 4. \end{cases}$$

The set of algebraic integers of a number field K is called the ring of integers of K. If $d > 0$, the quadratic number field K is called a real quadratic field. Otherwise, K is called an imaginary quadratic field. For any $x + y\sqrt{d} \in K$, the norm of $x + y\sqrt{d}$ is defined as

$$N(x + y\sqrt{d}) := x^2 - dy^2.$$

The norm function is multiplicative, that is,

$$N(\alpha\beta) = N(\alpha)N(\beta)$$

for all $\alpha, \beta \in \mathbb{Q}(\sqrt{d})$. We say R is norm Euclidean if for all $\alpha, \beta \in R$, $\beta \neq 0$, there exist $\epsilon, \gamma \in R$ such that $\alpha = \epsilon\beta + \gamma$ and $|N(\gamma)| < |N(\beta)|$. It is known that $\mathbb{Q}(\sqrt{d})$ is a norm Euclidean quadratic number field if and only if d is in the set

$$\{-1, -2, -3, -7, -11, 2, 3, 5, 6, 7, 11, 13, 17, 19, 21, 29, 33, 37, 41, 57, 73\}. \text{ [38]}$$

The ring of integers of a norm Euclidean quadratic number field is also norm Euclidean.

4.3 AFSR synthesis via lattice rational approximation algorithm

For Section 4.3, we discuss AFSRs over quadratic extensions of $\mathbb{Z}$. That is, fix $\pi \in \mathbb{Z}$ such that $\pi^2 = D$, where $D \in \mathbb{Z}$ is square free. So $x^2 - D$ is irreducible over the rational numbers $\mathbb{Q}$. Let $R = \mathbb{Z}[\pi]$, a quadratic extension of $\mathbb{Z}$. It is an integral domain in which every prime ideal is maximal. It can be proved that $S = \{0, 1, \ldots, D-1\}$ is a complete set of representatives for the quotient ring $R/(\pi)$. First of all, we give a different definition of the size and π-adic complexity to describe AFSRs. Then we construct a lattice based on the first k outputs of AFSRs that gives an approximation of the associated π-adic integer.

4.3.1 Size and π-adic complexity

Suppose $R = \mathbb{Z}[\pi] = \{a_0 + a_1\pi : a_0, a_1 \in \mathbb{Z}\}$ and $\pi^2 = D \in \mathbb{Z}$. To measure the size of the elements of R, let size function $\varphi_{R,\pi} : R \to \mathbb{Z}$ be

$$\varphi_{R,\pi}(q) = q_0^2 + q_1^2,$$

where $q = q_0 + q_1\pi$ and $q_0, q_1 \in \mathbb{Z}$.

Proposition 4.3.1. *For any $u, q \in \mathbb{Z}[\pi]$, we have*

1. $\varphi_{R,\pi}(u \pm q) \le 2(\varphi_{R,\pi}(u) + \varphi_{R,\pi}(q))$ *and*
2. $\varphi_{R,\pi}(uq) \le (D^2 + |1 + D|/2)\varphi_{R,\pi}(u)\varphi_{R,\pi}(q)$.

Proof: Let $u = u_0 + u_1\pi$ and $q = q_0 + q_1\pi$ where $u_0, u_1, q_0, q_1 \in \mathbb{Z}$. We have $u \pm q = (u_0 \pm q_0) + (u_1 \pm q_1)\pi$, so

$$\begin{aligned}
\varphi_{R,\pi}(u \pm q) &= (u_1 \pm q_1)^2 + (u_0 \pm q_0)^2 \\
&= u_1^2 + q_1^2 + u_0^2 + q_0^2 \pm 2u_1q_1 \pm 2u_0q_0 \\
&\le 2(u_1^2 + q_1^2 + u_0^2 + q_0^2) \\
&= 2(\varphi_{R,\pi}(u) + \varphi_{R,\pi}(q)).
\end{aligned}$$

We have $uq = (u_0 + u_1\pi)(q_0 + q_1\pi) = (u_0q_0 + Du_1q_1) + (u_0q_1 + u_1q_0)\pi$, so

$$\begin{aligned}
\varphi_{R,\pi}(uq) &= (u_0q_0 + Du_1q_1)^2 + (u_0q_1 + u_1q_0)^2 \\
&= u_0^2q_0^2 + D^2u_1^2q_1^2 + u_0^2q_1^2 + u_1^2q_0^2 + (2 + 2D)u_0u_1q_0q_1 \\
&\le u_0^2q_0^2 + D^2u_1^2q_1^2 + u_0^2q_1^2 + u_1^2q_0^2 + |(2 + 2D)| \cdot |u_0u_1q_0q_1|
\end{aligned}$$

Since $\varphi_{R,\pi}(u)\varphi_{R,\pi}(q) = u_0^2q_0^2 + u_1^2q_1^2 + u_0^2q_1^2 + u_1^2q_0^2 \geq 4|u_0u_1q_0q_1|$, we have

$$\begin{aligned}\varphi_{R,\pi}(uq) &\leq D^2\varphi_{R,\pi}(u)\varphi_{R,\pi}(q) + \frac{|2+2D|}{4}\varphi_{R,\pi}(u)\varphi_{R,\pi}(q)\\ &= (D^2 + \frac{|1+D|}{2})\varphi_{R,\pi}(u)\varphi_{R,\pi}(q).\end{aligned}$$

□

For any $u, q \in R$, let

$$\Phi_{R,\pi}(u,q) = \log_{|D|}(\varphi_{R,\pi}(u) + \varphi_{R,\pi}(q)).$$

We define $\Phi_{R,\pi}(u,q)$ to be the *size* of the AFSR constructed by Theorem 1.3.1. That is, u/q is a rational expression of α, the associated π-adic integer of sequence $\mathbf{a}$. Then the *π-adic complexity* of $\mathbf{a}$ is

$$\varphi_\pi(\mathbf{a}) = \min\{\Phi_{R,\pi}(u,q) : \alpha = u/q\}.$$

The AFSR synthesis problem in terms of the size and π-adic complexity defined above is as follow:

- **Given** A prefix of the eventually periodic sequence $\mathbf{a}$=$a_0, a_1, \cdots$ over $S = \{0, 1, \cdots, |D|-1\}$.
- **Find** $u, q \in R$ satisfying $\alpha = u/q$ and minimizing $\Phi_{R,\pi}(u,q)$.

4.3.2 k-th Approximation Lattices

Definition 4.3.1. *Let $\pi = \sqrt{D}$, where $D \in \mathbb{Z}$ is square free. Let $R = \mathbb{Z}[\pi]$ and let R_π be the ring of π-adic integers. Suppose $\alpha = a_0 + a_1\pi + a_2\pi^2 + \ldots$ is an element in R_π. The* kth approximation lattice *of α is defined as*

$$L_k = L_k(\alpha) := \{(u_1,u_2,u_3,u_4) \in \mathbb{Z}^4 : \alpha(u_3+u_4\pi) - (u_1+u_2\pi) \equiv 0 \pmod{\pi^k}\}$$

Notice that for every element (u_1,u_2,u_3,u_4) in $L_k(\alpha)$, we have

$$\alpha \equiv \frac{u_1+u_2\pi}{u_3+u_4\pi} \pmod{\pi^k} \quad \text{if} \quad \gcd(u_3, D) = 1.$$

Thus the pair (u,q) with $u = u_1 + u_2\pi$ and $q = u_3 + u_4\pi$ represents a fraction u/q whose π-adic expansion agrees with α in the first k places. We call (u,q) a rational

approximation of α up to k terms. If $\alpha_k = \sum_{i=0}^{k-1} a_i \pi^i = a + b\pi$, where $a, b \in \mathbb{Z}$, then $\mathbf{u}_1 = (a, b, 1, 0) \in L_k$. Also, it can be verified that $\mathbf{u}_2 = (Db, a, 0, 1) \in L_k$. Suppose

$$\pi^k = c + d\pi = \begin{cases} D^{\frac{k-1}{2}}\pi, & \text{if k is odd;} \\ D^{\frac{k}{2}}, & \text{if k is even.} \end{cases}$$

Then $\mathbf{u}_3 = (c, d, 0, 0) \in L_k$ and $\mathbf{u}_4 = (Dd, c, 0, 0) \in L_k$

Theorem 4.3.1. *$L_k(\alpha)$ is a four dimensional lattice and $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$ is a basis of $L_k(\alpha)$. L_{i+1} is a sublattice of L_i for any $i \in \mathbb{Z}$.*

Proof: If $\mathbf{u} = (u_1, u_2, u_3, u_4) \in L_k$ and $\mathbf{v} = (v_1, v_2, v_3, v_4) \in L_k$, then $\mathbf{u} + \mathbf{v} \in L$. So L_k is a lattice. The four vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ are linearly independent elements of L_k. Now suppose that $\mathbf{x} = (x_1, x_2, x_3, x_4)$ is an arbitrary vector in L_k. So $\alpha_k(x_3 + x_4\pi) - (x_1 + x_2\pi) = \gamma\pi^k$ for some $\gamma = r_1 + r_2\pi \in R$. Making corresponding terms equal, we have

$$\begin{cases} ax_3 + bx_4 D - x_1 = r_1 c + r_2 dD \\ bx_3 + ax_4 - x_2 = r_2 c + r_1 d. \end{cases}$$

This also means that $\mathbf{x} = x_3\mathbf{u}_1 + x_4\mathbf{u}_2 - r_1\mathbf{u}_3 - r_2\mathbf{u}_4$. So $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$ is a basis of L_k.

For any $(y_1, y_2, y_3, y_4) \in L_{i+1}$ and any $i \in \mathbb{Z}$ we have $\alpha(y_3 + y_4\pi) - (y_1 + y_2\pi) \equiv 0 \pmod{\pi^{i+1}}$. So $\alpha(y_3 + y_4\pi) - (y_1 + y_2\pi) \equiv 0 \pmod{\pi^i}$. That is, $(y_1, y_2, y_3, y_4) \in L_i$. So L_{i+1} is a sublattice of L_i for any $i \in \mathbb{Z}$. □

4.3.3 Lattice Approximation Algorithms

The approximation algorithm based on the lattice reduction greedy algorithm GreedyLatticeReduction (Figure 1.9) is given in Figure 4.2. Let $\mathbf{a}$ be a sequence with associated π-adic integer α. Given a sufficiently large prefix of $\mathbf{a}$, this algorithm finds the rational expression of α that realizes the π-adic complexity of $\mathbf{a}$. With the help of GreedyLatticeReduction, we can find the shortest vector of the kth approximation lattice which gives the best rational approximation of α up to k terms. Suppose the π-adic complexity is known. Theorem 4.3.2 shows that if k is chosen big enough, then such a rational approximation is exactly the rational expression we want. The algorithm shown in Figure 4.2 is just for the case when k is even. The odd case is similar, so details are omitted here.

Theorem 4.3.2. *Let $\mathbf{a}$ be a π-adic sequence with associated π-adic integer α. Suppose the size of the AFSR that generates $\mathbf{a}$ is less than or equal to n. That is, the π-adic complexity of $\mathbf{a}$, $\varphi_\pi(\mathbf{a})$, is less than or equal to n. Let* ApproxLattice *(Figure 4.2)*

1: **procedure** ApproxLattice($a_0, a_1, \ldots, a_{k-1}$)
2: **Input:** first k terms of sequence **a**
3: **Output:** $u, q \in R$ satisfying $\alpha = u/q$ and minimizing $\Phi_{R,\pi}(x, y)$
4: $a := \sum_{0 \le i \le k/2} a_{2i} D^i$
5: $b := \sum_{0 \le i \le (k-2)/2} a_{2i+1} D^i$
6: $c := D^{k/2}$
7: $\mathbf{u}_1 := (a, b, 1, 0)$
8: $\mathbf{u}_2 := (Db, a, 0, 1)$
9: $\mathbf{u}_3 := (c, 0, 0, 0)$
10: $\mathbf{u}_4 := (0, c, 0, 0)$
11: Sort $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ by their norm $\|\cdot\|$. Let $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$ be ordered.
12: Compute the Gram matrix G so that $G_{ij} = \langle \mathbf{u}_i, \mathbf{u}_j \rangle$.
13: $(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$:=GreedyLatticeReduction$(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$
14: Suppose $\mathbf{u}_1 = (u_0, u_1, q_0, q_1)$
15: **return** $(u_0 + u_1\pi, q_0 + q_1\pi)$
16: **end procedure**

Figure 4.2: Lattice Rational Approximation Algorithm for AFSRs over a quadratic extension

be executed with $k \ge 2n+2+\lceil \log_{|D|}(4D^2+2|1+D|)\rceil$. *Suppose the algorithm outputs a pair* (u, q) *of elements of* R. *Then*

$$\alpha = \sum_{i=0}^{\infty} a_i \pi^i = \frac{u}{q}.$$

Proof: Let u'/q' be a rational expression of α with $\Phi_{R,\pi}(u', q') = \varphi_\pi(\mathbf{a})$. That is

$$\alpha = \sum_{i=0}^{\infty} a_i \pi^i = \frac{u'}{q'}.$$

It follows that $\Phi_{R,\pi}(u', q') \le n$. Suppose $\mathbf{v}_1 = (v_1, v_2, v_3, v_4)$ where $u' = v_1 + v_2\pi$ and $q' = v_3 + v_4\pi$. So $\mathbf{v}_1 \in L_k(a)$.

