2000

Stream Cipher Analysis Based on FCSRs

Jinzhong Xu
University of Kentucky, jxu@accesstech.com

ABSTRACT OF DISSERTATION

Jinzhong Xu

The Graduate School
University of Kentucky
2000

STREAM CIPHER ANALYSIS BASED ON FCSRS

ABSTRACT OF DISSERTATION

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

By

Jinzhong Xu

Lexington, Kentucky

Director: Dr. Andrew Klapper, Professor of Computer Science

Lexington, Kentucky

2000

ABSTRACT OF DISSERTATION

STREAM CIPHER ANALYSIS BASED ON FCSRS

Cryptosystems are used to provide security in communications and data transmissions. Stream ciphers are private key systems that are often used to transform large volumn data. In order to have security, key streams used in stream ciphers must be fully analyzed so that they do not contain specific patterns, statistical infomation and structures with which attackers are able to quickly recover the entire key streams and then break down the systems.

Based on different schemes to generate sequences and different ways to represent them, there are a variety of stream cipher analyses. The most important one is the linear analysis based on linear feedback shift registers (LFSRs) which have been extensively studied since the 1960's. Every sequence over a finite field has a well defined linear complexity. If a sequence has small linear complexity, it can be efficiently recoverd by Berlekamp-Messay algorithm. Therefore, key streams must have large linear complexities. A lot of work have been done to generate and analyze sequences that have large linear complexities. In the early 1990's, Klapper and Goresky discovered feedback with carry shift registers over $Z/(p)$ (p-FCSRS), p is prime. Based on p-FCSRs, they developed a stream cipher analysis that has similar properties to linear analysis. For instance, every sequence over $Z/(p)$ has a well defined p-adic complexity and key streams of small p-adic complexity are not secure for use in stream ciphers.

This disstation focuses on stream cipher analysis based on feedback with carry shift registers. The first objective is to develop a stream cipher analysis based on feedback with carry shift registers over $Z/(N)$ (N-FCSRs), N is any integer greater than 1, not necessary prime. The core of the analysis is a new rational approximation

algorithm that can be used to efficiently compute rational representations of eventually periodic N-adic sequences. This algorithm is different from that used in p-adic sequence analysis which was given by Klapper and Goresky. Their algorithm is a modification of De Weger's rational approximation algorithm.

The second objective is to generalize feedback with carry shift register architecture to more general algebraic settings which are called algebraic feedback shift registers (AFSRs). By using algebraic operations and structures on certain rings, we are able to not only construct feedback with carry shift registers, but also develop rational approximation algorithms which create new analyses of stream ciphers.

The cryptographic implication of the current work is that any sequences used in stream ciphers must have large N-adic complexities and large AFSR-based complexities as well as large linear complexities.

Jinzhong Xu

Date

STREAM CIPHER ANALYSIS BASED ON FCSRS

by

Jinzhong Xu

Director of Dissertation

Director of Graduate Studies

Date

RULES FOR THE USE OF DISSERTATION

Unpublished dissertations submitted for the Doctor's degrees and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the thesis in whole or in part requires also the consent of the Dean of The Graduate School of the University of Kentucky.

A library which borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name and Address	Date

DISSERTATION

Jinzhong Xu

The Graduate School
University of Kentucky
2000

STREAM CIPHER ANALYSIS BASED ON FCSRS

DISSERTATION

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

By

Jinzhong Xu

Lexington, Kentucky

Director: Dr. Andrew Klapper, Professor of Computer Science

Lexington, Kentucky

2000

Table of Contents

List of Tables

List of Figures

Introduction

As the world enters the information era, securely protecting digital data has become an urgent and serious problem. For instance, with the fast development of internet technology, more and more businesses are operated electronically through the information highway. Every moment countless data streams such as bank transactions, credit card numbers, database queries, and confidential documents are transmitted online. On the other hand, computer hackers sneak into networks and eavesdroppers tap lines and scan digital signals. They threaten the security and privacy of business institutions as well as individuals. Therefore, extensive study must be made and advanced technology must be developed to provide secure communication systems. This is what cryptography does.

One way of protecting communication is to use a private key to transform (encrypt) meaningful message data (plain texts) into unreadable texts (cipher texts), and then send the ciphertexts through public channels. The message will be transformed back (deciphered) on the receiving site with the same key which is only shared by the sender and receiver. The following diagram represents such a scheme.

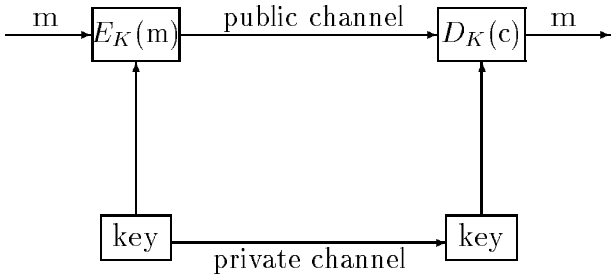

Figure 1: A Private Key System

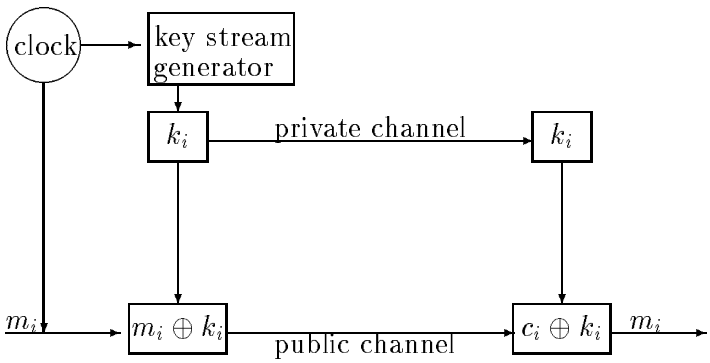

Figure 2: An One-Time-Pad System

There are many proposed and implemented cryptosystems in various applications. Different systems have different features and design goals. When transferring very large volume data, speed is an important concern. In this case, a class of private key cryptosystems called *stream ciphers* are often employed [31, 35]. Stream ciphers divide the plain text into characters and encrypt each character by adding time-varying noise. Hence the same plain-text character corresponds to different cipher-text characters at different times.

One of the most remarkable stream ciphers is the so called *one-time-pad* [39]. Assume that the data to be sent is binary. Bob and Alice securely share a purely random binary sequence (or key stream). When Bob wants to send data to Alice, he uses the next unused bits of the key stream to encipher the data and then sends the resulting cipher text to Alice. The encryption is done by doing an XOR operation $(0+0=0, 1+0=0+1=1, 1+1=0)$ between the data stream and the key stream.

Upon receiving the cipher text, Alice uses the same key stream bits to do an XOR operation with the received text. This will recover the message from the cipher text. Both Bob and Alice never reuse key stream bits. (This system was used in World War II with the key stream written on a pad of paper. Each sheet was used for one day then discarded, hence "one-time-pad"). The one-time-pad system is extremely fast due to its simplicity. The scheme can be depicted in Figire 2.

An attack on a system is a way to find the key without direct knowledge of

the key. For example, the cipher text only attack assumes that cryptanalysts only know the scheme and the cipher text. The security of a system depends on the 'difficulty' of successful attacks. For any system, besides its correctness, one of the most overwhelming concerns is its security. Therefore, before a cryptosystem can be used, a complete analysis against potential attacks must be made [2, 11, 12, 35]. Cryptology is the systematic study of the design and evaluation of cryptosystems. It defines security measures and provides mathematical models and analytical tools.

A cryptosystem is said to have *perfect secrecy* provided that based only on the cipher text, a cryptanalyst has no better attack than guessing, no matter how much computational power is available. The most distinguishing feature of the one-time-pad is that the system has perfect secrecy, and this feature is realized by the purely random property of the key stream [2, 35, 39]. Hence in order to use the one-time-pad, we must have: (1) a means for generating long purely random sequences; and (2) a secure channel to distribute the key stream. Since the key stream size is the same as the data stream size, securely distributing the key stream is as hard as securely sending the data stream itself. Also, we do not have an efficient and reliable device which can actually generate purely (uniformly) random sequences. Therefore the one-time-pad is not practical in most cases.

In practice, the purely random key stream generator is often replaced by a finite state machine. This makes the hardware implementation and the secure distribution of a key stream feasible because only a small size of data (key) is needed to initialize such a generator. However, good statistical properties of the key stream are essential in producing highly confusing cipher text and thus providing security. Hence stream cipher designers are looking for devices which can efficiently generate large period sequences that satisfy various statistical criterion. Also, since the key stream generator is a deterministic machine, designers must not allow attackers to recover the generator by analyzing the cipher text or a part of the key stream. Therefore, the analysis of key streams becomes very important in stream cipher design and application [19, 35].

The present dissertation focuses on stream cipher analysis based on *feedback with carry shift registers*. In Chapter 1 we give a brief review of *linear feedback shift registers* (LFSRs). LFSRs are feedback shift registers with no carry and all operations

conducted in a finite field such as $Z/(p)$ with p a prime number. Since the 1960's, LFSRs have been used as building blocks in key stream designs [35]. Because the Berlekamp-Messay algorithm can efficiently recover a generator of a sequence if its linear complexity is small [32], designers must analyze both the linear complexity and the statistical properties of the generated sequences. This raises a question whether there are other families of generators that have features similar to those of LFSRs and provide different complexities and thus different analysis. The effort to find such generators had not been greatly successful until the early 90's when Klapper and Goresky discovered feedback with carry shift registers over $Z/(2)$ (2-adic FCSRs) [20, 21, 22, 23].

In Chapter 2 we introduce feedback with carry shift registers over $Z/(N)$ for any positive integer $N > 1$ (N-FCSRs), and then in Chapter 3 develop the full analysis of N-adic sequences based on these registers. We use N-adic numbers as a mathematical model to represent N-adic sequences. Since the algebraic structure of general N-adic numbers is much weaker than that of p-adic numbers, the p-adic analytical tools and lattice theory cannot be easily utilized to design a synthesis algorithm for N-adic periodic sequences. The main contribution of this dissertation is the design of a new algorithm which can efficiently synthesize any N-adic sequence. This algorithm is different from that for 2-adic sequences which was given by Klapper and Goresky by modifying De Weger's rational approximation algorithm for p-adic numbers [10]. Also, we will show that if an N-adic sequence of length l is randomly picked, the probability of having its N-adic complexity at least $l/2$ is not less than $1/2$.

Note that an analysis of key streams based on a class of generators generally consists of three phases: (1) Describe the register architecture and implementation; (2) Discuss the properties of output sequences; and (3) Design algorithms to solve the register synthesis problem, i.e., given a part of a key stream, find the smallest size register that generates the entire key stream. Once such an algorithm is developed, an associated complexity can be defined for sequences, and then all sequences having relatively small such complexity must not be used in any stream cipher schemes.