Let (u, q) be the output of ApproxLattice. Then Theorem 1.5.1 shows that $\mathbf{u}_1 = (u_1, u_2, u_3, u_4)$ in step 14 is the minimal vector in $L_k(\alpha)$.
We have $u = u_1 + u_2\pi$ and $q = u_3 + u_4\pi$. So

$$\|\mathbf{u}_1\| = \sqrt{u_1^2 + u_2^2 + u_3^2 + u_4^2} \le \sqrt{v_1^2 + v_2^2 + v_3^2 + v_4^2} = \|\mathbf{v}_1\|.$$

So

$$\begin{aligned}\Phi_{R,\pi}(u,q) &= \log_{|D|}(u_1^2+u_2^2+u_3^2+u_4^2)\\ &\le \log_{|D|}(v_1^2+v_2^2+v_3^2+v_4^2)\\ &= \Phi_{R,\pi}(u',q') \le n.\end{aligned}$$

This shows that $\varphi_{R,\pi}(u'), \varphi_{R,\pi}(q'), \varphi_{R,\pi}(u), \varphi_{R,\pi}(q)$ are all less than or equal to $|D|^n$. We have

$$\frac{u}{q} \equiv \frac{u'}{q'} \pmod{\pi^k},$$

so

$$\pi^k \Big| \frac{uq'-u'q}{qq'}.$$

Thus there exists $t \in R$ such that $tqq'\pi^k = uq' - u'q$. From Proposition 4.3.1,

$$\begin{aligned}\varphi_{R,\pi}(uq'-u'q) &\le 2(\varphi_{R,\pi}(uq') + \varphi_{R,\pi}(u'q))\\ &\le (2D^2+|1+D|)(\varphi_{R,\pi}(u)\varphi_{R,\pi}(q') + \varphi_{R,\pi}(u')\varphi_{R,\pi}(q'))\\ &\le (4D^2+2|1+D|)|D|^{2n}.\end{aligned}$$

For any $e = e_1 + e_2\pi \neq 0 \in \mathbb{Z}[\pi]$, we have

$$e\pi^k = \begin{cases} e_1 D^{\frac{k}{2}} + e_2 D^{\frac{k}{2}}\pi & \text{if } k \text{ is even} \\ e_2 D^{\frac{k+1}{2}} + e_1 D^{\frac{k-1}{2}}\pi & \text{if } k \text{ is odd.}\end{cases}$$

Therefore $\varphi_{R,\pi}(e\pi^k) > |D|^{k-2}$. This is to say, $\varphi_{R,\pi}(tqq'\pi^k) > |D|^{k-2}$ if $t \neq 0$. But from $k \ge 2n+2+\lceil \log_{|D|}(4D^2+2|1+D|)\rceil$ we have $|D|^{k-2} \ge (4D^2+2|1+D|)|D|^{2n}$. So t must be 0, which also means $uq' - u'q = 0$. This proves that

$$\frac{u}{q} = \frac{u'}{q'} = \sum_{i=0}^{\infty} a_i \pi^i.$$

From the proof we also know that $\Phi_{R,\pi}(u,q)$ reaches the π-adic complexity of sequence $\mathbf{a}$ which means that we find the smallest AFSR that generates $\mathbf{a}$. □

Theorem 4.3.3. *The Lattice Rational Approximation Algorithm,* ApproxLattice, *runs in time $O(k^2)$ if k elements of $\mathbf{a}$ are used.*

Proof: The time complexity of getting $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4$ from step 4 to step 10 in Figure

(4.2) is $O(k \log k)$. Since

$$\begin{aligned} |a| &= \left| \sum_{0 \le i \le k/2} a_{2i} D^i \right| \le |D|^{k/2+1}, \\ |b| &= \left| \sum_{0 \le i \le (k-2)/2} a_{2i} D^i \right| \le |D|^{k/2}, \text{ and} \\ |c| &\le |D|^{k/2}, \end{aligned}$$

we have $\max(\|\mathbf{u}_1\|, \|\mathbf{u}_2\|, \|\mathbf{u}_3\|, \|\mathbf{u}_4\|) \le \sqrt{2}|D|^{(k+3)/2}$.

In step 11, to compute and sort $\|\mathbf{u}_1\|, \|\mathbf{u}_2\|, \|\mathbf{u}_3\|, \|\mathbf{u}_4\|$ takes time $O(k^2)$ because the dimension of L_k is fixed. Also, the time complexity for computing the Gram matrix G is $O(k^2)$.

The most costly step in ApproxLattice is Step 13 that calls GreedyLatticeReduction. According to Theorem (1.5.1), the time complexity is bounded by $O\left(\log(\sqrt{2}|D|^{\frac{k+3}{2}})[1 + \log(\sqrt{2}|D|^{\frac{k+3}{2}}) - \log \zeta_1(L_k)]\right) = O(k^2)$, where $\zeta_1(L)$ is the smallest vector in L_k. To sum up, the time complexity of ApproxLattice is $O(k^2)$. □

4.4 AFSR synthesis via the Extended Euclidean Rational Approximation Algorithm

In this section, we want to apply the Extended Euclidean algorithm, so we require R to be the ring of integers of $\mathbb{Q}(\sqrt{d})$, where $\mathbb{Q}(\sqrt{d})$ is the imaginary norm Euclidean quadratic field and $d \neq -1$. For other cases, the algorithm will not work. That is,

$$R = \begin{cases} \mathbb{Z} + \mathbb{Z} \cdot \sqrt{d} & if \quad d = -2, \\ \mathbb{Z} + \mathbb{Z} \cdot (\frac{1+\sqrt{d}}{2}) & if \quad d = -3, -7, \text{ or } -11. \end{cases}$$

It is known that R is a Euclidean domain with respect to the norm function. For any $x + y\sqrt{d} \in R$, we have $N(x + y\sqrt{d}) = x^2 - dy^2 \ge 0$. Let $\pi = \sqrt{d} \in R$. Then $N(\pi) = -d = |d|$. The ring R_π consists of elements $\alpha = a_0 + a_1\pi + \ldots$ with coefficients $a_i \in S = \{0, 1, \ldots, |d| - 1\}$.

An element $\mu \in R$ is a unit if and only if $N(\mu) = \pm 1$. When $d = -2, -7$, and -11, an element $\mu \in R$ is a unit if and only if $\mu = \pm 1$. When $d = -3$, an element $\mu \in R$ is a unit if and only if $\mu = (\frac{1+\sqrt{-3}}{2})^i$, for $0 \le i \le 5$.

Definition 4.4.1. *Let* $\boldsymbol{a}$ *be an eventually periodic sequence over* S*. If its associated* π*-adic integer* α *has a rational expression* u/q*, then the size of the corresponding*

AFSR that can generate $\boldsymbol{a}$ *is defined as follows:*

$$\Psi_{R,\pi}(u,q) = \log_{|d|}(\max\{|N(u)|, |N(q)|\})$$

The π*-adic complexity of the sequence* $\mathbf{a}$*, denoted by* $\phi_\pi(\alpha)$*, is the minimum of* $\Psi_{R,\pi}(u,q)$ *over all* u, q *with* $\alpha = u/q$*.*

Based on the Definition 4.4.1, the AFSR synthesis problem can be rephrased as follows:

- **Given** A prefix of the eventually periodic sequence $\mathbf{a}$=$a_0, a_1, \cdots$ over $S = \{0, 1, \ldots, |d| - 1\}$.
- **Find** $u, q \in R$ satisfying $\alpha = u/q$ and minimizing $\phi_\pi(\mathbf{a})$.

4.4.1 R-lattices

Definition 4.4.2. *An* R*-lattice of rank* k *is a subset* $L \subseteq \mathbb{C}^n$ *of the form*

$$L = \bigoplus_{i=1}^{k} R\vec{u}_i,$$

where $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k \in \mathbb{C}^n$ *are linearly independent vectors over* $\mathbb{C}$*. That is, it is a finitely generated free* R*-submodule in* $\mathbb{C}^n$*. We say* $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ *is a basis of* L*.* L *is full if* $k = n$*. We treat all vectors in* $\mathbb{C}^n$ *as column vectors.*

Definition 4.4.3. *Let* $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$ *be* k *linearly independent vectors over* R*. The matrix whose columns are* $\vec{u}_i$ *is denoted by* $[\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k]$*. Let* L *be a full* R*-lattice with basis* $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n$*. The volume of* L *is defined as*

$$\mathrm{vol}(L) = N(\det([\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n])).$$

Definition 4.4.4. $GL_n(R)$ *is the group of* $n \times n$ *matrices over* R *whose determinant is a unit in* R*.*

Lemma 4.4.1. *Let* L *be an* R*-lattice with basis* $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k$*. The vectors* $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k \in \mathbb{C}^n$ *form a basis of* L *if and only if there exists a matrix* $T \in GL_k(R)$ *such that*

$$[\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k] = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k] \cdot T.$$

Proof. We prove both directions:
"⇒" Since $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k \in L$, there is a $k \times k$ matrix T over R with

$$[\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k] = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k] \cdot T.$$

The vectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$ are linearly independent, so $\det(T) \neq 0$. It follows that

$$[\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k] \cdot T^{-1} = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k].$$

Since $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$ form a basis, T^{-1} has entries in R. $\det(T)$ and $\det(T^{-1})$ are both in R, so $\det(T)$ is invertible in R. That is to say $\det(T)$ is a unit.
"$\Leftarrow$" Let us suppose for some $T \in GL_k(R)$ that

$$[\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k] = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k] \cdot T.$$

It follows that $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k \in L$ and they are linearly independent. Suppose L' is the R-lattice with basis $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$. Similarly, we have

$$[\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k] \cdot T^{-1} = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k].$$

Then $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k \in L'$. Thus,

$$L = L'$$

■

The following corollary can be derived from the proof of Lemma 4.4.1.

Corollary 4.4.1. *Let L be a full R-lattice with basis $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n$. Suppose $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n$ are n linearly independent vectors in L. Then,*

$$\mathrm{vol}(L) \big| N(\det([\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n]))$$

Theorem 4.4.1. *Let L be a full R-lattice in $\mathbb{C}^n$. The volume $\mathrm{vol}(L)$ is independent of the choice of basis. Let $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$ be any k linearly independent vectors in L. The $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$ form a basis if and only if*

$$N(\det([\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k])) = \mathrm{vol}(L)$$

Proof. This follows directly from Lemma 4.4.1. ■

Definition 4.4.5. *Let L_1, L_2 be R-lattices of the same rank with $L_1 \subseteq L_2$. Then we say L_1 is a sub-lattice of L_2.*

Theorem 4.4.2. *Let L_1 be a sub-lattice of L_2, Then*

$$\mathrm{vol}(L_2) \big| \mathrm{vol}(L_1).$$

Proof. This is a direct result from Corollary 4.4.1. ■

Definition 4.4.6. *Let R be the ring of integers of $\mathbb{Q}(\sqrt{d})$ and $\pi^2 = d$ for $d = -2, -3, -7,$ or -11. Suppose $\alpha = \sum_{i=0}^{\infty} a_i \pi^i$ is a π-adic integer with $a_i \in S$ where $S = \{0, 1, \cdots, |d| - 1\}$ is the complete set of representatives of $R/(\pi)$. The kth approximation R-lattice of α is defined as*

$$L_k = L_k(\alpha) := \{(\mu_1, \mu_2) \in R \times R : \alpha\mu_2 - \mu_1 \equiv 0 \pmod{\pi^k}\}.$$

Consider the sequence $\mathbf{a}$ over $R/(\pi)$ which is associated with the π-adic integer α, that is, $\mathbf{a} = (a_0, a_1, a_2, a_3 \cdots)$. For every element $(\mu_1, \mu_2) \in L_k(\alpha)$, we have $\mu_1/\mu_2 \equiv \alpha \pmod{\pi^k}$, if μ_2 is coprime with π. In this case, μ_1/μ_2 is a rational approximation of $\mathbf{a}$ up to k terms. We see in Theorem 4.4.7 that when (μ_1, μ_2) is the output from the Extended Euclidean Rational Approximation Algorithm, then μ_2 is coprime with π.

Theorem 4.4.3. *Let L_k be the kth approximation R-lattice of $\alpha = \sum_{i=0}^{\infty} a_i \pi^i$. Then L_k is a full R-lattice in $\mathbb{C}^2$. $\mathrm{vol}(L_k) = N(\pi)^k$ and L_{i+1} is a sublattice of L_i for any $i \in \mathbb{Z}^+$.*

Proof. Let $A_k = a_0 + a_1\pi + a_2\pi^2 + \cdots + a_{k-1}\pi^{k-1} \in R$, so we have $(\pi^k, 0), (A_k, 1) \in L_k$. They are linearly independent, so L_k is full.

For any $(\mu_1, \mu_2) \in L_k$, we have $A_k\mu_2 - \mu_1 = \gamma\pi^k$ for some $\gamma \in R$. So

$$(\mu_1, \mu_2) = \mu_2(A_k, 1) - \gamma(\pi^k, 0)$$

It follows that $(\pi^k, 0), (A_k, 1)$ form a basis of L_k. We have

$$\mathrm{vol}(L_k) = N\big(\det \begin{pmatrix} \pi^k & A_k \\ 0 & 1 \end{pmatrix}\big) = N(\pi)^k = |d|^k.$$

For any $(\mu_1, \mu_2) \in L_{i+1}$ and any $i \in \mathbb{Z}$ we have,

$$a\mu_2 - \mu_1 \equiv 0 \pmod{\pi^{i+1}}.$$

So

$$a\mu_2 - \mu_1 \equiv 0 \pmod{\pi^i}.$$

That is, $(\mu_1, \mu_2) \in L_i$. So L_{i+1} is a sublattice of L_i for any $i \in \mathbb{Z}$.

■

Definition 4.4.7. *Let $\vec{u} = (\mu_1, \mu_2, \ldots, \mu_n)$ be a vector in an R-lattice L. Define a mapping $\Omega : \mathbb{C}^n \to \mathbb{R}$ by:*

$$\Omega(\vec{u}) = \max_{i=1,2,\ldots,n} \{|N(\mu_i)|\}.$$

More specifically, if $\vec{u} = (\mu_1, \mu_2) \in L_k \subset \mathbb{C}^2$, then $\Omega(\vec{u}) = \max\{|N(\mu_1)|, |N(\mu_2)|\}$.

Theorem 4.4.4. *$\Omega(\cdot)$ is a norm of L.*

Proof. For any $\mu, \nu \in R$, we have

$$N(\mu + \nu) \leq N(\mu) + N(\nu) \text{ and } N(\mu\nu) = N(\mu)N(\nu).$$

So it can be seen that for any $\vec{u}, \vec{v} \in L$ and $\gamma \in R$:

- $\Omega(\vec{u} \pm \vec{v}) \leq \Omega(\vec{u}) + \Omega(\vec{v})$
- $\Omega(\gamma\vec{u}) = N(\gamma)\Omega(\vec{u})$
- $\Omega(\vec{u}) = 0$ if and only if $\vec{u} = (0, 0)$.

■

Lemma 4.4.2. *Let L be a full R-lattice in $\mathbb{C}^2$. Let $\vec{u} \in L$ be a minimal nonzero vector, that is, $\Omega(\vec{u}) \leq \Omega(\vec{u}')$ for all $\vec{u}' \in L - \{(0, 0)\}$. Then there is a vector $\vec{w} \in L$ so that $\vec{u}$ and $\vec{w}$ form a basis of L.*

Proof. Let $\vec{m}, \vec{n}$ be a basis of L. Then there exist $\gamma_1, \gamma_2 \in R$, so that

$$\vec{u} = \gamma_1\vec{m} + \gamma_2\vec{n}.$$

It follows from the minimality of $\vec{u}$ that γ_1, γ_2 are coprime. So $N(\gcd(\gamma_1, \gamma_2)) = 1$. Otherwise, we let

$$\vec{u}' = \frac{\gamma_1}{\gcd(\gamma_1, \gamma_2)}\vec{m} + \frac{\gamma_2}{\gcd(\gamma_1, \gamma_2)}\vec{n}.$$

If $N(\gcd(\gamma_1, \gamma_2)) \neq 1$, then $\vec{u}' \in R$ and $\Omega(\vec{u}') = \Omega(\vec{u})/N(\gcd(\gamma_1, \gamma_2)) < \Omega(\vec{u})$.
R is an Euclidean domain, so there exist $\gamma_3, \gamma_4 \in R$ such that

$$\gamma_1\gamma_4 + \gamma_2\gamma_3 = \gcd(\gamma_1, \gamma_2).$$

Let $\vec{w} = \gamma_3\vec{m} - \gamma_4\vec{n}$, so

$$[\vec{u}, \vec{w}] = [\vec{m}, \vec{n}] \cdot \begin{pmatrix} \gamma_1 & \gamma_3 \\ \gamma_2 & -\gamma_4 \end{pmatrix}$$

and

$$N\big(\det \begin{pmatrix} \gamma_1 & \gamma_3 \\ \gamma_2 & -\gamma_4 \end{pmatrix}\big) = N(\gamma_1\gamma_4 + \gamma_2\gamma_3) = 1.$$

It follows that $\vec{u}, \vec{w}$ form a basis of L.

■

Lemma 4.4.3. *Suppose that $\alpha = a_0 + a_1\pi + a_2\pi^2 + \cdots$ is a π-adic integer with $\pi^2 = d$, and let $L_k(\alpha)$ be its kth approximation R-lattice. Let $\vec{u} = (\mu_1, \mu_2)$ and $\vec{v} = (\nu_1, \nu_2)$ be two linearly independent vectors in $L_k(\alpha)$ such that $\Omega(\vec{u}) < x$ and $\Omega(\vec{v}) < y$ for some $x, y \in \mathbb{Z}$. Then $xy > |d|^k/2$.*

Proof. Since $\vec{u}, \vec{v}$ are linearly independent vectors in $L_k(a)$,

$$\mathrm{vol}(L_k(a)) \big| N(\det([\vec{u}, \vec{v}])).$$

We also have $\det([\vec{u}, \vec{v}]) \neq 0$, so $N(\det([\vec{u}, \vec{v}])) \geq |d|^k$. That is,

$$N(\mu_1\nu_2 - \mu_2\nu_1) \geq |d|^k.$$

But

$$N(\mu_1\nu_2 - \mu_2\nu_1) \leq N(\mu_1)N(\nu_2) + N(\mu_2)N(\nu_1) < 2xy.$$

So $xy > |d|^k/2$. ∎

Definition 4.4.8. *We say (μ, ν) is a best kth-approximation if (μ, ν) is a minimal vector in L_k with respect to norm $\Omega(\cdot)$.*

Theorem 4.4.5. *Suppose that a sequence $\mathbf{a} = a_0, a_1, \ldots$ over $R/(\pi)$ is generated by an AFSR and $\mathbf{a}$ is identified with a π-adic number $\alpha = \sum_{i=0}^{\infty} a_i\pi^i$. Let ρ/χ be a rational approximation to at least k terms. That is, $(\rho, \chi) \in L_k(\alpha)$. Let $m = \sqrt{|d|^k/2}$. If $\Omega\big((\rho, \chi)\big) < m$, then $(\rho, \chi) = \gamma(\mu, \nu)$ for some $\gamma \in R$, where (μ, ν) is a best kth-approximation of $\mathbf{a}$.*

Proof. Since (μ, ν) is a minimal vector with respect to the norm Ω, we have

$$\Omega\big((\mu, \nu)\big) \leq \Omega\big((\rho, \chi)\big) < m.$$

We have

$$m^2 = \frac{|d|^k}{2}.$$

It follows from Lemma 4.4.3 that $(\mu, \nu), (\rho, \chi)$ are not linearly independent. Thus there is $\gamma \in \mathbb{C}$ such that $(\rho, \chi) = \gamma(\mu, \nu)$. By Lemma 4.4.2, (μ, ν) is an element of a basis for $L_k(a)$, so $\gamma \in R$.

∎

Corollary 4.4.2. *If (μ, ν) is a best kth-approximation, then (μ, ν) is unique up to a unit. That is, if $\Omega((\mu', \nu')) = \Omega((\mu, \nu))$, then $(\mu', \nu') = \gamma'(\mu, \nu)$ for some unit $\gamma' \in R$.*

Proof. If (μ', ν') is a k-th approximation of $\mathbf{a}$ with $\Omega((\mu', \nu')) = \Omega((\mu, \nu)) < m$, then we have for some $\gamma' \in R$,

$$(\mu', \nu') = \gamma'(\mu, \nu).$$

So $N(\gamma') = 1$, which means that γ' is a unit.

■

4.4.2 Division Algorithm in R

In this section, we give a rational approximation algorithm based on the extended Euclidean algorithm. It works when R is the ring of integers of $\mathbb{Q}(\sqrt{d})$ with $d = -2, -3, -7$, or -11. For other values of d, the algorithm may not output the exact rational expression of a sequence.

It is well known that R is a Euclidean domain when R is the ring of integers of $\mathbb{Q}(\sqrt{d})$ with $d = -2, -3, -7$, or -11. That is, for any elements ϵ and β in R, and $\beta \neq 0$ there are ξ and γ in R such that

$$\epsilon = \xi\beta + \gamma,$$

and $N(\gamma) < N(\beta)$.

- When $d = -2$, we can find ξ and γ by the following steps. We have $\epsilon/\beta = e + f\pi$, for some $e, f \in \mathbb{Q}$. Pick $g, h \in \mathbb{Z}$ such that

  $$|e - g| \leq 1/2, \text{ and } |f - h| \leq 1/2.$$

  Let $\xi = g + h\pi$. Then

  $$\gamma = \epsilon - \xi\beta = \beta\big((e - g) + (f - h)\pi\big).$$

  So we have $N(\gamma) = N(\beta)N\big((e - g) + (f - h)\pi\big) \leq 3/4N(\beta) < N(\beta)$.

- When $d = -3, -7$ or -11, for any element ϵ in R we have

  $$\epsilon = a + b(\frac{1 + \sqrt{d}}{2}) = (a + \frac{b}{2}) + \frac{b}{2}\sqrt{d}, \quad a, b \in \mathbb{Z}.$$

  Suppose $\epsilon/\beta = e + f\sqrt{d} \in \mathbb{Q}(\sqrt{d})$, for some $e, f \in \mathbb{Q}$. Pick $h \in \mathbb{Z}$ such that

  $$|f - h/2| \leq 1/4.$$

Then pick $g \in \mathbb{Z}$ such that

$$|e - h/2 - g| \leq 1/2.$$

Let $\xi = (g + \frac{h}{2}) + \frac{h}{2}\sqrt{d}$. Then

$$\gamma = \epsilon - \xi\beta = \beta\big((e - g - h/2) + (f - h/2)\sqrt{d}\big).$$

So we have $N(\gamma) = N(\beta)N\big((e-g-h/2)+(f-h/2)\sqrt{d}\big) \leq (1/4-d/16)N(\beta) < N(\beta)$.

4.4.3 The Extended Euclidean Rational Approximation Algorithm

Let ϵ, β be two elements in R. The extended Euclidean algorithm computes the greatest common divisor and the associated Bézout coefficients of ϵ and β as follows:

$$\begin{aligned}(\gamma_0, \rho_0, \chi_0) &= (\epsilon, 1, 0)\\ (\gamma_1, \rho_1, \chi_1) &= (\beta, 0, 1).\end{aligned}$$

For $i \geq 1$,

$$\begin{aligned}\gamma_{i+1} &= \gamma_{i-1} - \xi_i\gamma_i\\ \rho_{i+1} &= \rho_{i-1} - \xi_i\rho_i\\ \chi_{i+1} &= \chi_{i-1} - \xi_i\chi_i,\end{aligned}$$

where $N(\gamma_{i+1}) < N(\gamma_i)$ using the procedure mentioned in Section 4.4.2. The computation stops at $N(\gamma_t) = 0$ for some $t \in \mathbb{N}$. The element γ_{t-1} is the greatest common divisor of ϵ and β and $(\rho_i, \chi_i, \gamma_i)_{0\leq i\leq t}$ is called the Bézout sequence of ϵ and β.