Another aim of this dissertation is to construct more general feedback with carry shift registers and develop related analyses of stream ciphers. In Chapter 4, based on algebraic operations on certain rings, we introduce algebraic feedback with carry shift registers (AFSRs). Although the description of AFSRs is abstract, there are several

interesting cases in which registers can be practically implemented, for instance, function fields, quadratic number fields and ramification of N-FCSRs. In this chapter, we first describe a framework for a general rational approximation algorithm, and show that the algorithm converges efficiently when some specific conditions are satisfied. We then show these specific conditions can be realized in many useful instances such as quadratic number fields and ramification of N-FCSRs.

As a summary, this dissertation is a study of stream cipher analysis based on feedback with carry shift registers. The results are mainly divided into two parts: one for N-FCSR based analysis, and the other for AFSR based analysis. In both cases, register synthesis algorithms are created and then the corresponding analysis is developed. Examples and experimental data are also provided to support both the theoretical results and algorithm implementations. The cryptographic implication of the current work is that any sequences used in stream ciphers must have large complexities as defined in this dissertation.

Chapter 1

Linear Feedback Shift Registers

Linear feedback shift registers (LFSRs) have been widely used in various areas such as cryptography and coding theory. The most important characteristics of LFSRs are: (1) they are simple and thus fast; (2) the statistical properties of the output sequences can be fully analyzed by using efficient algebraic tools; (3) the Berlekamp-Massey algorithm efficiently solves the register synthesis problem. There are many fast devices that use LFSRs as building blocks and generate sequences whose linear complexities are provably large and whose statistical properties are good. Hence LFSRs are fundamental in both theoretical research and real applications. In this chapter we give a brief review of the main properties of LFSRs.

1.1 Register Construction and Output Sequences

Let $F = GF(p^m)$ be a Galois field. For any integer $r > 0$ and r fixed elements $\{ q_i \in F : 1 \leq i \leq r \}$ (called taps), an LFSR of length r consists of r cells with initial contents $\{ a_i \in F : 0 \leq i \leq r-1 \}$. For any $n \geq r$, if the current state is $(a_{n-1}, \cdots, a_{n-r})$, then a_n is determined by the linear recurrence relation

$$a_n = -\sum_{i=1}^{r} a_{n-i} q_i \ .$$

The device outputs the rightmost element a_{n-r}, shifts all the cells one unit right, and feeds a_n back to the leftmost cell. That is, the state change is given by

$$(a_{n-1}, a_{n-2}, \cdots, a_{n-r}) \longrightarrow (a_n, a_{n-1}, \cdots, a_{n-r+1}) \ .$$

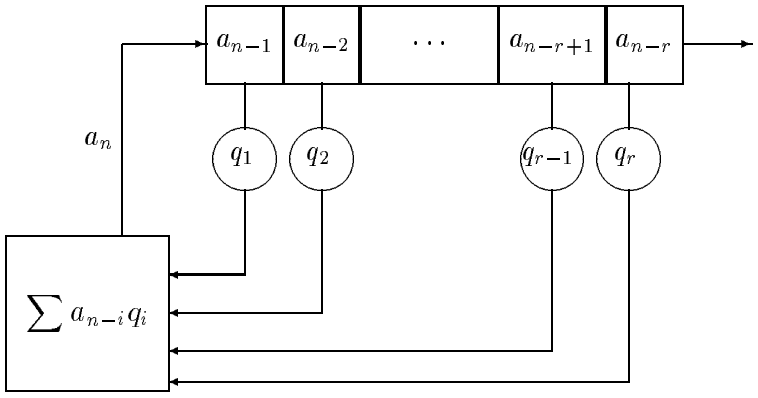

Figure 1.1: A Linear Feedback Shift Register

Here the summation is taken as addition in the field F. The architecture of a length r register is depicted in Figure 1.1.

Any configuration of the r cells forms a state of the LFSR. If $q_r \neq 0$, the following polynomial $q(x) \in F[x]$ of degree r is useful in the analysis of LFSRs:

$$q(x) = q_0 + q_1 x + q_2 x^2 + \cdots + q_r x^r \quad \text{with} \quad q_0 = -1 \quad .$$

This polynomial is called the *connection polynomial*. An infinite sequence $A = \{a_i \in F, i \geq 0\}$ has period T if for any $i \geq 0$, $a_{i+T} = a_i$. Such a sequence is called *periodic*. If this is only true for i greater than some index i_0, then the sequence is called *eventually periodic*. For an LFSR of length r over F, we have the following facts (see [35]).

1. There are only finitely many possible states, and the state with all the cells zero will produce a 0-sequence. The output sequence is eventually periodic and the maximal possible period is $p^{mr} - 1$.

2. The power series $g_s(x) = \sum_{i=0}^{\infty} a_i x^i$ associated with the output sequence is called the *generating function* of the sequence. It is a rational function over F in the form,

$$g_S(x) = \frac{p(x)}{q(x)}$$

with $deg(p(x)) < r$. The output sequence is strictly periodic if and only if $deg(p(x)) < deg(q(x))$.

3. There is a one-to-one correspondence between LFSRs of length r with $q_r \neq 0$ and rational functions $p(x)/q(x)$ with $deg(q(x)) = r$ and $deg(p(x)) < r$.

4. The period of the output sequence is the smallest T such that $q(x)|(x^T - 1)$ or, equivalently, the order of x in the multiplicative group $(F[x]/(q(x)))^*$.

Note that for the output sequence $\{ a_i \}$ of an LFSR of length r and any $n \geq r$, there is a relation

$$\sum_{i=0}^{r} a_{n-i} q_i = 0 \ .$$

Such a relation is called *a linear recurrence relation* of length r, and may be represented by the characteristic polynomial

$$c(x) = q_0 x^r + q_1 x^{r-1} + \cdots + q_{r-1} x + q_r$$

which is reciprocal to the polynomial $q(x)$. In general, a sequence A over F may have many linear recurrence relations. The one having smallest length is called a minimal recurrence relation. It is unique up to a constant multiple. The corresponding characteristic polynomial with the leading coefficient 1 is called the *minimal polynomial*, denoted by $m_A(x)$. Therefore, for a given sequence A over F, finding a shortest LFSR that generates A is equivalent to finding the minimal polynomial.

Definition 1.1.1 *The linear complexity (span) of a sequence A over F is the degree of its minimal polynomial, denoted by $\lambda(A)$.*

The best known efficient way to find the minimal polynomial of a sequence is the Berlekamp-Massey algorithm (B-M algorithm for short). This algorithm is not just important in cryptanalysis, but also in coding theory. In fact, it was originally designed as a decoding algorithm for BCH codes [3, 32]. The following is the conclusion of the B-M algorithm.

Theorem 1.1.1 *For a sequence A over F with linear complexity λ, the B-M algorithm can compute its minimal polynomial by processing at most 2λ consecutive bits. The time complexity of the algorithm is quadratic, i.e., $O(\lambda^2)$.*

From the point of view of cryptography, the B-M algorithm provides an efficient attack on stream ciphers if the key streams have relatively small linear complexity.

In other words, the linear complexity of sequences is an important security measure. Any useful key stream generators must be secure against the B-M attack. In this dissertation, we will point out that this alone is not secure enough because there exist other attacks that are as efficient as the B-M algorithm, but whose corresponding security measures are radically different.

1.2 Psuedo-Random Generators and m-Sequences

LFSRs are deterministic devices and the sequences they generate cannot be purely random. However, by properly setting an LFSR, we can get an output sequence which looks like a random sequence in certain regards. For instance, the occurrence of symbols is nearly balanced and patterns of strings are distributed uniformally. A sequence generator is called *pseudo-random* if the generated sequence satisfies certain statistical criteria. There are many suggested such criteria and several proposed by Golomb have been widely accepted in the cryptographic community [14]. To gain maximal security in stream ciphers, key streams must be highly unpredictable and the substring patterns cannot be used to accumulate statistics and then to utilize an efficient attack.

Statistical properties of sequences generated by LFSRs can be characterized by their minimal polynomials. Recall that an element α in a finite field F is *primitive* if every non-zero element in F is a power of α, i.e., α^i for some $i \geq 0$. An irreducible monic polynomial in $F[x]$ is called *primitive* if it has a root which is primitive in the extension field it generates.

Definition 1.2.1 *A sequence A is called an m-sequence if its minimal polynomial $m_A(x)$ is irreducible and it has a primitive root.*

Suppose the minimal polynomial $m_A(x)$ of an m-sequnce has degree r. Then the m-sequence can be generated by an LFSR of length r. Hence the period of the m-sequence reaches the maximal possible value $p^{mr} - 1$ for registers of length r. Let $t = p^m$ and E be the extension field of $F = GF(t)$, formed by adjoining a root of the minimal polynomial. For any element $\beta \in E$, the *trace function* is defined as

$$Tr(\beta) = \beta + \beta^t + \cdots + \beta^{t^{r-1}} .$$

By Galois theory, $Tr(\beta) \in F$ and the trace function is linear [17]. The following theorem is often called the trace representation of m-sequences [35].

Theorem 1.2.1 *Let $m(x)$ be a primitive polynomial of degree r and $\alpha \in E$ a primitive root of $m(x)$. Then for every sequence $A = \{a_i : i \geq 0\}$ in F with minimal polynomial $m(x)$, there is an element $B \in E$ such that $a_i = Tr(B\alpha^i),\ i \geq 0$.*

For the binary case, i.e., $F = \{0, 1\}$, some statistical properties of m-sequences can be stated more precisely.

Theorem 1.2.2 *Let the hypotheses be the same as in Theorem 1.2.1, but the characteristic of the field be 2. Suppose $A = \{a_i\}$ is an m-sequence. Then (1) in one period, the number of $0's$ is $2^{r-1} - 1$ and number of $1's$ is 2^{r-1}; (2) for any s, $1 \leq s \leq r$ and any subsequence B of length s, the number of occurences of B in one period of A is $2^{r-s} - 1$ if $B \neq \bar{0}$; 2^{r-s} otherwise.*

Because of the Berlekamp-Massey algorithm, the linear complexity is an important security measure of a binary sequence. Note that an m-sequence has nice statistical properties, but low linear complexity. We are interested in sequences that have good statistical properties and large linear complexities. Here is an interesting question: if A is a random binary sequence of length n, what is the probability that its linear complexity is greater than $n/2$? In [35], Rueppel carefully investigated the linear complexity profile of binary sequences. Here we state one of his results.

Theorem 1.2.3 *Let $N_n(L)$ be number of binary sequences of period n having linear complexity exactly L. If $n \geq L > 0$, then*

$$N_n(L) = 2^{\min(2n-2L, 2L-1)} .$$

If $n > L = 0$, then $N_n(L) = 1$.

This result tells us that the vast majority of the possible binary sequences of length n have linear complexity close to $n/2$. However, high linear complexity alone is not a sufficient security criterion. As pointed out by Rueppel in [35], the complexities of all the prefixes of a given sequence (*complexity profile*) are more important. In general, complexity profiles are harder to analyze.

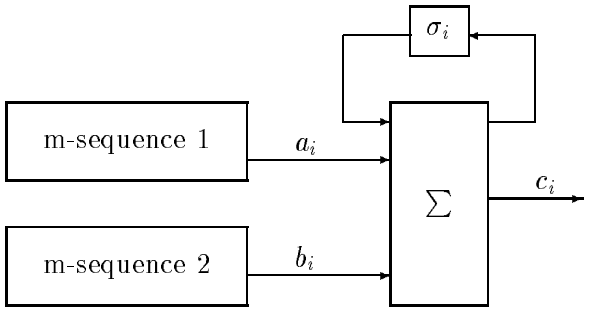

Figure 1.2: A Summation Cipher

By using LFSRs as building blocks, various types of key stream generators have been proposed. One such type is the so-called *nonlinear combiner*. It takes several m-sequences as inputs, combines them by a nonlinear function (*filter*), and then outputs a sequence. The desire is that (1) if the input m-sequences have good statistical properties, they will be inherited by the output sequence; (2) if the input m-sequences have small linear complexities, the nonlinear combining function will increase the linear complexity significantly. One example is *the summation cipher* which was proposed by Rueppel [35]. A simple case is depicted in Figure 1.2.

Here the summation is taken as integers and σ_i is the carry bit. Note that integer addition is a highly non-linear operation when considered in $F = \{0, 1\}$. The following result ([35] Property 9.3) characterizes the output sequence of the summation cipher.

Proposition 1.2.4 *Let A and B be two binary m-sequences whose primitive minimal polynomials have relatively prime degrees r_1 and r_2. If A and B are added over the reals, then the sum sequence C has linear complexity close to its period length, i.e., $\lambda(C) \leq (2^{r_1} - 1)(2^{r_2} - 1)$ with near equality.*

1.3 An Extension of LFSRs

The register architecture of LFSRs is still valid when F is replaced by any ring. In particular, we can take $R = Z/(N)$ for any positive $N > 1$. We call these registers N-LFSRs.

N-LFSRs have applications as random number generators and for codes defined over the integers modulo N ([4, 5, 37]). Compared with LFSRs, the biggest difference is that the ring $Z/(N)$ may have non-trivial zero divisors. This adds complexity to the analysis of the registers and the decoding procedures. Several authors have investigated N-LFSRs from different point of views. The most important problems are to find a proper algebraic tool for analyzing output sequences, and to design an efficient synthesis algorithm. Fortunately, since N can be factored into a product of distinct prime powers, by the Chinese remainder theorem the problem can be reduced to the case $Z/(p^d)$. In [34], Reeds and Sloane successfully extended the Berlekamp-Massey algorithm to N-LFSRs by an elegant modification of the original procedure. Dai and Qi in [33] successfully established the trace representation by using p-adic number fields. For a periodic sequence A over $Z/(N)$, the linear complexity of A is defined to be the smallest length of N-LFSRs that generate A.

Chapter 2

Feedback with Carry Shift Registers over $Z/(N)$

As described in the previous chapter, the algebraic tools associated with LFSRs provide a systematic analysis of sequences. This analysis leads to a variety of applications based on LFSRs. Since 1955, effort has been directed towards the study of other ("non-linear") feedback architectures which would give rise to fundamentally new or different kinds of pseudorandom sequence generators and analyses [8, 13, 14, 16, 35, 41, 42]. However, the LFSR-based analysis had been the only general purpose tool until the early 90's when Klapper and Goresky discovered *feedback with carry shift registers* (FCSRs). Their work focused on FCSRs that generate sequences over $Z/(p)$, where p is a prime number. They proved that FCSRs share many important properties with LFSRs. In particular, by modifying de Weger's rational approximation algorithm [10], they designed an efficient algorithm to solve the register synthesis problem. Furthermore, their algorithm leads to an efficient attack on the summation cipher [20, 21, 22, 23]. In this chapter, we first describe feedback with carry shift registers over $Z/(N)$ for any integer $N > 1$ (N-FCSRs), and then present the basic properties of the defined registers. Although most of the properties are parallel to those described in [23] for the binary case (2-FCSRs), we present some new properties of general N-FCSRs. For example, we show that there are many N-adic sequences which can be generated by small N-FCSRs, but cannot be generated by small N-LFSRs. We use N-adic numbers as algebraic tools to analyze sequences generated by N-FCSRs. For a composite N, the algebraic structure of N-adic numbers is much weaker than that of p-adic numbers [26]. This weakness hinders the generalization

of the synthesis algorithm of p-FCSRs. We investigate register synthesis in the next chapter.

2.1 Register Description and Examples

Let $N > 1$ be an integer and $S = \{a : 0 \leq a \leq N-1\}$. For any integer $r \geq 1$, the state of a *feedback with carry shift register* over $Z/(N)$ consists of r integers $a_0, a_1, a_2 \cdots, a_{r-1} \in S$ and an arbitrary integer $M = M_{r-1}$, the *memory*. The state change function is determined by $r+1$ integers $d, q_1, q_2, \cdots, q_r \in S$ such that $gcd(d, N) = 1$ and $q_r \neq 0$ as follows:

Step 1: Compute the integer sum,

$$\sigma = M_{r-1} + a_{r-1}q_1 + a_{r-2}q_2 + \cdots + a_0 q_r$$

Step 2: Compute $a_r \in S$, $M_r \in Z$ such that

$$\sigma = da_r + M_r N$$

Step 3: Change the memory M_{r-1} to M_r.

Step 4: Output a_0 and use a_r to shift the register loading cells, replacing

$$(a_{r-1}, \cdots, a_0) \quad \text{by} \quad (a_r, \cdots, a_1).$$

Here is how a_r and M_r in Step 2 can be computed. First, divide σ by N and get a non-negative remainder b_r, i.e., $\sigma = b_r + kN$ for some $k \in Z$. Let w be the inverse of d modular N, i.e., $0 < w < N$ and $wd = 1 + lN$ for some $l \in Z$. Since w and b_r are in S, there are $a_r, h \in S$ such that $wb_r = a_r + hN$. It follows that $da_r = dwb_r - dhN = (1 + lN)b_r - dhN = b_r + (l - dh)N$, and $\sigma = b_r + kN = da_r - (l - dh)N + kN = da_r + M_r N$, where $M_r = k - l + d$. If d is 1, we simply have $a_r = b_r$ and $M_r = k$. An N-FCSR is depicted in Figure 2.1

As seen, an N-FCSR is a simple device and is similar to an LFSR. The most computationally costly operations are integer multiplication and division in Step 2, but they can be implemented efficiently for small N. An N-FCSR outputs a sequence $A = \{a_0, a_1, a_2, \cdots, \}$ over S by infinitely iterating the state change. The first r output symbols are the initial register configuration. For $n \geq r$, a_n is determined by

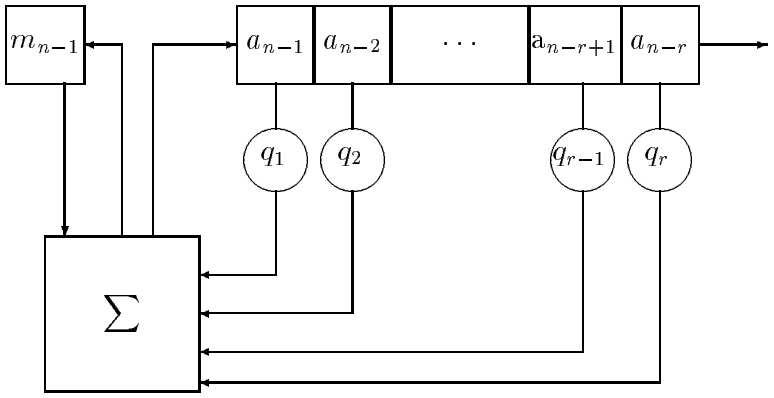

Figure 2.1: An N-FCSR Architecture

both the memory and the current register cells. In the whole processing, the coefficients $\{d, q_1, q_2, \cdots, q_r\}$ (called *taps*) are fixed. To explore their importance, we call the following integer

$$q = -d + q_1 N + q_2 N^2 + \cdots + q_r N^r$$

the *connection number*. For convenience, let $q_0 = -d$, so $q = \sum_{i=0}^{r} q_i N^i$. We will see that this number acts similarly to the connection polynomial in the analysis of LFSR ([35] or Chapter 1). The connection numbers with $q_0 = -1$ are special, and make the register state change operation and implementation easier. Let us first look at an example.

Example 1: Set $N = 6$, $S = \{0, 1, 2, 3, 4, 5\}$. Choose $r = 3$, $(a_0, a_1, a_2) = (1, 2, 3)$, $(d, q_1, q_2, q_3) = (1, 1, 1, 5)$, and the initial memory $M = 0$. Then the connection number $q = -1 + 6 + 36 + 5 * 6^3 = 1121$. Table 2.1 displays the states and the outputs of first 15 iterations.

In 1991, Marsaglia and Zaman [30] suggested and statistically analyzed a type of pseudo-random number generators over $Z/(N)$. The proposed generators use a truncated sum or difference of two previous bits. For instance, one of the Marsaglia-Zaman schemes is the so called *add-with-carry*. For $0 < s < r$, $a_i \in S$, assume $a_0, a_1, \cdots, a_{r-1}$ and c_{r-1} are initialized with $a_i \in S$ and $c_{r-1} \in \{0, 1\}$. Then for $n \geq r$, a_n is determined by

$$a_n = a_{n-s} + a_{n-r} + c_{n-1} \bmod N$$

Table 2.1: Output of a 6-FCSR

memory	q_1 1	q_2 1	q_3 5	output
0	1	2	3	3
3	0	1	2	2
2	2	0	1	1
1	3	2	0	0
1	0	3	2	2
2	2	0	3	3
3	1	2	0	0
1	0	1	2	2
2	0	0	1	1
1	1	0	0	0
0	2	1	0	0
0	3	2	1	1
1	4	3	2	2
3	0	4	3	3
3	4	0	4	4
4	3	4	0	0

and

$$c_n = \lfloor \frac{a_{n-s} + a_{n-r} + c_{n-1}}{N} \rfloor \ .$$

In particular, when $s = 1$ and $r = 2$, the generator operates like a Fibonacci sequence. We see that these generators are nothing but the special cases of N-FCSRs with all the taps $q_i = 0$ except for $q_r = q_s = 1$. In 1998 Bach cryptographically analyzed the Marsaglia-Zaman's generators and developed an algorithm (see [1]) to recover the generator. Compared with the Bach's results, our analysis of N-FCSR and N-adic sequences will be different, more general and more comprehensive.