Theorem 4.4.6. *[68] Let $(\rho_i, \chi_i, \gamma_i)_{0\leq i\leq t}$ be the Bézout sequence of ϵ and β. We have the following properties:*

1. *$\rho_i\epsilon + \chi_i\beta = \gamma_i$, for all $i \in \{0, \ldots, t\}$.*
2. *$\rho_i\chi_{i+1} - \rho_{i+1}\chi_i = (-1)^i$, for all $i \in \{0, \ldots, t-1\}$.*

Let $\mathbf{a}$ be an eventually periodic sequence over $R/(\pi)$ and let $\alpha = \sum_{i=0}^{\infty} a_i\pi$ be the π-adic integer associated with sequence $\mathbf{a}$. Suppose the first k symbols $a_0, a_1, \ldots, a_{k-1}$ are available. We execute the extended Euclidean algorithm with $\epsilon = \pi^k$ and $\beta = \sum_{i=0}^{k-1} a_i\pi^i$. Then,

$$\gamma_i = \rho_i\epsilon + \chi_i\beta.$$

That is,

$$\chi_i\beta - \gamma_i \equiv 0 \pmod{\pi^k},$$

so $(\gamma_i, \chi_i) \in L_k(\alpha)$. The algorithm is given in Figure 4.3. Note that EEAapprox stops when $N(\gamma_i) \leq |d|^{k/2}$ which is different from the Euclidean algorithm (line 4, Figure 4.3).

```
1:  procedure EEAApprox(a_0, ..., a_{k-1})
2:      (γ_0, ρ_0, χ_0) = (π^k, 1, 0)
3:      (γ_1, ρ_1, χ_1) = (Σ_{i=0}^{k-1} a_i π^i, 0, 1)
4:      while N(γ_1) > |d|^{k/2} do
5:          Let γ_0 = ξγ_1 + γ_2 with N(γ_2) < N(γ_1)
6:          (ρ_2, χ_2) = (ρ_0 − ξρ_1, χ_0 − ξχ_1)
7:          (γ_0, ρ_0, χ_0) = (γ_1, ρ_1, χ_1)
8:          (γ_1, ρ_1, χ_1) = (γ_2, ρ_2, χ_2)
9:      end while
10:     if max{|N(γ_1)|, |N(χ_1)|} < sqrt(|d|^k/2) then
11:         return (γ_1, χ_1)
12:     else
13:         return FALSE
14:     end if
15: end procedure
```

Figure 4.3: The Extended Euclidean Rational Approximation Algorithm

Theorem 4.4.7. *Suppose the size of the AFSR that generates the π-adic sequence* $\mathbf{a}$ *is less than or equal to n. That is, the π-adic complexity of* $\mathbf{a}$, $\phi_\pi(\mathbf{a})$, *is less than or equal to n. Let the Extended Euclidean Rational Approximation Algorithm be executed with inputs $(a_0, ..., a_{k-1})$ and $k > 2n+1$. It outputs a pair (γ_1, χ_1) of elements of R. Then χ_1 is coprime with π and*

$$\alpha = \sum_{i=0}^{\infty} a_i\pi^i = \frac{\gamma_1}{\chi_1}.$$

Proof. Let μ/ν be a best approximation of sequence $\mathbf{a}$. That is $\gcd(\mu, \nu) = 1$, $\gcd(\pi, \nu) = 1$, and

$$\alpha = \sum_{i=0}^{\infty} a_i\pi^i = \frac{\mu}{\nu}.$$

We have $\Omega((\mu, \nu)) \leq |d|^n$, because

$$\phi_\pi(\mathbf{a}) = \log_{|d|}(\max\{|N(\mu)|, |N(\nu)|\}) \leq n.$$

Let (σ, τ) be a minimal vector in $L_k(a)$. Then

$$\Omega((\sigma, \tau)) \leq \Omega((\mu, \nu)) \leq |d|^n.$$

We have

$$\frac{\sigma}{\tau} = \frac{\mu}{\nu} \pmod{\pi^k}.$$

Thus $\sigma\nu - \mu\tau = \pi^k \delta \tau \nu$, for some $\delta \in R$. But

$$N(\sigma\nu - \mu\tau) \leq N(\sigma)N(\nu) + N(\mu)N(\tau) \leq 2|d|^{2n},$$

and

$$N(\pi^k) = |d|^k > |d|^{2n+1} \geq 2|d|^{2n}.$$

It follows that $\sigma/\tau = \mu/\nu = \alpha$. Note that τ and π must be coprime. To see this, suppose π divides τ (the only other possibility since π is irreducible). We have $\sigma\nu = \mu\tau$, so π divides σ. Then $(\sigma/\pi)/(\tau/\pi) = \mu/\nu = \alpha$, so $(\sigma/\pi, \tau/\pi) \in L_k$, which contradicts the minimality of (σ, τ). Since (γ_1, χ_1) is the output of the algorithm, then $N(\gamma_1) < \sqrt{|d|^k/2}$ and $N(\chi_1) < \sqrt{|d|^k/2}$. We have $(\gamma_1, \chi_1) \in L_k(a)$, so $\Omega((\gamma_1, \chi_1)) < \sqrt{|d|^k/2}$. By Theorem 4.4.5, $(\gamma_1, \chi_1) = \omega(\sigma, \tau)$, for some $\omega \in R$. By Theorem 4.4.6 we have

$$\rho_1 \pi^k + \chi_1 \sum_{i=0}^{k-1} a_i \pi^i = \gamma_1.$$

Thus

$$\rho_1 \pi^k = \omega(\sigma - \tau \sum_{i=0}^{k-1} a_i \pi^i).$$

But by Theorem 4.4.6, ρ_1, χ_1 are coprime. So ω is a unit. (γ_1, χ_1) is also a minimal vector in L_k.

In conclusion,

$$\frac{\gamma_1}{\chi_1} = \frac{\sigma}{\tau} = \frac{\mu}{\nu} = \sum_{i=0}^{\infty} a_i \pi^i.$$

■

The Euclidean rational approximation algorithm runs in time $O(k^2 \log(k))$ if k elements are used. If $\gamma \in R$ is the remainder after dividing $\beta \in R$ into $\alpha \in R$ according to the division algorithm in R, then $N(\gamma) < cN(\beta)$ for some constant $c < 1$. Let $n = \max\{N(\pi^k), N(\sum_{i=0}^{k-1} a_i \pi^i)\}$, so $n \in O(2^k)$. Then the complexity is $O(\log(n) \cdot C(n))$, where $C(n)$ is the time required for one division of two elements .

If fast Fourier transforms are used for multiplication, then $C(n) \in O(k \cdot \log(k))$. So the total time complexity of the Euclidean rational approximation algorithm is $O(k^2 \log(k))$.

4.5 Comparison

In this section, we compare Xu's rational approximation algorithm (Figure 4.1) with Lattice Rational Approximation Algorithm (ApproxLattice, Figure 4.2) and the Extended Euclidean Rational Approximation Algorithm (EEAapprox, Figure 4.3).

4.5.1 ApproxLattice and Xu's algorithm

Let $R = \mathbb{Z}[\pi]$, where π is a root of the polynomial $x^2 = D$, which is an irreducible polynomial over $\mathbb{Z}$. The complete set is chosen to be $S = \{0, 1, \cdots, |D|-1\}$. For any $x = x_0 + x_1\pi, x_i \in \mathbb{Z}$, the corresponding size function used in Xu's algorithm is

$$\psi_{R,\pi}(x) = \max\{2\lfloor \log_{|D|} |x_0| \rfloor, \lfloor 2 \log_{|D|} |x_1| \rfloor + 1\}.$$

Then the size of the AFSR related to u/q where $u, q \in \mathbb{Z}[\pi]$ is

$$\Gamma_{R,\pi}(u, q) = \max\{\psi_{R,\pi}(u), \psi_{R,\pi}(q)\}.$$

The π-adic complexity defined for Xu's algorithm is

$$\lambda_\pi(\mathbf{a}) = \inf\{\Gamma_{R,\pi}(u, q) : \alpha = u/q\},$$

where α is the associated π-adic number for sequence $\mathbf{a}$.

The corresponding size function defined in ApproxLattice is

$$\varphi_{R,\pi}(x) = x_0^2 + x_1^2, \text{ where } x = x_0 + x_1\pi \in \mathbb{Z}[\pi].$$

Then the size of the AFSR related to u/q where $u, q \in \mathbb{Z}[\pi]$ is

$$\Phi_{R,\pi}(u, q) = \log_{|D|}(\varphi_{R,\pi}(u) + \varphi_{R,\pi}(q)).$$

The π-adic complexity defined for ApproxLattice is

$$\varphi_\pi(\mathbf{a}) = \inf\{\Phi_{R,\pi}(u, q) : \alpha = u/q\},$$

where α is the associated π-adic number for the sequence $\mathbf{a}$.

Let $\mathbf{a} = a_0, a_1, a_2 \cdots$ be an eventually periodic sequence over S. It can be associated with a π-adic number α. Assume u^*/q^* is a rational expression of α with

$\Phi_{R,\pi}(u^*, q^*) = \varphi_\pi(\mathbf{a})$. Let $u^* = u_0^* + u_1^*\pi$, $q^* = q_0^* + q_1^*\pi$ with $u_0^*, u_1^*, q_0^*, q_1^* \in \mathbb{Z}$. Assume $\bar{u}/\bar{q}$ is a rational expression of α with $\Gamma_{R,\pi}(\bar{u}, \bar{q}) = \lambda_\pi(\mathbf{a})$. Let $\bar{u} = \bar{u}_0 + \bar{u}_1\pi$, $\bar{q} = \bar{q}_0 + \bar{q}_1\pi$ with $\bar{u}_0, \bar{u}_1, \bar{q}_0, \bar{q}_1 \in \mathbb{Z}$. So we have

$$\begin{aligned}
\lambda_\pi(\mathbf{a}) &= \Gamma_{R,\pi}(\bar{u}, \bar{q}) \\
&\leq \Gamma_{R,\pi}(u^*, q^*) \\
&= \max\left(2\lfloor \log_{|D|} |u_0^*| \rfloor, 2\lfloor \log_{|D|} |u_1^*| \rfloor + 1, 2\lfloor \log_{|D|} |q_0^*| \rfloor, 2\lfloor \log_{|D|} |q_1^*| \rfloor + 1\right) \\
&\leq 2\max\left(\lfloor \log_{|D|} |u_0^*| \rfloor, \lfloor \log_{|D|} |u_1^*| \rfloor, \lfloor \log_{|D|} |q_0^*| \rfloor, \lfloor \log_{|D|} |q_1^*| \rfloor\right) + 1 \\
&\leq \log_{|D|}(\max(u_0^{*2}, u_1^{*2}, q_0^{*2}, q_1^{*2})) + 1 \\
&\leq \Phi_{R,\pi}(u^*, q^*) + 1 \\
&= \varphi_\pi(\mathbf{a}) + 1,
\end{aligned}$$

and

$$\begin{aligned}
\varphi_\pi(\mathbf{a}) &= \Phi_{R,\pi}(u^*, q^*) \\
&\leq \Phi_{R,\pi}(\bar{u}, \bar{q}) \\
&= \log_{|D|}(\bar{u}_0^2 + \bar{u}_1^2 + \bar{q}_0^2 + \bar{q}_1^2) \\
&\leq \log_{|D|}(4\max(\bar{u}_0^2, \bar{u}_1^2, \bar{q}_0^2, \bar{q}_1^2)) \\
&\leq \log_{|D|} 4 + 2\max\left(\lfloor \log_{|D|} |\bar{u}_0| \rfloor, \lfloor \log_{|D|} |\bar{u}_1| \rfloor, \lfloor \log_{|D|} |\bar{q}_0| \rfloor, \lfloor \log_{|D|} |\bar{q}_1| \rfloor\right) + 2 \\
&\leq \Gamma_{R,\pi}(\bar{u}, \bar{q}) + 2 + \log_{|D|} 4 \\
&= \lambda_\pi(\mathbf{a}) + 2 + \log_{|D|} 4.
\end{aligned}$$

That is,

$$\lambda_\pi(\mathbf{a}) - 1 \leq \varphi_\pi(\mathbf{a}) \leq \lambda_\pi(\mathbf{a}) + 2 + \log_{|D|} 4.$$

This means that $\lambda_\pi(\mathbf{a})$ and $\varphi_\pi(\mathbf{a})$ are almost the same neglecting small constants.

Xu's algorithm has worst case time complexity $O(\sum_{k=1}^{\lambda_\pi(\mathbf{a})} \sigma(k))$, where $\sigma(k)$ is the time needed to add two elements $a, b \in \mathbb{Z}[\pi]$ with the length of π-adic expansion at most k. So it runs in quadratic time. But it may not output the smallest AFSR for sequence $\mathbf{a}$. With the same time complexity, ApproxLattice can output the smallest AFSR with regard to the size function $\varphi_{R,\pi}$.

The number of terms needed to get the exact rational expression for Xu's algorithm is $O(\lambda_\pi(\mathbf{a}) \log(|S|))$. It grows with the the cardinality of the complete set S. However, ApproxLattice only needs $O(2\varphi_\pi(\mathbf{a}))$ terms to get the exact rational expression, with fixed coefficent.