2.2 Characteristics of N-FCSRs

In this section we study characteristics of output sequences of an N-FCSR. Most of them are generalizations of those described by Klapper and Goresky for 2-FCSRs [23]. We first show that the output sequence is eventually periodic. By the register construction, this amounts to showing that the memory is bounded.

Proposition 2.2.1 *For an N-FCSR, let w be the Hamming weight of $q - q_0$ with respect to N, the number of non-zero q_i among $\{q_1, \cdots, q_r\}$. Suppose that at the state with index $n \geq r$, the register has the memory $M = M_n$. Then the following hold.*

(1) If $|M_{n-1}| \leq w(N-1)$, then $|M_n| \leq w(N-1)$;

(2) If $|M_{n-1}| > w(N-1)$, then $|M_n| \leq |M_{n-1}| - 1$.

Proof: Without loss of generality, we may assume the current state is $(a_{r-1}, a_{r-2}, \cdots, a_0)$, i.e., $n = r$. From the register construction we have,

$$\sigma = \sum_{i=0}^{r-1} a_i q_{r-1} + M_{r-1} = da_r + M_r N \ .$$

Let $v = \sum_{i=0}^{r-1} a_i q_{r-i}$. Then $v \geq 0$.

Suppose $|M_{r-1}| \leq w(N-1)$. Note that $a_i, q_i \leq (N-1), 0 \leq i \leq r-1$. If $\sigma > 0$, then $|\sigma - da_r| \leq max(\sigma, da_r) \leq max\{w(N-1)^2 + w(N-1), (N-1)^2\} = w(N-1)N$. Hence $|M_r| = |\sigma - da_r|/N \leq w(N-1)N/N = w(N-1)$. If $\sigma < 0$, then $|\sigma| = |v + M_{r-1}| \leq |M_{r-1}| \leq w(N-1)$. Therefore $|M_r| = |\sigma - da_r|/N \leq (|\sigma| + da_r)/N \leq (w(N-1) + (N-1)^2)/N \leq w(N-1)N/N = w(N-1)$.

Now assume $|M_{r-1}| \geq w(N-1) + 1$. We then have two cases: (1) $|M_{r-1}| = w(N-1) + 1$; (2) $|M_{r-1}| \geq w(N-1) + 2$. For Case 1, if $\sigma > 0$, then $|\sigma - da_r| \leq max\{\sigma, da_r\} \leq max\{w(N-1)^2 + w(N-1) + 1, (N-1)^2\} \leq w(N-1)N + 1$. Hence $|M_r| = |\sigma - da_r|/N \leq (w(N-1)N + 1)/N$. Since $|M_r|$ is an integer, it follows that $|M_r| \leq w(N-1) \leq |M_{r-1}| - 1$. If $\sigma < 0$, then $|\sigma| = |v + M_{r-1}| \leq |M_{r-1}| = w(N-1) + 1$. Hence $|M_r| = |\sigma - da_r|/N \leq (|\sigma| + da_r)/N \leq (w(N-1) + 1 + w(N-1)^2)/N \leq (w(N-1)N + 1)/N = w(N-1) + 1/N$. Since $N > 1$ and $|M_r|$ is an integer, $|M_r| \leq w(N-1) \leq |M_{r-1}| - 1$.

For Case 2, we have that $(N-1)|M_{r-1}| \geq (N-1)(w(N-1)+2) \geq w(N-1)^2 + N$ since $N \geq 2$. This implies that $|\sigma| \leq w(N-1)^2 + |M_{r-1}| \leq (|M_{r-1}| - 1)N$. If $\sigma > 0$, we have $|\sigma - da_r| \leq max\{\sigma, da_r\} \leq max\{(|M_{r-1}| - 1)N, w(N-1)N\}$. It follows that $|M_r| = |\sigma - da_r|/N \leq max\{|M_{r-1}| - 1, w(N-1)\} = |M_{r-1}| - 1$. If $\sigma < 0$, we then have $|\sigma| \leq |M_{r-1}|$ and $|\sigma - da_r| \leq |M_{r-1}| + (N-1)^2 \leq w(N-1)^2 + |M_{r-1}| \leq (|M_{r-1}| - 1)N$. This implies that $|M_r| = |\sigma - da_r|/N \leq (|M_{r-1}| - 1)$. □

This Proposition shows that after a finite number of iterations, the absolute value of the memory is bounded by $w(N-1)$. In particular, after a finite number of state

changes, an N-FCSR will fall into a *periodic mode*, and then the register will generate the periodic part of the output sequence. Hence, when an N-FCSR enters a periodic mode, we may assume that the memory has its absolute value bounded by $w(N-1)$. This implies that the hardware memory required in implementing such an N-FCSR is fully determined by the parameters N, r and the Hamming weight $w \leq r$.

Since an N-FCSR is in effect a finite state machine, the output is an eventually periodic sequence and the period is determined by the number of distinct states which the register actually goes through. Since a state consists of the memory M with $|M| \leq w(N-1)$ and the register cells $0 \leq a_i \leq N-1$ and $0 \leq i \leq r-1$, a state is determined by r N-nary bits and a memory integer between $-w(N-1)$ and $w(N-1)$. Note that the state having all $a_i's$ and M zero causes the N-FCSR to output all 0's. Hence the total maximal number of states or maximal possible period of the output sequence is $\leq (2w(N-1)+1)N^r - 1$. This number may not be achievable by an N-FCSR. In fact, in the case when $q_0 = -1$ we can show that the memory will eventually fall into the range $[0, w(N-1)]$.

Proposition 2.2.2 *For an N-FCSR of length r with $d = 1$ (i.e., $q_0 = -1$) and initial memory M_{r-1}, after a finite number of iterations, the register has $0 \leq M_n \leq w(N-1)$.*

Proof: By Proposition 2.2.1, we only need to show that the memory $M_n \geq 0$ after a finite number of iterations. Note that if at a state with index n we have $M_n \geq 0$, then $\sigma = \sum_{i=1}^{r} a_{n-i} q_i + M_n \geq 0$. Hence, $\sigma = a_n + M_{n+1} N$ with $a_n \in S$ and $M_{n+1} \geq 0$.

Now we consider the case when $M_n < 0$. Suppose $M_n = -1$. Let $v_n = \sum_{i=1}^{r} a_{n-i} q_i$. If $v_n = 0$, then $\sigma = M_n = -1 = (N-1) + (-1)N$. So, $a_n = N-1$ and $M_{n+1} = -1$. Since q_i are not all zero, $v_n, v_{n+1}, \cdots$ cannot be all zero. Namely, there is an integer k such that $v_n = \cdots = v_{n+k} = 0$ and $M_n = M_{n+1} = \cdots = M_{n+k} = -1$, but $v_{n+k+1} > 0$. We then have $\sigma = M_{k+n} + v_{n+k+1} \geq 0$. This implies that $M_{n+k+1} \geq 0$.

Suppose $M_n = -2$, or, -3 (i.e. $|M_n| = 2$, or 3). From the equation $\sigma = M_n + v_n = a_{n+1} + M_{n+1} N$, it follows that $M_{n+1} \geq M_n + 1$. Suppose $M_n \leq -4$ (i.e., $|M_n| \geq 4$). We claim that $|M_{n+1}| \leq |M_n| - 1$, i.e., $M_{n+1} \geq M_n + 1$. Note that if $\sigma = M_n + v_n \geq 0$, then $M_{n+1} \geq 0$. Hence, we assume that $\sigma < 0$. Let $M_n = -b - kN$ with $b \in S$ and $k \geq 0$. If $v_n - b \geq 0$, then $\sigma = (v_n - b) - kN$ gives rise to $a_{n+1} = v_n - b$ and $M_{n+1} = -k$.

Table 2.2: Output of a 3-FCSR

memory	q_1	q_2	q_3	output
	1	0	1	
-1	2	1	2	2
1	0	2	1	1
0	1	0	2	2
1	0	1	0	0
-1	2	0	1	1
0	1	2	0	0
-1	2	1	2	2

If $v_n - b < 0$, then $\sigma = (N + v_n - b) - (k+1)N$ gives rise to $a_{n+1} = N + v_n - b$ and $M_{n+1} = -(k+1)$. We show that $k+1 \leq |M_n| - 1$, i.e., $(|M_n| - b)/N + 1 \leq |M_n| - 1$. It is equivalent to show that

$$(N-1)|M_n| \geq 2N - b \ .$$

For $N \geq 2$, it is easy to show that $2N/(N-1) \leq 4$. Hence, $2N/(N-1) \leq 4 \leq |M_n|$. It follows that

$$(N-1)|M_n| \geq 2N \geq 2N - b \ .$$

This completes the proof. □

For an N-FCSR of length r, the above proposition shows that if $d = 1$, the maximal possible period of the output sequence is less than or equal to $(w(N-1)+1)N^r - 1$. The following example shows that if an N-FCSR has $d \neq 1$, then the memory may switch between positive and negative as the register changes states.

Example 2: Set $N = 3$, $S = \{0, 1, 2\}$. Choose $r = 3$, $(a_0, a_1, a_2) = (2, 1, 2)$, $(d, q_1, q_2, q_3) = (2, 1, 0, 1)$, and the initial memory $M = -1$. Then the connection number $q = -2 + 3 + 27 = 28$. Table 2.2 displays the states and the outputs of the register in one period.

In this example, $q = 28$ and the Hamming weight of $q - d$ is 2. According to Proposition 2.2.1, the number of possible states is $(2w(N-1)+1)N^3 - 1 = 3^5 - 1 = 242$, but the period is only 6. Hence it seems hard to compute the exact period by counting the number of distinct states that the register goes through. We return to the computation of period of output sequences for N-FCSRs later, once we have established some algebraic machinery.

Recall that the algebraic tools used in the analysis of LFSRs are polynomials and formal power series over finite fields (see Chapter 1). For p-FCSRs (p prime), p-adic numbers are the algebraic tools employed in analysis (see Klapper and Goresky [23]). In their analysis, the assumption that p is prime is crucial. However for any $N > 1$ (prime or composite), the general theory of N-adic numbers still holds. See Koblitz's book [26] and Mahler's book [29] for reference.

An N-adic number is a sum of type $\sum_{i=0}^{\infty} a_i N^i, a_i \in S$. Any two N-adic numbers can be added and multiplied. Algebraically, the set of N-adic numbers $\widehat{Z/(N)}$ is a ring. Also, N-adic numbers are equipped with a topology with respect to which addition and multiplication are continuous and $\widehat{Z/(N)}$ is complete. In this topology, an element is near 0 if it is divisible by a high power of N. Note that $-1 = (N-1) + (N-1)N + (N-1)N^2 + \cdots$, and for every non-negative integer x there is a unique finite N-adic expansion

$$x = a_0 + a_1 N + a_2 N^2 + \cdots + a_t N^t$$

with $a_i \in S$ and $a_t \neq 0$. Hence all integers are N-adic numbers.

We now use N-adic numbers to represent output sequences of N-FCSRs. Let $A = \{a_0, a_1, \cdots, a_{r-1}, a_r, \cdots\}$ be an infinite sequence over S. We call the N-adic number

$$\alpha = \alpha(A, N) = \sum_{i=0}^{\infty} a_i N^i$$

the *generating number of* A, and A the *coefficient sequence* of α. We need the following lemma:

Lemma 2.2.3 *Let A and B be two eventually periodic N-adic sequences, and let $\alpha(A, N)$ and $\alpha(B, N)$ be the corresponding N-adic numbers. Then the sum: $\alpha(A, N) + \alpha(B, N)$ is an N-adic number whose coefficient sequence is eventually periodic.*

Proof: Suppose A has the prefix part $\{a_0, a_1, \cdots, a_{u-1}\}$ and the strictly periodic part $\{a_u, a_{u+1}, a_{u+2}, \cdots, a_{L+u-1}\}$, where L is the period. Similarly, suppose B has the prefix part $\{b_0, b_1, \cdots, b_{v-1}\}$ and the strictly periodic part $\{b_v, b_{v+1}, b_{v+2}, \cdots, b_{T+v-1}\}$, where T is the period. Note that any strictly periodic sequence can be generated by a *pure-cycling register*. Then A can be generated by combining a pure-cycling register and A's prefix sequence. This is also true for the sequence B. To get the

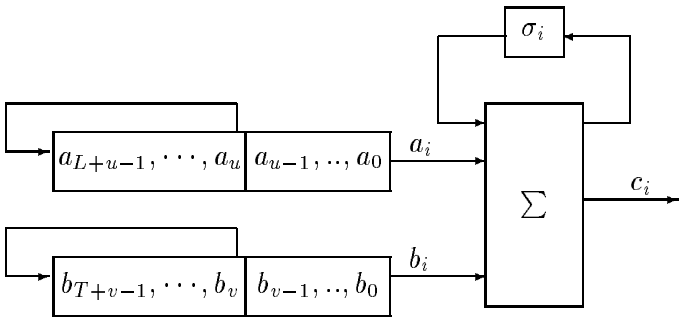

Figure 2.2: Summation of N-adic Sequences

summation sequence of A and B, let $\sigma_{-1} = 0$. Then at every index $i \geq 0$, we have $a_i + b_i + \sigma_{i-1} = c_i + \sigma_i N$, and $C = \{c_i\}$ is the expected output sequence. It follows that C is the coefficient sequence of the N-adic number $\alpha(A, N) + \alpha(B, N)$. Figure 2.2 displays the device construction.

In Figure 2.2, the feedback arrows represent the pure cycling which produce the strictly periodic parts of A and B, respectively. Since $a_i, b_i < N$, the carry bit σ_i is either 0 or 1. This implies that the device depicted in Figure 2.2 only has finitely many states. Therefore, the output sequence C is eventually periodic. □

The following result states the relation between rational numbers and eventually periodic sequences over S.

Proposition 2.2.4 *Let A be an N-adic sequence. Then*

(1) $\alpha(A, N) = -1$ if and only if $A = (N-1, N-1, N-1, \cdots)$;

(2) $\alpha(A, N) = -u/q, gcd(u, q) = 1$ with $q > 1$ and $gcd(N, q) = 1$ if and only if A is eventually periodic;

(3) $0 \leq u \leq q$ if and only A is strictly periodic.

Proof: We only need to show (2) and (3). First we assume that the sequence A is strictly periodic, i.e., $a_{i+L} = a_i,\ i \geq 0$ for a fixed L. Multiplying α by $(1 - N^L)$, we

get that

$$\begin{aligned}(1-N^L)\alpha &= \sum_{i=0}^{\infty} a_i N^i - \sum_{i=0}^{\infty} a_i N^{i+L} \\ &= \sum_{i=0}^{L-1} a_i N^i + (\sum_{i=L}^{\infty} a_i N^i - \sum_{i=0}^{\infty} a_i N^{i+L}) \\ &= \sum_{i=0}^{L-1} a_i N^i \ .\end{aligned}$$

Thus $\alpha = \sum_{i=0}^{L-1} a_i N^i/(1-N^L) = -u/q$ with $q > u \geq 0$ and $gcd(N,q) = 1$ and $gcd(u,q) = 1$.

Conversely, let $L = ord_q(N)$, i.e., the smallest positive integer L such that $N^L \equiv 1 \ (mod\ q)$. Let k be an integer such that $N^L - 1 = kq$. Then $-u/q = ku/(1-N^L)$. Since $ku < kq = N^L - 1$, the N-adic expansion of ku has the highest term N^j with $j \leq L-1$. That is,

$$ku = a_0 + a_1 N + a_2 N^2 + \cdots + a_{L-1} N^{L-1}.$$

Moreover,

$$\frac{1}{1-N^L} = 1 + N^L + N^{2L} + \cdots.$$

It follows that

$$\frac{ku}{1-N^L} = \sum_{i=0}^{\infty} a_i N^i$$

with $a_i = a_{i \ mod\ L}$. Therefore, $-u/q$ has a strictly periodic N-adic coefficient sequence.

For an eventually periodic sequence A, there is an index i_0 such that

$$A' = \{a_{i_0}, a_{i_0+1}, a_{i_0+2}, \cdots, a_{i_0+j}, \cdots, \}$$

is strictly periodic. Therefore, by the the first part proof there is a u_0 such that

$$\alpha(A,N) = \sum_{i=0}^{i_0-1} a_i N^i + N^{i_0} \frac{-u_0}{q} \ .$$

It follows that $\alpha(A,N) = -u/q$ for some u.

Conversely, if $\alpha = -u/q$ with $q > 1$ and $gcd(N,q) = 1$, we can rewrite the rational number as $\alpha = v - (u_0/q)$ such that $0 \leq u_0 < q$. By the first part proof, $-u_0/q$ is the

generating number of a strictly periodic sequence $B = \{b_i\}$, i.e., $-u_0/q = \sum_{i=0}^{\infty} b_i N^i$. If v is positive, it has a finite N-adic expansion, i.e., $v = v_0 + v_1 N + \cdots + v_t N^t$ for some $t > 0$. Let $v_i = 0, i \geq t$. Then by Lemma 2.2.3, the summation sequence of $\{v_i\}$ and B is eventually periodic.

If $v < 0$, we have $v = -|v|$, and $|v| = v_0 + v_1 N + \cdots + v_t N^t$ for some $t > 0$. Note that for each v_i, $-v_i = (N - v_i) + (N-1)N + (N-1)N^2 + \cdots +$. Hence the N-adic number $v = -v_0 - v_1 N - \cdots - v_t N^t$ is a sum of finitely many N-adic numbers whose coefficient sequences are eventually periodic. Thus, $\alpha = v - (u_0/q)$ itself is a sum of finitely many N-adic numbers whose coefficient sequences are eventually periodic. It then follows from Lemma 2.2.3 that the resulting sequence is eventually periodic. □

Here we are only concerned with the problem that for any rational number $-u/q$ with $gcd(N, q) = 1$, the coefficient sequence of its N-adic expansion is eventually periodic. In the next chapter, we give an algorithm which explicitly constructs an N-FCSR to generate the sequence corresponding to the number $-u/q$, and the size of the constructed register will be minimized.

By Propositions 2.2.1 and 2.2.4, we see that the generating number for the output sequence of an N-FCSR is a rational number with the denominator relatively prime to N. Furthermore, this rational number can be explicitly expressed by using the register configuration.

Theorem 2.2.5 *For an N-FCSR, let q be the connection number with $q_0 = -d$:*

$$q = q_0 + q_1 N + q_2 N^2 + \cdots + q_r N^r,$$

and let the initial memory be M_{r-1} and the initial state $(a_{r-1}, \cdots, a_0)$. Then the generating number with respect to N can be expressed as follows:

$$\alpha = \frac{\sum_{n=0}^{r-1} (\sum_{i=0}^{n} q_i a_{n-i}) N^n - M_{r-1} N^r}{q}.$$

Proof: At a given state of index $n-1$, suppose the memory is M_{n-1} $(n \geq r)$ and the register cells are loaded as: $(a_{n-1}, \cdots, a_{n-r})$. We then look at the state change. By definition, we have that

$$\begin{aligned}\sigma_n &= M_{n-1} + \sum_{i=1}^{r} a_{n-i} q_i, \\ &= da_n + M_n * N.\end{aligned}$$

Therefore, we have that

$$\begin{aligned}da_n &= M_{n-1} + \sum_{i=1}^{r} a_{n-i} q_i - M_n * N \\ &= \sum_{i=1}^{r} a_{n-i} q_i + (M_{n-1} - M_n * N).\end{aligned}$$

Thus

$$\begin{aligned}\alpha d &= d(\sum_{i=0}^{\infty} a_i N^i) \\ &= d(\sum_{i=0}^{r-1} a_i N^i + \sum_{n=r}^{\infty} a_n N^n) \\ &= d(\sum_{i=0}^{r-1} a_i N^i) + \sum_{n=r}^{\infty}(\sum_{i=1}^{r} q_i a_{n-i}) N^n + \sum_{n=r}^{\infty}(M_{n-1} - M_n * N) N^n \\ &= d(\sum_{i=0}^{r-1} a_i N^i) + M_{r-1} N^r + \sum_{i=1}^{r} q_i N^i (\sum_{n=r}^{\infty} a_{n-i} N^{n-i}) \\ &= (\sum_{i=1}^{r} q_i N^i)\alpha + d(\sum_{i=0}^{r-1} a_i N^i) - \sum_{i=1}^{r-1} \sum_{j=0}^{r-i-1} q_i N^i a_j N^j + M_{r-1} N^r.\end{aligned}$$

Note that $q_0 = -d$, and we then can move the first term to the left and rearrange the indices. It follows that

$$\alpha = \frac{\sum_{n=0}^{r-1}(\sum_{i=0}^{n} q_i a_{n-i}) N^n - M_{r-1} N^r}{q}. \qquad \square$$

2.3 Exponential Representation

The trace representation of LFSR sequences is a powerful technique in the analysis of sequences (see Chapter 1). Although the algebra used in analyzing N-FCSRs is different, there is a similar representation for periodic N-adic sequence generated by an N-FSCR.