4.5.2 EEAapprox and Xu's algorithm

Let R be the ring of integers of $\mathbb{Q}(\sqrt{d})$ where $\mathbb{Q}(\sqrt{d})$ is norm Euclidean and $d < -1$. That is, $d = -2, -3, -7, -11$. The size functions used in Xu's algorithm and EEAapprox are the same without considering the floor function. That is,

$$\psi_{R,\pi}(x) = \log_{|d|}(|N(x)|) = \log_{|d|}(x_0^2 - dx_1^2), \text{ where } x = x_0 + x_1\pi \in R.$$

Then the size of the AFSR related to u/q where $u, q \in \mathbb{Z}[\pi]$ is

$$\Psi(u, q) = \Gamma_{R,\pi}(u, q) = \max\{\psi_{R,\pi}(u), \psi_{R,\pi}(q)\}.$$

The π-adic complexity is

$$\phi_\pi(\mathbf{a}) = \lambda_\pi(\mathbf{a}) = \inf\{\Gamma_{R,\pi}(u, q) : \alpha = u/q\},$$

where α is the associated π-adic number for sequence $\mathbf{a}$.

Xu's algorithm runs in quadratic time. To get the smallest AFSR, we need to apply the extended Euclidean algorithm on the output u, q to find the greatest common divisor of u and q. So the complexity of Xu's algorithm and EEAapprox is the same.

Similarly as ApproxLattice, EEAapprox needs $O(2\phi_\pi(\mathbf{a}))$ terms to get the exact rational expression, which is better than the $O(\log |S|\phi_\pi(\mathbf{a}))$ required for Xu's algorithm.

4.5.3 EEAapprox and ApproxLattice

To compare EEAapprox and ApproxLattice, we require $\pi = \sqrt{-2}$, $R = \mathbb{Z}[\pi]$, and $S = \{0, 1\}$.

The corresponding size function used in ApproxLattice is

$$\varphi_{R,\pi}(x) = x_0^2 + x_1^2, \text{ where } x = x_0 + x_1\pi \in \mathbb{Z}[\pi].$$

Then the size of the AFSR related to u/q where $u, q \in \mathbb{Z}[\pi]$ is

$$\Phi_{R,\pi}(u, q) = \log(\varphi_{R,\pi}(u) + \varphi_{R,\pi}(q)).$$

The π-adic complexity defined for ApproxLattice is

$$\varphi_\pi(\mathbf{a}) = \inf\{\Phi_{R,\pi}(u, q) : \alpha = u/q\},$$

where α is the associated π-adic number for the sequence $\mathbf{a}$.

The size of the AFSR related to u/q where $u, q \in \mathbb{Z}[\pi]$ defined in EEAApprox is

$$\Psi(u, q) = \max(\log(|N(u)|), \log(|N(q)|)).$$

The π-adic complexity is

$$\phi_\pi(\mathbf{a}) = \inf\{\Psi_{R,\pi}(u, q) : \alpha = u/q\},$$

where α is the associated π-adic number for sequence $\mathbf{a}$.

Let $\mathbf{a} = a_0, a_1, a_2 \cdots$ be an eventually periodic sequence over S. It can be associated with a π-adic number $\alpha \in R_\pi$. Assume u^*/q^* is a rational expression of α with $\Phi_{R,\pi}(u^*, q^*) = \varphi_\pi(\mathbf{a})$. Let $u^* = u_0^* + u_1^*\pi$, $q^* = q_0^* + q_1^*\pi$ with $u_0^*, u_1^*, q_0^*, q_1^* \in \mathbb{Z}$. Assume $\hat{u}/\hat{q}$ is a rational expression of α with $\Psi_{R,\pi}(\hat{u}, \hat{q}) = \phi_\pi(\mathbf{a})$. Let $\hat{u} = \hat{u}_0 + \hat{u}_1\pi$, $\hat{q} = \hat{q}_0 + \hat{q}_1\pi$ with $\hat{u}_0, \hat{u}_1, \hat{q}_0, \hat{q}_1 \in \mathbb{Z}$. So we have

$$\begin{aligned}
\phi_\pi(\mathbf{a}) &= \Psi_{R,\pi}(\hat{u}, \hat{q}) \\
&\leq \Psi_{R,\pi}(u^*, q^*) \\
&= \max\left(\log(u_0^{*2} + 2u_1^{*2}), \log(q_0^{*2} + 2q_1^{*2})\right) \\
&\leq \log\left(2(u_0^{*2} + u_1^{*2} + q_0^{*2} + q_1^{*2})\right) \\
&= \varphi_\pi(\mathbf{a}) + 1,
\end{aligned}$$

and

$$\begin{aligned}
\varphi_\pi(\mathbf{a}) &= \Phi_{R,\pi}(u^*, q^*) \\
&\leq \Phi_{R,\pi}(\hat{u}, \hat{q}) \\
&= \log(\hat{u}_0^2 + \hat{u}_1^2 + \hat{q}_0^2 + \hat{q}_1^2) \\
&\leq \log\left(2\max(\hat{u}_0^2 + 2\hat{u}_1^2, \hat{q}_0^2 + 2\hat{q}_1^2)\right) \\
&= \phi_\pi(\mathbf{a}) + 1.
\end{aligned}$$

That is,

$$\phi_\pi(\mathbf{a}) - 1 \leq \varphi_\pi(\mathbf{a}) \leq \phi_\pi(\mathbf{a}) + 1.$$

Theorem 4.4.7 illustrates that when $k > 2\phi_\pi(\mathbf{a}) + 1$, EEAApprox outputs the smallest AFSR with respect to the size function Ψ. The number of bits needed for ApproxLattice is $2\varphi_\pi(\mathbf{a}) + 7$. So EEAApprox saves several bits. However, the time complexity of ApproxLattice is quadratic which is better than EEAApprox's $O(k^2 log(k))$.

5 Conclusions and Future work

This dissertation explores the problem of register synthesis with regard to different kinds of pseudorandom sequence generators. We discuss the complexity measures that are related to the synthesis algorithms, such as linear complexity, N-adic complexity, joint N-adic complexity, and π-adic complexity.

The main contribution of Chapter 2 is the study of the linear complexity of sequences generated by FCSRs. We give a lower bound of the linear complexity of two special FCSR sequences. Chapter 3 is about two synthesis algorithms, ApproxGreedy and ApproxLLL, which solve the problem of FCSR synthesis for multi-sequences based on lattice reduction algorithms. In Chapter 4, we develop two algorithms for the AFSR synthesis problem. The work on the lattice rational approximation algorithm has been published in the proceedings of the conference Sequences and Their Applications-SETA 2014 [41] and the work on the extended Euclidean rational approximation algorithm has been accepted by the journal Advances in Mathematics of Communications in 2015 [42].

In the future, I will continue my research on register synthesis problems and the analysis of the related complexity measures. I plan to work on three main related topics: the study of linear complexity, two-dimensional Euclidean algorithm and its applications on register synthesis, and AFSRs synthesis with the LLL algorithm.

5.1 The study of linear complexity

In addition to the two special cases in Chapter 2, we can also consider the case of 4-adic FCSRs with special connection integers. We want to determine whether we can use the same idea to study the sequences generated by AFSRs, such as the maximal period d-FCSR sequences. We believe that the complementary property should exist in some special d-FCSRs. This will help us find a characteristic polynomial of the corresponding d-FCSR sequences.

5.2 Two-dimensional Euclidean algorithm and its applications to register synthesis

Inspired by the generalized Euclidean algorithm that is used to solve the multi-sequence LFSR synthesis problem [16], I came up with an algorithm called two-dimensional Euclidean algorithm that generalizes the Euclidean algorithm over the integers (Figure 3.1). The Euclidean algorithm works over the integers because the

set of integers $\mathbb{Z}$ is an Euclidean domain. That is, the division with remainder property (Theorem 3.1.1) is true for the set of integers. Similarly, the two-dimensional Euclidean algorithm is based on the two-dimensional division with reminder property shown in Theorem 5.2.1.

Let $\pi^2 = N$, where $N \in \mathbb{Z}$ is square free. Assume that for any $\alpha = a + b\pi \in \mathbb{Z}[\pi]$, $\varphi(\alpha)$ is the size of α which is defined in Theorem 3.2.1. So $\varphi(\alpha) = \max\{|a|, |b|\}$.

Definition 5.2.1. *We define two sets* $[\pi^0]$ *and* $[\pi^1]$ *as :*

$$[\pi^0] = \{\alpha : \alpha = a + b\pi \in \mathbb{Z}[\pi], |a| > |b|\},$$

and

$$[\pi^1] = \{\alpha : \alpha = a + b\pi \in \mathbb{Z}[\pi], |a| \leq |b|, \alpha \neq 0\}.$$

Notice that $\mathbb{Z}[\pi] = [\pi^0] \cup [\pi^1] \cup \{0\}$. This is a partition of $\mathbb{Z}[\pi]$. The equivalence relation based on this partition is:

$$\alpha \sim \beta \text{ if } \alpha \in [\pi^0] \text{ and } \beta \in [\pi^0], \text{ if } a \in [\pi^1] \text{ and } b \in [\pi^1], \text{ or if } \alpha = \beta = 0.$$

Lemma 5.2.1. *Let* $\alpha = a + b\pi, a, b \in \mathbb{Z}$, *and* $\beta = c + d\pi, c, d \in \mathbb{Z}$. *Suppose* $\alpha \in [\pi^0]$ *and* $\beta \in [\pi^1]$, *then the vectors* (a, b) *and* (c, d) *are linearly independent over* $\mathbb{R}$.

Proof: If there exists $k \in \mathbb{R}$ such that $(a, b) = k(c, d)$ and $\alpha \in [\pi^0]$, then $|a| > |b|$. Therefore $|kc| > |kd|$. But $\beta \in [\pi^1]$. □

Theorem 5.2.1. *(Two-dimensional division with reminder) Let* $\alpha = a+b\pi \neq 0, a, b \in \mathbb{Z}$, $\beta_0 = c + d\pi, c, d \in \mathbb{Z}$, *and* $\beta_1 = e + f\pi, e, f \in \mathbb{Z}$. *Suppose* $\beta_0 \in [\pi^0]$ *and* $\beta_1 \in [\pi^1]$. *There exist* $q_0, q_1 \in \mathbb{Z}$ *and* $\gamma = g + f\pi \in \mathbb{Z}[\pi]$ *such that*

$$\alpha = q_0\beta_0 + q_1\beta_1 + \gamma$$

where $|g| < \varphi(\beta_0) = |c|$ *and* $\gamma \in [\pi^0]$, *or* $|h| < \varphi(\beta_1) = |f|$ *and* $\gamma \in [\pi^1]$.

The proof is given in Appendix. The proof not only shows the existence of the two-dimensional division with remainder but also identifies how to do the computation. The two-dimensional Euclidean algorithm can be described as below.

Given $\alpha_1, \beta_1^{(0)}$, and $\beta_1^{(1)} \in \mathbb{Z}[\pi]$, let $\alpha_1 \in [\pi^{(v_0)}]$ ($v_0 = 0$ or 1), $\beta_1^{(0)} \in [\pi^0]$ and $\beta_1^{(1)} \in [\pi^1]$ and $\varphi(\alpha) \geq \varphi(\beta_1^{v_0})$. We repeatedly apply the two-dimensional division with reminder to obtain the following series of equations

$$\alpha_j = q_j^{(0)}\beta_j^{(0)} + q_j^{(1)}\beta_j^{(1)} + \gamma_j \text{ for } j = 1, 2, 3, 4, \cdots, \tag{5.1}$$

until $j = t$ for some t such that $\gamma_t = 0$. These equations also have to satisfy the following requirements:

1. $\varphi(\gamma_j) < \varphi(\beta_j^{(v_j)})$, for some $v_j \in \{0,1\}$ such that $\gamma_j \sim \beta_j^{(v_j)}$.

2. $\alpha_{j+1} = \beta_j^{v_j}$.

3. $\beta_{j+1}^{(v_j)} = \gamma_j$.

4. $\beta_{j+1}^{(h)} = \beta_{j+1}^{(h)}$ for $h \neq v_j$.

The first requirement is guaranteed by the two-dimensional division and the other three specify the updates from step j to step $j+1$. Lemma 5.2.2 follows directly from these requirements and ensures that the iterations will stop at step t.