Theorem 2.3.1 *Suppose a periodic N-adic sequence* $A = \{a_i : i \geq 0\}$ *is generated by an N-FCSR with the connection number* $q > 1$. *Let* $\gamma = N^{-1} \in Z/(q)$ *be the multiplicative inverse of* N *in the ring* $Z/(q)$. *Then there exists an element* $C \in Z/(q)$ *and an invertible element* $D \in Z/(N)$ *such that for all* $i \geq 0$

$$a_i = D[C\gamma^i \ (mod\ q)] \ (mod\ N) \ .$$

Here the notation means that first the integer $C\gamma^i$ *is reduced to the remainder after division by* q *(mod* q *operation); then the remainder is multiplied by a constant* D $(0 < D < N-1)$, *and then the product is reduced further to a number between* 0 *and* $N-1$ *by the modular operation mod* N. *Furthermore,* $D = (-q)^{-1}(mod\ N)$ *is only dependent on* q *and* N, *and* $D = 1$ *if* $q \equiv -1\ (mod\ N)$. *This is always true when* $N = 2$. *Also* $\alpha(A) = -C/q$.

Note that if the sequence A is a zero sequence, the result is still valid by choosing C to be zero. Also note that if $\alpha = -1$, then the sequence must be $A = \{a_i = N-1 : i \geq 0\}$. In this case the result is valid by choosing γ to be 1, C to be 1 and D to be $N-1$.

Proof: In the initial state S_0, suppose the memory has the value M and the register cells are loaded with $a_0, a_1, \cdots, a_{r-1}$. By the assumption, the sequence is strictly periodic. Hence, the rational representation is $\alpha = -u_0/q$ with $0 < u_0 < q$ and the period is $T = ord_q(N)$. Starting at the initial state, every state S_t afterward has an associated rational number with the same denominator q and a numerator u_t, denoted by

$$f(S_t) = -\frac{u_t}{q} = \sum_{i=0}^{\infty} a_{i+t} N^i$$

with $0 \leq u_t \leq q-1$. Consider the states S_{t-1} and S_t. We have that

$$f(S_{t-1}) = a_{t-1} + Nf(S_t), \ i.e. \quad -N\frac{u_t}{q} + a_{t-1} = \ -\frac{u_{t-1}}{q}$$

This implies that $u_{t-1} = Nu_t - a_{t-1}q \in Z$. This follows that $u_{t-1} \equiv Nu_t (mod\ q)$, or equivalently, $u_t = N^{-1}u_{t-1}(mod\ q)$. Hence each term of the sequence $\{u_0, u_1, \cdots, \}$ is obtained by multiplying the previous term by γ and reducing modulo q.

Hence, $u_t = \gamma^t u_0 \ (mod\ q)$. Note that $-a_{t-1}q \equiv u_{t-1} \ (mod\ N)$. Let $D = (-q)^{-1}\ mod\ N$, we then have that

$$a_{t-1} \equiv Du_{t-1} \ (mod\ N) = D[u_0\gamma^{t-1}\ mod\ q] \ (mod\ N) \ . \quad \square$$

Example 3: Consider an N-FCSR with the configuration: $N = 10, r = 3, q = 1109$, and the initial loading $a_0 = 2, a_1 = 7, a_2 = 9$ and the initial memory $m = 0$. Then the output sequence is $\{2, 7, 9, 8, 5, 4, 9, 9, 3, 3, 7, 4, 5, 7, 7, 0, 6, 4, 1, 2, \cdots, \}$. By Theorem 2.2.5, the rational number is $\alpha = -52/1109$. Note that $\gamma = 10^{-1}(mod\ 1109) = 111$, $D = (-q)^{-1} mod\ 10 = 1$ and $u = 52$. It follows that

$$a_i = [52 * (111)^i\ mod(1109)](mod\ 10)$$

for all $i \geq 0$.

Remark: In Theorem 2.3.1, the constant D is necessary, and it cannot be moved to the inside of the modular operation. For example, let $N = 10, r = 3, q = 1103$. Then $D = 3$ and $\gamma = 331$. Assume a sequence $A = \{a_i\}$ is generated as

$$a_i = D[552\gamma^i(mod\ q)](mod\ N) = 3[552 * 331^i(mod\ 1103)](mod\ 10).$$

For $i = 0$, $a_0 = 3[552(mod\ 1103)](mod\ 10) = 1656(mod\ 10) = 6$, but if D is moved to the inside, it gives that

$$\begin{aligned} [D552\gamma^0(mod\ 1103)](mod\ N) &= [3 * 552(mod\ 1103)](mod\ 10) \\ &= [1656(mod\ 1103)](mod\ 10) \\ &= [553](mod\ 10) \\ &= 3\ . \end{aligned}$$

Hence, $a_0 \neq [D552\gamma^0(mod\ 1103)](mod\ N)$.

2.4 l-sequences

By Theorem 2.2.5, an N-FCSR with a given initial state and memory has an associated rational number $\alpha = u/q$. We call an N-FCSR *reduced* if $gcd(q, u) = 1$. Then the period of the output sequence can be computed by just using the connection number q. The following is a corollary to Theorem 2.3.1.

Corollary 2.4.1 *Let $\alpha = u/q$. Then the output sequence of an N-FCSR with the connection number q has period $T = ord_q(N)$ provided that $gcd(u, q) = 1$. In particular, if q is a prime power and N is a primitive root of $Z/(q)^*$, then $T = \varphi(q)$.*

One consequence of this corollary is that for $q = p^m$, an odd prime power of p, there is an integer N and an N-FCSR with connection number q whose output sequence has period equal to the size $\varphi(q)$ of the multiplicative group $Z/(q)^*$, where φ is the Euler phi-function. This is because in this case there exists at least one primitive root N in $Z/(q)^*$ (p.24, [9]). In [23], the notion of an l-sequence is defined for $N = 2$. That is, a sequence generated by a 2-FCSR whose connection number q is a prime power such that 2 is a primitive root modulo q. In general, we have the following definition.

Definition 2.4.1 *An N-adic periodic sequence A is called an l-sequence if it can be generated by an N-FCSR with a prime power connection number q such that the element N is primitive in $Z/(q)$.*

For an odd prime power q of p, to find a primitive root modulo q there is no known better way to proceed than as follows. Try $g = 2,\ g = 3$, etc ... until g is a primitive root. See Cohen's book [9] for more detailed discussion of this problem. In the reference, it is pointed out that for some special cases there are better algorithms. For example, if the prime factorization of $\varphi(q)$ is known, then the Chinese remainder theorem can be used to reduce the problem to that for a small prime power. Table 2.3 displays all the primitive roots of $Z/(q)^*$ for prime numbers $q < 100$. The table was made by an exhaustive search.

As we see in Table 2.3, for example, for some prime numbers, 2 is primitive root, but 10 is not; or, vice versa. Schneier gave a complete list of prime numbers $< 10,000$ for which 2 is primitive [36]. There are efficient techniques for finding large primes q for which 2 is a primitive root [9]. The following result holds.

Proposition 2.4.2 *For a prime power q, if 2 is a primitive root in $Z/(q)$, then N is a primitive root if and only if $N = 2^t \pmod q$ for some t with t relatively prime to $\varphi(q)$.*

The next two results explore some distribution properties of N-adic l-sequences. First we consider l-sequences with prime connection numbers. In this case, for any two different symbols s_i and s_j, the number of the occurrences of s_i differs that of s_j at most by one.

Table 2.3: Primitive Roots of $Z/(q)$

prime number	primitive roots
3	2
5	2, 3
7	3, 5
11	2, 6, 7, 8
13	2, 6, 7, 11
17	3, 5, 6, 7, 10, 11, 12, 14
19	2, 3, 10, 13, 14, 15
23	5, 7, 10, 11, 14, 15, 17, 19, 20, 21
29	2 , 3, 8, 10, 11, 14, 15, 18, 19, 21, 26, 27
31	3, 11, 12, 13, 17, 21, 22, 24
37	2, 5, 13, 15, 17, 18, 19, 20, 22, 24, 32, 35
41	6, 7, 11, 12, 13, 15, 17, 19, 22, 24, 26, 28, 29, 30, 34
	35
43	3, 5, 12, 18, 19, 20, 26, 28, 29, 30, 33, 34
47	5, 10, 11 13 15 19 20 22 23 26 29 30 31 33 35
	38, 39, 40, 41, 43, 44, 45
53	2, 3, 5, 8, 12, 14, 18, 19, 20, 21, 22, 26, 27, 31, 32
	33, 34, 35, 39, 41, 45, 48, 50, 51,
59	2, 6, 8, 10, 11, 13, 14, 18, 23, 24, 30, 31, 32, 33, 34
	37, 38, 39, 40, 42, 43, 44, 47, 50, 52, 54, 55, 56
61	2, 6, 7, 10, 17, 18, 26, 30, 31, 35, 43, 44, 51, 54, 55
	50
67	2, 7, 11, 12, 13, 18, 20, 28, 31, 32, 34, 41, 44, 46, 48
	50, 51, 57, 61, 63
71	7, 11, 13, 21, 22, 28, 31, 33, 35, 42, 44, 47, 52, 53, 55
	56, 59, 61, 62, 63, 65, 67, 68, 69
73	5, 11, 13, 14, 15, 20, 26, 28, 29, 31, 33, 34, 39, 40, 42
	44, 45, 47, 53, 58, 59, 60, 62, 68
79	3, 6, 7, 28, 29, 30, 34, 35, 37, 39, 43, 47, 48, 53, 54
	59, 60, 63, 66, 68, 70, 74, 75, 77
83	2, 5, 6, 8, 13, 14, 15, 18, 19, 20, 22, 24, 32, 34, 35
	39, 42, 43, 45, 46, 47, 50, 52, 53, 54, 55, 56, 57, 58, 60
	62, 66, 67, 71, 72, 73, 74, 76, 79, 80
89	3, 6, 7, 13, 14, 15, 19, 23, 24, 26, 27, 28, 29, 30, 31
	33, 35, 38, 41, 43, 46, 48, 51, 54, 56, 58, 59, 60, 61, 62
	63, 65, 66, 70, 74, 75, 76, 82, 83, 86
97	5, 7, 10, 13, 14, 15, 17, 21, 23, 26, 29, 37, 38, 39, 40
	41, 56, 57, 58, 59, 60, 68, 71, 74, 76, 80, 82, 83, 84, 87
	90, 92

Proposition 2.4.3 *Let q be a prime and N a primitive root in $Z/(q)$. Let A be a periodic N-adic l-sequence, i.e., a periodic sequence generated by an N-FCSR with the connection number q. In a single period, define $A_j = |\{i : a_i = j, 0 \le i < q-1\}|$ for each $j : 0 \le j \le N-1$. Then, for any $j \ne k$ we have that $|A_j - A_k| \le 1$.*

Proof: Let $\alpha(A) = -u/q$ be the rational representation of the sequence A. By Theorem 2.3.1, $a_i = D[uN^{-i}(mod\ q)]\ (mod\ N), i \ge 0$. Here $D = (-q)^{-1}\ mod\ N$. Since q is a prime, it follows that both N^{-1} and u are invertible modulo q. Note that N^{-1} is primitive in $Z/(q)$. This then implies that $\{uN^{-i}(mod\ q)|0 \le i < q\} = (Z/(q))^*$, the multiplicative group. Thus, $\{b_i = uN^{-i}(mod\ q)\} = \{1, 2, \cdots, q-1\}$. Let $q = kN + v,\ 1 \le v \le N-1$. Since $a_i = Db_i(mod\ N)$ and D is invertible, the number A_j is the same as the number of occurrences of j in the set $\{b_i(mod\ N) : 0 \le i \le q-1\}$. We can list b_i in a matrix as follows:

$$\begin{array}{cccccc}
 & 1 & 2 & \cdots & v-1 & \cdots & N-1 \\
N & N+1 & N+2 & \cdots & N+v-1 & \cdots & 2N-1 \\
2N & 2N+1 & 2N+2 & \cdots & 2N+v-1 & \cdots & 3N-1 \\
\vdots & \vdots & \vdots & & \vdots & & \vdots \\
(k-1)N & (k-1)N+1 & (k-1)N+2 & \cdots & (k-1)N+v-1 & \cdots & kN-1 \\
kN & kN+1 & kN+2 & \cdots & kN+v-1 & &
\end{array}$$

The elements in j-th column contribute to A_j. Hence, $A_j = u+1$ for $1 \le j \le v-1$; $A_j = u$ for $j = 0$ or $v \le j \le N-1$. □

When connection numbers are prime powers, the numbers of occurrences of different symbols are still close to being balanced. To be precise, we recall a distribution property of finite sequences. An N-adic sequence A of period T is said to have the de Bruijn property if for any two distinct subsequences B and C of length less than $\log_2(T)$, in a single period of A the numbers of occurrences of B and C are the same. The de Bruijn property is one of Golomb's randomness postulates [14, 35]. Next we show that N-adic l-sequences are close to having the de Bruijn property.

Theorem 2.4.4 *Let q be a power of a prime p, say $q = p^e$, for some $e > 0$. Assume that N is primitive modulo q. Let A be a (purely) periodic N-adic l-sequence generated by an N-FCSR with connection number q. For any integer $s > 0$, let S_1 and S_2 be two distinct sequences with length s. Then the numbers of occurrences of S_1 and S_2*

with their starting positions in a fixed period of A differ by at most 2. Furthermore, if $s > \lfloor \log_N(q+N) \rfloor$, *a subsequence of A of length s occurs at most once.*

Proof: The purely periodic N-adic sequence with connection number q is precisely the N-adic expansion of a rational number $-x/q$ with $0 < x < q$. The sequence has maximal period if and only if $gcd(x, q) = 1$. Since N is primitive modulo q, the cyclic shifts of A correspond to the set of all rational numbers $-x/q$ with $0 < x < q$ and $gcd(x, q) = 1$. Thus S_1 occurs in A if and only if it occurs as the first s-coefficients in the N-adic expansion of some rational number $-x/q$ with $0 < x < q$ and $gcd(x, q) = 1$. Two such rational numbers $-x_1/q$ and $-x_2/q$ have the same first s-coefficients in their N-adic expansions if and only if $-x_1/q \equiv -x_2/q \ (mod N^s)$. That is,

$$x_1 - x_2 \equiv 0 \ mod \ N^s.$$

Thus we only need to count the number of x such that $0 < x < q$, $p \not| x$ and x has the first s-symbols fixed. (Equivalently, the expansion of $-x/q$ has the prefix S_1.)

Let $r = \lfloor \log_N(q+N) \rfloor$. Then $(q+N)$ has an N-adic expansion:

$$q + N = b_0 + q_1 N + \cdots + q_r N^r \ , 0 \le q_i \le N-1, 0 < b_0 \le N-1, q_r \neq 0.$$

Let $d = N - b_0$ and $q_0 = -d$. Then $q = \sum_{i=0}^{r} q_i N^i$. Since $gcd(q, N) = gcd(b_0, N) = 1$, it follows that $gcd(d, N) = 1$. Therefore, by the definition of N-FCSRs, the corresponding register has length r.

Suppose $s > r$. Since the N-FCSR associated with $-x/q$ has length r, $0 < x < q < N^s$ and then x is uniquely determined by its congruence class mod N^s. Thus, there is either zero or one such x for a given r symbols. Hence a subsequence of A having length s occurs at most once.

We now assume $s \le r$. First we count the number of x such that $0 < x < q$ and the first s-symbols are fixed, ignoring the condition $gcd(x, q) = 1$. Let $C = c_0, c_1, \cdots, c_{s-1}$, and

$$w = \sum_{i=0}^{s-1} c_i N^i, \quad q = \sum_{i=0}^{r} b_i N^i, \quad q' = \sum_{i=0}^{s-1} b_i N^i \ .$$

If $w < q'$, then every choice of $c_s, \cdots, c_r$ with $\sum_{i=s}^{r} c_i N^i \le \sum_{i=s}^{r} b_i N^i$ gives a unique x such that $0 < x < q$. If $w \ge q'$, then every choice of $c_s, \cdots, c_r$ with $\sum_{i=s}^{r} c_i N^i <$

$\sum_{i=s}^{r} b_i N^i$ gives a unique x in the right range. Note that for every x with $0 < x < q$, by comparing the coefficients $c_r, b_r; c_{r-1}, b_{r-1}; \cdots, c_s, b_s$ in order, the number x must be in one of the two cases discussed. Also note that $\sum_{i=s}^{r} c_i N^i = \sum_{i=s}^{r} b_i N^i$ if and only if all the coefficients are the same, i.e., there is only one choice of $c_s, \cdots, c_r$ that makes the equality hold. Therefore, for different choices of first s-symbols, the numbers of corresponding x's differ by at most one.

Next we consider the condition $gcd(x, q) = 1$. If $gcd(x, q) > 1$, then $x = py$ for some y, $0 < y < p^{e-1}$. For $x_1 = py_1, x_2 = py_2$, since $gcd(N, q) = 1$, we have that $x_1 \equiv x_2 \ (mod\ N^s)$ if and only if $y_1 \equiv y_2 \ (mod\ N^s)$. By the proof of the preceding paragraph, the numbers of such y's differ by at most one. Let G_w^q be the number of x's with $0 < x < q$, $gcd(x, q) = 1$ and $x \equiv w(mod\ N^s)$; $G_{w,1}^q$ be the number of x's with $0 < x < q$ and $x \equiv w(mod\ N^s)$; and $G_{w,2}^q$ be the number of x's with $0 < x < q$, $p|x$ and $x \equiv w(mod\ N^s)$. Thus $G_{w,1}^q = G_w^q + G_{w_2}^q$. For any $x = py$, if $x \equiv w(mod\ N^s)$, then multiplying by the inverse of p in $Z/(N^s)$, we have $y \equiv w^*(mod\ N^s)$, where $w^* = p^{-1}w(mod\ N^s)$. Conversely, if $y \equiv w^*(mod\ N^s)$, then $x = py \equiv w(mod\ N^s)$. That is, $G_{w,2}^q = G_{w^*,1}^{q/p}$. Therefore, for two distinct w, w', the following equations hold

$$|G_{w,2}^q - G_{w',2}^q| = |G_{w^*,1}^{q/p} - G_{w'^*,1}^{q/p}| \leq 1.$$

Therefore, the G_w^q can be derived as follows:

$$\begin{aligned} G_w^q &= |\{x : 0 < x < q,\ gcd(x, q) = 1,\ x \equiv w\ mod\ N^s\}| \\ &= |\{x : 0 < x < q, x \equiv w\ modN^s\}| - |\{x : 0 < x < q, p|x, x \equiv w\ modN^s\}| \\ &= G_{w,1}^q - G_{w,2}^q\ . \end{aligned}$$

Thus, for two distinct w and w', we have that

$$\begin{aligned} |G_w^q - G_{w'}^q| &\leq |G_{w,1}^q - G_{w',1}^q| + |G_{w,2}^q - G_{w',2}^q| \\ &\leq |G_{w,1}^q - G_{w',1}^q| + |G_{w^*,1}^{q/p} - G_{w'^*,1}^{q/p}| \\ &= 2\ . \end{aligned}$$

This completes the proof. □

2.5 Linear Complexities of N-adic l-Sequences

As discussed in Section 1.3, the linear complexity of an N-adic sequence A is defined as the smallest length of an N-LFSR that generates A. Reeds and Sloane extended the Berlekamp-Massey algorithm to N-LFSRs [34]. Hence, for an N-adic sequence linear complexity is an important security measure. However, high linear complexity alone is not sufficient when other methods of cryptoanalysis exist. Note that one way to increase linear complexities is to employ non-linear operations in registers. Since N-FCSRs use integer addition with carry, this operation is highly non-linear compared with operations used in N-LFSRs. However, for a general N-adic sequence there may be no deterministic relation between its linear complexity and the smallest length of an N-FCSR that generates the sequence. In this section, we consider some special N-adic l-sequences and investigate their linear complexities. We show that the linear complexities of certain l-sequences are almost half of their periods. We must point out that Kim, Seo, Lee and Lim observed the same result as the author did in the special case when $N = 2$. They explored more statistical data to show that the existence of such sequences is not rare [40]. Here, we discuss the problem in a more general setting.