Lemma 5.2.2. *1. $\varphi(\beta_{j+1}^{(0)}) \leq \varphi(\beta_j^{(0)})$ and $\varphi(\beta_{j+1}^{(1)}) \leq \varphi(\beta_j^{(1)})$.*

2. Whenever $\gamma_i \sim \gamma_j$ for $i < j$, then $\varphi(\gamma_i) > \varphi(\gamma_j)$.

Example 5.2.1. *Take $\pi^2 = 2$. We let $\alpha = 2340 + 2184\pi$, $\beta_1^{(0)} = 2048 + 0\pi$, and $\beta_1^{(1)} = 0 + 2048\pi$. The two-dimensional Euclidean algorithm preforms as a chain of equations shown below.*

$$\begin{aligned}
2340 + 2184\pi &= 1 \cdot (2048 + 0\pi) + 1 \cdot (0 + 2048\pi) + (292 + 136\pi) \\
2048 + 0\pi &= 7 \cdot (292 + 136\pi) + 0 \cdot (0 + 2048\pi) + (4 - 952\pi) \\
0 + 2048\pi &= 0 \cdot (292 + 136\pi) + 2 \cdot (4 - 952\pi) + (8 + 144\pi) \\
4 - 952\pi &= 0 \cdot (292 + 136\pi) + (-6) \cdot (8 + 144\pi) + (52 - 88\pi) \\
8 + 144\pi &= 0 \cdot (292 + 136\pi) + (-1) \cdot (52 - 88\pi) + (60 + 56\pi) \\
292 + 136\pi &= 4 \cdot (60 + 56\pi) + 1 \cdot (52 - 88\pi) + (0 + 0\pi)
\end{aligned}$$

From Equation 5.1, we know that

$$\gamma_j = \alpha_j - q_j^{(0)}\beta_j^{(0)} - q_j^{(1)}\beta_j^{(1)} \quad \text{for } 1 \leq j \leq t.$$

Without loss of generality, we suppose $\alpha_j \in [\pi^0]$. Then

$$\alpha_j = \beta_{j-1}^{(0)}, \tag{5.2}$$

$$\beta_j^{(0)} = \gamma_{j-1}, \tag{5.3}$$

$$\text{and} \quad \beta_j^{(1)} = \beta_{j-1}^{(1)}. \tag{5.4}$$

So

$$\gamma_j = \beta_{j-1}^{(0)} - q_j^{(0)}\gamma_{j-1} - q_j^{(1)}\beta_{j-1}^{(1)} \quad \text{for } 1 < j \leq t. \tag{5.5}$$

Recursively, for every $1 \leq j \leq t$, we will have

$$\gamma_j = Q_j\alpha + Q_j^{(0)}\beta_1^{(0)} + Q_j^{(1)}\beta_1^{(1)} \quad \text{for some } Q_j, Q_j^{(0)}, Q_j^{(1)} \in \mathbb{Z}. \tag{5.6}$$

In Example 5.2.1, we have the following equations:

$$\begin{aligned}
\gamma_1 &= 292 + 136\pi = (2340 + 2184\pi) - (2048 + 0\pi) - (0 + 2048\pi) \\
\gamma_2 &= 4 - 952\pi = (-7)(2340 + 2184\pi) + 8(2048 + 0\pi) + 7(0 + 2048\pi) \\
\gamma_3 &= 8 + 144\pi = (-14)(2340 + 2184\pi) + 16(2048 + 0\pi) + 15(0 + 2048\pi) \\
\gamma_4 &= 52 - 88\pi = (-91)(2340 + 2184\pi) + 104(2048 + 0\pi) + 97(0 + 2048\pi) \\
\gamma_5 &= 60 + 56\pi = (-105)(2340 + 2184\pi) + 120(2048 + 0\pi) + 112(0 + 2048\pi) \\
\gamma_6 &= 0 + 0\pi = 512(2340 + 2184\pi) + (-585)(2048 + 0\pi) + (-546)(0 + 2048\pi).
\end{aligned}$$

Consider the 2-fold N-ary eventually periodic multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)})$. Let $\varsigma \in R_\pi$, where R_π is the ring of π-adic numbers with $R = \mathbb{Z}[\pi]$ and $\pi^2 = N$. Suppose ς is associated with the interleaved sequence. That is,

$$\begin{aligned}
\varsigma &= s_0^{(0)} + s_0^{(1)}\pi + s_1^{(0)}\pi^2 + s_1^{(1)}\pi^3 + s_2^{(0)}\pi^4 + s_2^{(1)}\pi^5 + \cdots \qquad (5.7) \\
&= \gamma/q \text{ for some } \gamma \in \mathbb{Z}[\pi] \text{ and } q \in \mathbb{Z}.
\end{aligned}$$

Without loss of generality, we suppose k is even. Assume that

$$\varsigma_k = s_0^{(0)} + s_0^{(1)}\pi + s_1^{(0)}\pi^2 + s_1^{(1)}\pi^3 + \cdots + s_{k/2-1}^{(0)}\pi^{k-1} + s_{k/2-1}^{(1)}\pi^k.$$

We execute the two-dimensional Euclidean algorithm with $\alpha = \varsigma_k, \beta_1^{(0)} = \pi^k$, and $\beta_1^{(1)} = \pi^{k+1}$. According to Equation (5.6) for every each j, we have

$$\gamma_j = Q_j\varsigma_k + Q_j^{(0)}\pi^k + Q_j^{(1)}\pi^{k+1} \quad \text{for some } Q_j, Q_j^{(0)}, Q_j^{(1)} \in \mathbb{Z}.$$

It also means that

$$\gamma_j \equiv Q_j\varsigma_k \equiv Q_j\varsigma \pmod{\pi^k}.$$

If $\gamma_j = r_j^{(0)} + r_j^{(1)}\pi$, then $(r_j^{(0)}, r_j^{(1)}, Q_j)$ is in the kth integer approximation lattice of ς for every j. Instead of stopping at step j where $\gamma_j = 0$, we stop when $\varphi(\gamma_j)$ first becomes less than $|Q_j|$. In Example 5.2.1, the iteration will stop at $j = 5$ where $\gamma_5 = 60 + 56\pi$, $Q_5 = -105$.

Assumption 5.2.1. *Suppose that N is not a square and the joint N-adic complexity, $\lambda_{N,2}(\mathcal{S})$, of the multi-sequence $\mathcal{S} = (\mathbf{S}^{(0)}, \mathbf{S}^{(1)})$ is less than or equal to n. We assume that $k > 2n + c$, for some constant c. Let the two-dimensional Euclidean algorithm be executed with $\alpha = \varsigma_k, \beta_1^{(0)} = \pi^k$, and $\beta_1^{(1)} = \pi^{k+1}$ and let it be stopped when $\varphi(\gamma_j)$ first becomes less than $|Q_j|$. Then*

$$\varsigma = \frac{\gamma_j}{Q_j}$$

and $\max(\varphi(\gamma), |Q_j|) = \lambda_{N,2}(\mathcal{S})$.

This assumption is similar to the assumption in Theorem 3.1.3 for the extended Euclidean rational approximation algorithm (Figure 3.2). However, experimental results show that $\varsigma = \gamma_j/Q_j$ is not alway true. When $N = 2$, Figure 5.1 shows how the number of iterations grows as k becomes larger. When $k \leq 34$, we tested every possible element in $\mathbb{Z}[\pi]$. The number of iterations for the algorithm to stop is shown as the blue line. When $k > 34$, we randomly selected 2^{30} elements in $\mathbb{Z}[\pi]$. The number of iterations required is shown as the red line. For each iteration, the complexity is determined by the division of two integers which are less than $N^{k/2}$. So the total complexity may be $O(k^2 \log k)$ for the two-dimensional Euclidean algorithm.

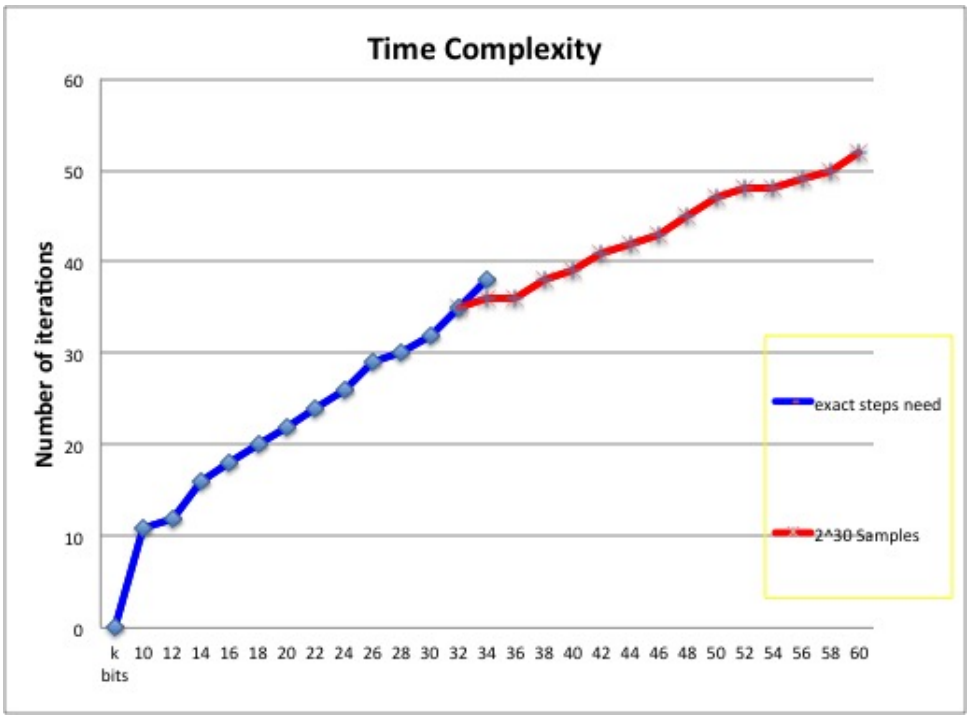

Figure 5.1: Number of iterations for the two-dimensional Euclidean algorithm

We can generalize the two-dimensional Euclidean algorithm to higher dimensions. Let $R = \mathbb{Z}[\pi]$ and $\pi^M = N$, where $x^M - N$ is irreducible over the rational numbers. For any element $\alpha \in R$, α has the form $\alpha = a_0 + a_1\pi + \cdots + a_{M-1}\pi^{M-1}$. Define a partition $\{[\pi^0], [\pi^1], \cdots, [\pi^{M-1}], \{0\}\}$, where

$$[\pi^i] = \{\alpha \neq 0 : |a_j| \leq |a_i| \text{ if } j < i \text{ and } |a_j| < |a_i| \text{ if } j > i\}.$$

The N-dimensional division with reminder property can be stated as: let $\alpha = a_0 + a_1\pi + \cdots + a_{M-1}\pi^{M-1} \in \mathbb{Z}[\pi]$ and $\alpha \neq 0$. Suppose $\beta^{(i)} \in [\pi^i]$ for $i = 0, 1, 2, \cdots, M-1$. There exist $q_0, q_1, \cdots, q_{M-1} \in \mathbb{Z}$ and $\gamma \in \mathbb{Z}[\pi]$ such that

$$\alpha = q_0\beta^{(0)} + q_1\beta^{(1)} + \cdots + q_{M-1}\beta^{(M-1)} + \gamma,$$

where $\varphi(\gamma) < \varphi(\beta^{i_0})$ if $\gamma \sim \beta^{(i_0)}$. So the N-dimensional Euclidean algorithm can be studied using the same method as the two-dimensional Euclidean algorithm.

5.3 AFSRs synthesis with the LLL algorithm

Based on the lattice reduction greedy algorithm, GreedyLatticeReduction (Figure 1.9), we proposed the lattice rational approximation algorithm, ApproxLattice (Figure 4.2), which solves the AFSR synthesis problem for AFSRs over $R = \mathbb{Z}[\pi]$ and $\pi^2 = D$. We may ask whether the approach can be extended to cubic or higher extensions of $\mathbb{Z}$. This becomes complicated because of the complexity of GreedyLatticeReduction. However, we can consider other lattice reduction algorithms, such as the LLL algorithm. It is possible that we can extend ApproxLattice with the LLL algorithm to solve the synthesis problem for all d-FCSRs (Definition 1.3.7). A d-FCSR is an AFSR over $(R = \mathbb{Z}[\pi], \pi, S)$, where $N \geq 2$ and $d \geq 1$ are integers such that the polynomial $x^d - N$ is irreducible over the rational number field $\mathbb{Q}$, $\pi \in \mathbb{C}$ is a root of this polynomial in an extension field of $\mathbb{Q}$, and $S = \mathbb{Z}/(N) = \{0, 1, 2, \cdots, N-1\}$. In this case, any eventually periodic sequence $\mathbf{a}$ over S can be identified with an element α in R_π and $\alpha = u/q$ for some $u, q \in R = \mathbb{Z}[\pi]$. Suppose $q = q_0 + q_1\pi + \cdots + q_{d-1}\pi^{d-1}$ and $u = u_0 + u_1\pi + \cdots + u_{d-1}\pi^{d-1}$,where $q_i, u_i \in \mathbb{Z}$ for $i = 0, 1, 2 \cdots, d-1$. We can defined the kth integer approximation lattice of α as :

$$L_k(\alpha) := \{(u_0, \ldots, u_{d-1}, q_0, \ldots, q_{d-1}) \in \mathbb{Z}^{2d} : \alpha q - u \equiv 0 \pmod{\pi^k}\}.$$

The LLL algorithm will find a vector in this integer lattice that almost has the minimal super norm. It is possible that when k is sufficiently large, the vector found gives exactly a rational expression of the smallest d-FCSR for the given sequence $\mathbf{a}$.

Appendix

Proof of Theorem 5.2.1.

Theorem 5.2.1 (Two-dimensional division with reminder) Let $\alpha = a + b\pi \neq 0, a, b \in \mathbb{Z}$, $\beta_0 = c + d\pi, c, d \in \mathbb{Z}$, and $\beta_1 = e + f\pi, e, f \in \mathbb{Z}$. Suppose $\beta_0 \in [\pi^0]$ and $\beta_1 \in [\pi^1]$. There exist $q_0, q_1 \in \mathbb{Z}$ and $\gamma = g + f\pi \in \mathbb{Z}[\pi]$ such that

$$\alpha = q_0\beta_0 + q_1\beta_1 + \gamma$$

where $|g| < \varphi(\beta_0) = |c|$ and $\gamma \in [\pi^0]$, or $|h| < \varphi(\beta_1) = |f|$ and $\gamma \in [\pi^1]$.

Proof: From Lemma 5.2.1 we have that (c, d) and (e, f) are linearly independent, so there exists $x, y \in \mathbb{R}$ such that

$$(a, b) = x(c, d) + y(e, f).$$

Let $\lfloor x \rceil$ be the nearest integer to x (if $x = n + 1/2$ for some $n \in \mathbb{Z}$, then let $\lfloor x \rceil = n$). Let $\bar{x} = x - \lfloor x \rceil$. So $0 \leq |\bar{x}| \leq 1/2$. Similarly, $0 \leq |\bar{y}| \leq 1/2$. Let

$$a_0 + b_0\pi = a + b\pi - \lfloor x \rceil(c + d\pi) - \lfloor y \rceil(e + f\pi) = \bar{x}(c + d\pi) + \bar{y}(e + f\pi).$$

We have $a_0 + b_0\pi \in \mathbb{Z}[\pi]$, $a_0 = \bar{x}c + \bar{y}e$, and $b_0 = \bar{x}d + \bar{y}f$. We now proceed by cases.

Case 1 $|c| > |d| \geq |f| \geq |e|$

If $a_0 + b_0 \in [\pi^0]$, then let $\gamma = a_0 + b_0\pi$, $q_0 = \lfloor x \rceil$, $q_1 = \lfloor y \rceil$. We have $|a_0| < |c|$.

If $a_0 + b_0\pi \in [\pi^1]$ and $|b_0| < |f|$, then let $\gamma = a_0 + b_0\pi$, $q_0 = \lfloor x \rceil$, $q_1 = \lfloor y \rceil$.

If $a_0 + b_0\pi \in [\pi^1]$, and $|b_0| \geq |f|$, then we discuss the problem in the four cases. Without loss of generality, we let $c > 0$, $f > 0$ and $b_0 \geq 0$. Also $d \neq 0$, otherwise $\beta_1 = e + f\pi = 0$.

For $b_0, f \in \mathbb{Z}$, there exist unique $l, b_0' \in \mathbb{Z}$ such that

$$b_0 = lf + b_0',$$

where $0 \leq b_0' < f$ and $l \geq 1$.

We have four cases to consider based on the value of d and e.

1. $d > 0$ and $e \geq 0$,
   - If $a_0 \geq 0$, we let $a_1 + b_1\pi = a_0 + b_0\pi - l(e + f\pi) = (a_0 - le) + b_0'\pi$. We have

     $$|a_1| = |a_0 - le| \leq \max\{a_0, le\} \leq \max\{a_0, lf\} \leq \max\{a_0, b_0\} = b_0 < c.$$

     Also, $|b_1| = |b_0'| < f$.
     So in this case, we let $q_0 = \lfloor x \rceil$, $q_1 = \lfloor y \rceil + l$ and $\gamma = a_1 + b_1\pi$.

- If $a_0 < 0$, we can consider the following four cases.
  - $\bar{x} \geq 0$ and $\bar{y} \geq 0$
    $a_0 = \bar{x}c + \bar{y}e \geq 0$. It is a contradiction that $a_0 < 0$.
  - $\bar{x} \geq 0$ and $\bar{y} < 0$
    $\bar{x}c \geq \bar{x}d \geq -\bar{y}f \geq -\bar{y}e$.
    This is a contradiction to $a_0 = \bar{x}c + \bar{y}e < 0$.
  - $\bar{x} < 0$ and $\bar{y} \geq 0$
    $b_0 = \bar{x}d + \bar{y}f \leq \bar{y}f < f$. It is a contradiction that $b_0 \geq f$.
  - $\bar{x} < 0$ and $\bar{y} < 0$
    $b_0 = \bar{x}d + \bar{y}f < 0$. It is a contradiction.

  So a_0 can not less than 0 if $d > 0$ and $e \geq 0$.

2. $d > 0$ and $e \leq 0$
   - If $a_0 > 0$, we let $a_2 + b_2\pi = a_0 + b_0\pi - (c + d\pi)$. So

     $$|a_2| = |a_0 - c| = c - a_0 > d - b_0 = |b_2|.$$

     And $|a_2| < c$. So $q_0 = \lfloor x \rceil + 1$, $q_0 = \lfloor y \rceil$ and $\gamma = a_2 + b_2\pi$ are what we want.
   - If $a_0 \leq 0$, we consider the following six cases.
     - $\bar{x} > 0$ and $\bar{y} > 0$
       $b_0 = \bar{x}d + \bar{y}f < \bar{x}c + \bar{y}f \leq -\bar{y}e + \bar{y}f = \bar{y}(|e| + f) \leq 2\bar{y}f$.
       This is a contradiction to $b_0 \geq f$.
     - $\bar{x} > 0$ and $\bar{y} \leq 0$
       It is contradiction that $a_0 = \bar{x}c + \bar{y}e \leq 0$
     - $\bar{x} = 0$ and $\bar{y} \geq 0$.
       $b_0 = \bar{y}f < f$. It is contradiction that $b_0 \geq f$.
     - $\bar{x} = 0$ and $\bar{y} < 0$
       $b_0 = \bar{y}f < 0$. It is contradiction that $b_0 \geq 0$.
     - $\bar{x} < 0$ and $\bar{y} > 0$
       $b_0 = \bar{x}d + \bar{y}f < \bar{y}f < f$. It is contradiction that $b_0 \geq f$.
     - $\bar{x} < 0$ and $\bar{y} \leq 0$
       $b_0 = \bar{x}d + \bar{y}f < 0$. It is contradiction that $b_0 \geq 0$.

   So $a_0 \leq 0$ is not valid when $d > 0$ and $e \leq 0$.

3. $d < 0$ and $e \geq 0$

- If $a_0 < 0$, we let $a_3 + b_3\pi = a_0 + b_0\pi + (c + d\pi)$. So

$$|b_3| = |b_0 + d| = |d| - b_0 < c - b_0 \leq c + a_0 = |a_3|.$$

And $|a_3| = |c + a_0| = c - |a_0| < c$.
So $q_0 = \lfloor x \rceil - 1$, $q_0 = \lfloor y \rceil$ and $\gamma = a_3 + b_3\pi$ are what we want.

- The inequality $a_0 \geq 0$ cannot happen because the following discussions on $\bar{x}$ and $\bar{y}$.
  - $\bar{x} > 0$ and $\bar{y} > 0$
    $b_0 = \bar{x}d + \bar{y}f < \bar{y}f < f$. It is contradiction that $b_0 \geq f$.
  - $\bar{x} > 0$ and $\bar{y} \leq 0$
    $b_0 = \bar{x}d + \bar{y}f < 0$. It is contradiction that $b_0 \geq 0$.
  - $\bar{x} = 0$
    $b_0 = \bar{y}f < f$. It is contradiction that $b_0 \geq f$.
  - $\bar{x} < 0$ and $\bar{y} \geq 0$
    $b_0 = \bar{x}d + \bar{y}f = |\bar{x}||d| + \bar{y}f < |\bar{x}|c + \bar{y}f \leq \bar{y}e + \bar{y}f \leq f$.
    It is contradiction that $b_0 \geq f$.
  - $\bar{x} < 0$ and $\bar{y} < 0$
    $a_0 = \bar{x}c + \bar{y}e < 0$. It is contradiction that $a_0 \geq 0$.

4. $d < 0$ and $e \leq 0$

- If $a_0 \leq 0$, we let $a_4 + b_4\pi = a_0 + b_0\pi - l(e + f\pi) = (a_0 - le) + b_0'\pi$.
We have

$$|a_4| = |a_0 - le| \leq \max\{|a_0|, |le|\} \leq \max\{|a_0|, lf\} \leq \max\{|a_0|, b_0\} = b_0 < c.$$

Also, $|b_4| = |b_0'| < f$.
So in the case, we let $q_0 = \lfloor x \rceil$, $q_1 = \lfloor y \rceil + l$ and $\gamma = a_4 + b_4\pi$.

- The inequality $a_0 > 0$ cannot happen because the following discussions on $\bar{x}$ and $\bar{y}$.
  - $\bar{x} \leq 0$ and $\bar{y} \geq 0$
    $a_0 = \bar{x}c + \bar{y}e \leq 0$. It is a contradiction that $a_0 > 0$.
  - $\bar{x} \leq 0$ and $\bar{y} < 0$
    $\bar{y}e = |\bar{y}||e| \leq |\bar{y}|f = -\bar{y}f \leq \bar{x}d \leq -\bar{x}c$.
    It is a contradiction to $a_0 = \bar{x}c + \bar{y}e > 0$.
  - $\bar{x} > 0$ and $\bar{y} \leq 0$
    $b_0 = \bar{x}d + \bar{y}f < 0$. It is a contradiction that $b_0 \geq 0$.

– $\bar{x} > 0$ and $\bar{y} > 0$

$b_0 = \bar{x}d + \bar{y}f < \bar{y}f < f$. It is a contradiction that $b_0 \geq f$.

Case 2 $|f| \geq |e| \geq |c| > |d|$ It is similar to Case 1.

Case 3 all other c,d,e,f

We have $|c| > |e|$ and $|f| > |d|$, so $|a_0| = |\bar{x}c + \bar{y}e| \leq \frac{1}{2}(|c| + |e|) < |c|$ and $|b_0| = \bar{x}d + \bar{y}f \leq \frac{1}{2}(|d| + |f|) < |f|$. Let $\gamma = a_0 + b_0\pi$, $q_0 = \lfloor x \rceil$, $q_1 = \lfloor y \rceil$.

□

Bibliography

[1] Random.org. https://www.random.org/.

[2] M. Ajtai. The shortest vector problem in L^2 is NP-hard for randomized reductions. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 10–19, New York, NY, USA, 1998. ACM.

[3] F. Arnault, T. Berger, and C. Lauradoux. Update on F-FCSR stream cipher. In *SASC, State of the Art of Stream Ciphers Workshop, Leuven, Belgium*, pages 267–277, 2006.

[4] F. Arnault, T. Berger, C. Lauradoux, and M. Minier. X-FCSR–a new software oriented stream cipher based upon FCSRs. In K. Srinathan, C. P. Rangan, and M. Yung, editors, *Progress in Cryptology–INDOCRYPT 2007*, pages 341–350. Springer, 2007.

[5] F. Arnault, T. Berger, C. Lauradoux, M. Minier, and B. Pousse. A new approach for FCSRs. In M. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867, pages 433–448. Springer, 2009.

[6] F. Arnault and T. P. Berger. F-FCSR: design of a new class of stream ciphers. In H. Gilbert and H. Handschuh, editors, *Fast Software Encryption*, pages 83–97. Springer, 2005.

[7] F. Arnault, T. P. Berger, and A. Necer. Feedback with carry shift registers synthesis with the Euclidean algorithm. *IEEE Transactions on Information Theory*, 50(5):910–917, May 2004.

[8] P. M. Cohn. *Algebraic numbers and algebraic functions*, volume 4. CRC Press, 1991.

[9] G. Cooke. A weakening of the Euclidean property for integral domains and applications to algebraic number theory. *Journal fr Mathematik. Band*, 282:18, 1976.

[10] D. Coppersmith. Finding a small root of a univariate modular equation. In U. Maurer, editor, *Advances in cryptology—EUROCRYPT'96*, pages 155–165. Springer, 1996.

[11] D. Coppersmith and A. Shamir. Lattice attacks on ntru. In W. Fumy, editor, *Advances in Cryptology—EUROCRYPT'97*, pages 52–61. Springer, 1997.

[12] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2001.

[13] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In E. Biham, editor, *Advances in Cryptology—EUROCRYPT 2003*, pages 345–359. Springer, 2003.

[14] B. de Weger. Approximation lattices of p-adic numbers. *Journal of Number Theory*, 24(1):70–88, 1986.

[15] C. Dwork. Lattices and their application to cryptography. *Lecture Notes, Stanford University*, 1998.

[16] G. Feng and K. Tzeng. A generalized Euclidean algorithm for multisequence shift-register synthesis. *IEEE Transactions on Information Theory*, 35(3):584–594, 1989.

[17] P. FIBS. 140-1, Federal Information Processing Standards Publication,(Jan. 11, 1994) Security Requirements for Cryptographic Modules, US Department of Commerce. *Brown, Secretary, National Institute of Standards and Technology*, pages 1–51.

[18] S. Fischer, W. Meier, and D. Stegemann. Equivalent representations of the F-FCSR keystream generator. In *ECRYPT Network of Excellence-SASC Workshop*, pages 87–94, 2008.

[19] A. M. Frieze, J. Hastad, R. Kannan, J. C. Lagarias, and A. Shamir. Reconstructing truncated integer variables satisfying linear congruences. *SIAM Journal on Computing*, 17(2):262–280, 1988.

[20] C. F. Gauss. *Disquisitiones Arithmeticae*. reprinted in English translation by Yale University Press, 1966.

[21] S. W. Golomb. *Shift register sequences*. Aegean Park Press, 1982.

[22] M. Goresky and A. Klapper. Feedback registers based on ramified extensions of the 2-adic numbers. In A. De Santis, editor, *Advances in Cryptology—EUROCRYPT '94*, volume 950, pages 215–222. Springer, 1995.

[23] M. Goresky and A. Klapper. Fibonacci and Galois representations of feedback-with-carry shift registers. *IEEE Transactions on Information Theory*, 48(11):2826–2836, 2002.

[24] M. Goresky and A. Klapper. Periodicity and correlation properties of d-FCSR sequences. *Designs, Codes and Cryptography*, 33(2):123–148, 2004.

[25] M. Goresky and A. Klapper. *Algebraic Shift Register Sequences*. Cambridge University Press, 2012.

[26] M. Hell and T. Johansson. Breaking the stream ciphers F-FCSR-H and F-FCSR-16 in real time. *Journal of Cryptology*, 24(3):427–445, 2011.

[27] C. Hermite. Extraits de lettres de m. ch. hermite à m. jacobi sur différents objects de la théorie des nombres. *Journal für die reine und angewandte Mathematik*, 40:261–277, 1850.

[28] N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In M. Darnell, editor, *Crytography and Coding*, pages 131–142. Springer, 1997.

[29] H. Hu, L. Hu, and D. Feng. On the expected value of the joint 2-adic complexity of periodic binary multisequences. In G. Gong, T. Helleseth, H. Song, and K. Yang, editors, *Sequences and Their Applications–SETA 2006*, pages 199–208. Springer, 2006.

[30] A. Klapper and M. Goresky. Cryptanalysis based on 2-adic rational approximation. In D. Coppersmith, editor, *Advances in Cryptology—CRYPTO'95*, volume 963, pages 262–273. Springer, 1995.

[31] A. Klapper and M. Goresky. Feedback shift registers, 2-adic span, and combiners with memory. *Journal of Cryptology*, 10(2):111–147, 1997.

[32] A. Klapper and J. Xu. Algebraic feedback shift registers. *Theoretical Computer Science*, 226(1):61–92, 1999.

[33] A. Klapper and J. Xu. Register synthesis for algebraic feedback shift registers based on non-primes. *Designs, Codes and Cryptography*, 31(3):227–250, 2004.

[34] N. Koblitz. *p-adic Numbers, p-adic Analysis, and Zeta-Functions*, volume 58. Springer Science & Business Media, 2012.

[35] A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Mathematische Annalen*, 6(3):366–389, 1873.

[36] D. Lee, J. Kim, J. Hong, J. Han, and D. Moon. Algebraic attacks on summation generators. In B. Roy and W. Meier, editors, *Fast Software Encryption*, pages 34–48. Springer, 2004.

[37] A. Lenstra, H. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[38] W. LeVeque. *Topics in number theory*. Courier Corporation, 2002.

[39] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications*. Cambridge University Press, 1994.

[40] R. Lidl and H. Niederreiter. *Finite fields*, volume 20. Cambridge University Press, 1997.

[41] W. Liu and A. Klapper. A lattice rational approximation algorithm for AFSRs over quadratic integer rings. In K. Schmidt and A. Winterhof, editors, *Sequences and Their Applications-SETA 2014*, pages 200–211. Springer, 2014.

[42] W. Liu and A. Klapper. AFSRs synthesis with extended euclidean rational approximation algorithm. *Advances in Mathematics of Communications*, 2015. Accepted.

[43] K. Mahler. On a geometrical representation of p-adic numbers. *The Annals of Mathematics*, 41(1):8–56, 1940.

[44] J. L. Massey. Shift register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.

[45] A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC press, 1996.

[46] R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *Information Theory, IEEE Transactions on*, 24(5):525–530, 1978.

[47] F. Miller. *Telegraphic code to insure privacy and secrecy in the transmission of telegrams*. CM Cornwell, 1882.

[48] H. Minkowski. *Geometrie der zahlen*. Teubner-Verlag, 1896.

[49] P. Q. Nguyen and D. Stehlé. An LLL algorithm with quadratic complexity. *SIAM Journal on Computing*, 39(3):874–903, 2009.

[50] P. Q. Nguyen and D. Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Transactions on Algorithms (TALG)*, 5(4):46, 2009.

[51] P. Q. Nguyen and J. Stern. Lattice reduction in cryptology: An update. In W. Bosma, editor, *Algorithmic Number Theory*, pages 85–112. Springer, 2000.

[52] H. Niederreiter. The probabilistic theory of linear complexity. In D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and C. G. Günther, editors, *Advances in Cryptology—EUROCRYPT'88*, pages 191–209. Springer, 1988.

[53] H. Niederreiter. Sequences with almost perfect linear complexity profile. In D. Chaum and P. W., editors, *Advances in Cryptology—EUROCRYPT'87*, pages 37–51. Springer, 1988.

[54] W. Qi and H. Xu. On the linear complexity of fcsr sequences. *Applied mathematics-A journal of Chinese universities*, 18(3):318–324, 2003.

[55] A. Robert. *A course in p-adic analysis*, volume 198. Springer, 2000.

[56] R. A. Rueppel. Linear complexity and random sequences. In F. Pichler, editor, *Advances in Cryptology—EUROCRYPT'85*, pages 167–188. Springer, 1985.

[57] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, DTIC Document, 2001.

[58] S. Ryshkov. Hermite, minkowski and venkov reduction of positive quadratic forms of n variables. *DOKLADY AKADEMII NAUK SSSR*, 207(5):1054–1056, 1972.

[59] S. Sakata. Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array. *Journal of Symbolic Computation*, 5(3):321–337, 1988.

[60] G. Schmidt and V. Sidorenko. Multi-sequence linear shift-register synthesis: The varying length case. In *Information Theory, 2006 IEEE International Symposium on*, pages 1738–1742, 2006.

[61] C. Seo, S. Lee, Y. Sung, K. Han, and S. Kim. A lower bound on the linear span of an fcsr. *Information Theory, IEEE Transactions on*, 46(2):691–693, 2000.

[62] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[63] V. Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, 2008.

[64] P. Stankovski, M. Hell, and T. Johansson. An efficient state recovery attack on X-FCSR-256. In O. Dunkelman, editor, *Fast Software Encryption*, pages 23–37. Springer, 2009.

[65] P. Stankovski, M. Hell, and T. Johansson. An efficient state recovery attack on the x-fcsr family of stream ciphers. *Journal of cryptology*, 27(1):1–22, 2014.

[66] J. Stern and P. Toffin. Cryptanalysis of a public-key cryptosystem based on approximations by rational numbers. In I. B. Damgård, editor, *Advances in Cryptology—EUROCRYPT'90*, pages 313–317. Springer, 1990.

[67] G. S. Vernam. Secret signaling system, July 22 1919. US Patent 1,310,719.

[68] J. Von Zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

[69] J. Walker. Hotbits: Genuine random numbers, generated by radioactive decay. http://www.fourmilab.ch/hotbits/, September 2006.

[70] L. Wang and Y. Zhu. $f[x]$-lattice basis reduction algorithm and multisequence synthesis. *Science in China Series: Information Sciences*, 44(5):321–328, 2001.

[71] M. Yang, D. Lin, and X. G. Generalized Fourier transform and the joint N-adic complexity of a multisequence. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 97(9), 2014.

[72] L. Zhao and Q. Wen. On the joint 2-adic complexity of binary multisequences. *RAIRO-Theoretical Informatics and Applications*, 46(03):401–412, 2012.

WEIHUA LIU

EDUCATION

Graduate Certificate in College Teaching and Learning December 2014

University of Kentucky, Lexington, KY

B.S. in Applied Mathematics June 2008

Southwest Jiaotong University, Chengdu, China

TEACHING EXPERIENCE

Primary instructor Summer, 2015

Course Title: Discrete Mathematics (CS 275)

- Developed the course syllabus, lecture notes, classroom activities, assignments and tests independently.
- Lectured and kept classroom activities five days a week during eight weeks.

Primary instructor Spring, 2014

Course Title: Introduction to Computers (CS101)

- Participated in developing the course (syllabus, assignments, lecture slides, demonstrations and projects).
- Lectured three hours a week and tutored lab sessions.

Lab instructor Fall, 2013; Fall, 2015

Course Title: Introduction to Computers (CS101)

- Lectured 5 lab sessions and tutored the students on lab projects.
- Participated in assigning homework, graded exams, proctored exams, and maintained regular office hours.

Teaching Assistant Spring, 2013

Course Title: Algorithm Design and Analysis (CS315), Systems Programming (CS 485)

- Graded homework submissions, projects and exam papers.
- Tutored the students and maintained regular office hours.

Course Title: Introduction to Cryptology (CS378)

- Lectured during the professor's absences.

RESEARCH EXPERIENCE

Graduate Research Assistant August 2010 till now

University of Kentucky

- Conducted research on cryptography, stream ciphers, and secure nonlinear functions.
- Developed efficient algorithms for attacking pseudorandom sequences generators.
- Participated in writing an NSF grant proposal.

Graduate Research Assistant August 2008 to July 2010

Southwest Jiaotong University

- Explored public key cryptography and lattice theory.

Research Intern

Summer, 2007

Hwadee Information Technology Co., Ltd, Chengdu, China

- Developed a hotel management System and a library management system.

Undergraduate Research Assistant 2007

Southwest Jiaotong University

- Designed questionnaires and investigated the factors that influence high-school students' higher education choice.
- Analyzed big data with different statistical methods, such as factor analysis and regressions.
- The work was awarded the "Yanghua Cup" prize and published in an academic journal of the university.

China Undergraduate Mathematical Contest in Modeling 2007

- Established a statistical model to study the relations between the population growth rate and the fertility, mortality, and migration rate.
- Modified the Leslie growth model to predict the population growth in the next 50 years.

PUBLICATIONS

- **Weihua Liu**, and Andrew Klapper. "A Lattice Rational Approximation Algorithm for AFSRs Over Quadratic Integer Rings." Sequences and Their Applications-SETA 2014. Springer International Publishing, 2014. 200-211.
- **Weihua Liu**, and Andrew Klapper. "AFSRs synthesis with Extended Euclidean Rational Approximation Algorithm." Accepted. Advances in Mathematics of Communications. 2015

PRESENTATIONS AND TALKS

- Sequences and Their Applications-SETA 2014, Melbourne, Australia, November 2014.
- The 12th International Conference on Finite Fields and Their Applications, Saratoga Springs, New York, July 2015.
- Invited Speaker, University of Tennessee at Martin, October 2015.
- Keep Current Seminar, Department of Computer Science, University of Kentucky, September 2014.
- Crypto Seminar, Department of Computer Science, University of Kentucky, multiple times.

AWARDS

- MIC Networking Fellowship, Department of Computer Science, University of Kentucky, 2012.
- Conference and Research Student Support Funding, University of Kentucky, 2014.
- Student travel support from Computing Research Association-Women (CRA-W) to attend Grad Cohort Workshop, 2012.
- National scholarship for graduate students, Southwest Jiaotong University, 2009.
- Graduation with Honor, Southwest Jiaotong University, 2008.

- "Yanghua Cup", Southwest Jiaotong University, 2007.
- National fellowship, Southwest Jiaotong University, 2006.

SERVICE AND OUTREACH

- Invited Reviewer, Cryptography and Communications - Discrete Structures, Boolean Functions and Sequences, 2014.
- Microteaching Group Leader, University of Kentucky TA Orientation, August 2015.
- Volunteer, University of Kentucky Engineering Day, February 2014.
- Volunteer Teacher, GEMS program (Girls Enjoy Math and Science), November 2013.
- Volunteer, Leestown Middle School's Science Night, Lexington, KY, March 2013.
- Volunteer, Harrison Elementary School Science Night, Lexington, KY, March 2013.
- Member of UK Women's Choir (2013-2015), The ACDA National Convention, Dallas, Texas, 2013.