Recall that a binary bit $\bar{b}$ is called the complement of b if $b + \bar{b} = 1$. For $N > 1$, an N-adic symbol s is an integer such that $0 \leq s \leq N-1$. We can generalize the notion of binary bit-complement to N-adic symbols. Two N-adic symbols s and $\bar{s}$ are called *complementary* if $s + \bar{s} = N - 1$. An N-adic sequence of even length $2k$, $A = (a_0, a_1, \cdots, a_{2k-1})$ is called *symmetrically complementary* if $a_i + a_{i+k} = N-1, i = 0, 1, \cdots, k-1$. That is, if the second half of the sequence is added to the first half, then the resulting sequence is $(N-1, N-1, \cdots, N-1)$ of length k. Klapper and Goresky [15] first observed that 2-adic l sequences are symmetrically complementary. We use the notation $[x]_q$ for the reduced residue of x modulo q. Thus, $[x]_q = y$ if and only if $x \equiv y \ (mod\ q)$ with $0 \leq y < q$. For N-adic l-sequences, we present the following result.

Theorem 2.5.1 *Let $A = \{a_0, a_1, \cdots\}$ be an N-adic l-sequence generated by an N-FCSR with prime power connection number q. Then A is symmetrically complementary.*

Proof: Since A is an N-adic l-sequence, by Theorem 2.3.1, there is an integer $D = (-q)^{-1}(mod\ N)$ and a non zero integer $C \in Z/(q)$ such that $a_i = [D[CN^{-i}]_q]_N$ for $i \geq 0$. Let $\gamma = N^{-1}\ (mod\ q)$. Note that $\varphi(q) = 2t$ for some t and $\gamma^{2t} - 1 = 0\ (mod\ q)$. This implies that $(\gamma^t - 1)(\gamma^t + 1) \equiv 0\ (mod\ q)$. Since N is primitive modulo q, N^{-1} is primitive, too. It follows that $\gamma^t \equiv -1\ (mod\ q)$ and $[\gamma^t]_q = q - 1$. Therefore,

$$\begin{aligned} a_{i+t} &= [D[C\gamma^{i+t}]_q]_N \\ &= [D[-C\gamma^i]_q]_N \end{aligned}$$

for all $i \geq 0$. Let $[C\gamma^i]_q = t_i$. Since C and γ are relatively prime to q, $t_i > 0$. It then follows that $[-C\gamma^i]_q = q - t_i$ and $a_{i+t} + a_i = [Dt_i]_N + [D(q - t_i)]_N$. Since $D \equiv -q^{-1}(mod\ N)$, $Dq \equiv -1\ (mod\ N) \equiv N - 1(mod\ N)$. Let $[Dt_i]_N = z$. Then $Dt_i \equiv z\ (mod\ N)$. This implies that $D(q - t_i) \equiv (N - 1 - z)\ (mod\ N)$ and $0 \leq (N - 1 - z) < N$. Hence, $[D(q - t_i)]_N = N - 1 - z$ and $a_{i+t} + a_i = N - 1$. □

Corollary 2.5.2 *Let all the assumptions be the same as before and $\varphi(q) = 2t$ for some t. Then the linear complexity of the sequence A is at most $t + 1 = (\varphi(q) + 2)/2$.*

Proof: To compute the linear complexity of A over $R = Z/(N)$, we need to find a linear recurrence of A with minimal length. This is equivalent to finding a minimal degree polynomial $r(x) = \sum_{i=0}^{d} r_i x^i$ with $r_d \neq 0$ and r_0 invertible in R such that $\sum_{i=0}^{d} r_i a_{n-i} = 0$. Here, all coefficients are considered in the residue ring R, and all polynomials are considered in $R[x]$. Let $\alpha_A(x) = \sum_{i=0}^{\infty} a_i x^i$ be the generating function. By Theorem 2.5.1, we have that

$$\begin{aligned} \alpha &= \frac{\sum_{i=0}^{2t-1} a_i x^i}{1 - x^{2t}} \\ &= \frac{\sum_{i=0}^{t-1} a_i x^i + \sum_{i=0}^{t-1}(N - 1 - a_i)x^{i+t}}{1 - x^{2t}} \\ &= \frac{(\sum_{i=0}^{t-1} x^i)((1 - x)(a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}) + (N - 1)x^t)}{1 - x^{2t}} \\ &= \frac{(1 - x)(a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}) + (N - 1)x^t}{(1 - x)(1 + x^t)}. \end{aligned}$$

Since $(1 - x)(1 + x^t) = 1 - x + x^t - x^{t+1}$ gives a recurrence relation of A with the length $t + 1$, the linear complexity of A is not greater than $t + 1$.□

Note that since the second half of the sequence of A is the complement of the first half, the complexity of the sequence is essentially determined by its first half. We have not found a general condition on t and N such that the corresponding N-adic l-sequence has the linear complexity equal to $t+1$. Since both R and $R[x]$ have zero divisors when N is a composite integer, it is more complicated to define the fractions (quotients) over R and $R[x]$. For a detailed characterization, we refer to [7]. We first concern ourselves with the case when $N=2$. Since $-1=1$ over $Z/(2)$, $(1+x)(1+x^t)=(1+x)^2(\sum_{i=0}^{t-1}x^i)$, which we denote by $r(x)$. Let $d(x)=1+x+\cdots+x^{t-1}$ and $h(x)=(1-x)(a_0+a_1x+\cdots+a_{t-1}x^{t-1})+(N-1)x^t$. By the proof of Corollary 2.5.2, we see that $h(x)/r(x)$ is a rational representation of A over $R[x]$. We would like to know when

$$gcd(h(x),r(x))=gcd(\sum_{i=0}^{t-1}a_ix^i,\sum_{i=0}^{t-1}x^i)=1\ .$$

This is true when $d(x)$ is irreducible over $Z/(2)$ unless all the a_i are 1 or 0. This leads to the following result.

Theorem 2.5.3 *Let $A=\{a_0,a_1,\cdots,\cdots\}$ be a periodic 2-adic l-sequence generated by a 2-FCSR with the prime connection number $q=2p+1$. If p is an odd prime and 2 is primitive mod p, then the linear complexity of A is $\lambda(A)=p+1$.*

Proof: Note that $h(x)/r(x)$ is the rational representation of A over $R[x]$ and $h(1)=(N-1)1^p\neq 0$. It follows that $gcd(h(x),(1-x)^2)=1$. Since $r(x)=(1-x)^2d(x)$, we only need to show that $gcd(h(x),d(x))=1$. By assumption, 2 is primitive modulo p. I.e., the only cyclotomic cosets modulo p over $Z/(2)$ are $C_0=\{0\}$ and $C_1=\{1,2,\cdots,p-1\}$. Hence $d(x)$ is the minimal polynomial of θ, where θ is a primitive root of x^p-1. Thus $d(x)$ is irreducible [28, 7.5, p.197]. It remains to show that $d(x)\not| h(x)$. Suppose $d(x)|h(x)$. Then $h(\theta)=0$. Since $\theta^p=1$, we have

$$\begin{aligned}0 &= (1-\theta)(a_0+a_1\theta+\cdots+a_{p-1}\theta^{p-1})+\theta^p\\ &= (a_0-a_{p-1}+1)+(a_1-a_0)\theta+(a_2-a_1)\theta^2+\cdots+(a_{p-1}-a_{p-2})\theta^{p-1}\end{aligned}$$

Since $d(x)$ is the minimal polynomial of θ, the polynomial

$$f(x)=(a_0-a_{p-1}+1)+(a_1-a_0)x+\cdots+(a_{p-1}-a_{p-2})x^{p-1}$$

must be a multiple of $d(x)$. That is, $a_{p-i}-a_{p-i-1}=0, (1\leq i\leq p-1)$ and $a_0-a_{p-1}+1=0$; or $a_{p-i}-a_{p-i-1}=1, (1\leq i\leq p-1)$ and $a_0-a_{p-1}+1=1$. In the first case, we have that $a_0=a_1=\cdots=a_{p-1}$ and $a_0-a_{p-1}+1=0$, a contradiction.

For the second case, $a_0, a_1, \cdots, a_{p-1}$ can be solved as a solution of a system of linear equations over $Z/(2)$. In fact, the linear system has the following coefficient matrix over $Z/(2)$:

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & -1 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ & & & \cdots & & \\ & & & \cdots & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}$$

Since the matrix has rank $p-1$ and $Z/(2)$ only has two elements, the system has two solutions $a_0 = a_2 = \cdots = a_{p-1} = 1$ and $a_1 = a_3 = \cdots = a_{p-2} = 0$, or $a_0 = a_2 = \cdots = a_{p-1} = 0$ and $a_1 = a_3 = \cdots = a_{p-2} = 1$. That is, $A = (101010\cdots)$ or $A = (010101\cdots)$. Therefore $\alpha(A) = 1/(1-2^2)$ or $2/(1-2^2)$. This implies that $q = 3$ and $p = 1$. This is a contradiction since p is not prime. □

For convenience, a prime q is called a *2-prime* if 2 is primitive modulo q. A number $q = 2p+1$ is a *strong 2-prime* if q and p are prime and 2 is primitive modulo q and p. For such a q, the corresponding 2-FCSRs produce l-sequences whose periods are $2p$, and whose linear spans are $p+1$, more than half of their periods.

For a large integer m, what is the chance of finding a strong 2-prime q near m? We do not have a precise answer, but can give an approximation. By the Prime Number Theorem [9, 27], we need to try about $\ln(m)$ numbers near m to get a prime q. Since about one third of the primes have 2 as a primitive root [25], we need to try about $3\ln(m)$ numbers near m to get a prime q such that 2 is primitive modulo q. Write $q = 2p+1$. We want p to be prime and 2 to be primitive modulo p. Hence we have to try about $3\ln(m/2)$ such q's. Therefore, to find a strong 2-prime number q near m, we have to try about $9\ln(m)\ln(m/2)$ numbers near m.

Kim, Seo, Lee and Lim in [40] checked all the prime numbers having bit length up to 24. Their data (Table 2.4) indicates that the number of strong 2-primes increases proportionally to the bit length.

Let $\nu(m)$ stand for the number of strong 2-primes having bit length $\ln(m)$. There are about $m/2$ integers having length $\ln(m)$. Hence, the number of strong 2-primes

Table 2.4: Number of Strong 2-Primes

bit length	10	11	12	13	14
2-primes	70	127	232	425	814
strong 2-primes	1	3	7	10	17
bit length	15	16	17	18	19
2-primes	1521	2861	5393	10179	19424
strong 2-primes	32	62	97	172	295
bit length	20	21	22	23	24
2-primes	36912	70499	134766	257971	495113
2-strong prime	542	924	1748	3162	5838

of length $\ln(m)$ is about $m/(18\ln(m)\ln(m/2))$. From Table 2.4, we may observe the following:

$$\frac{m}{18 \cdot \ln(m)\ln(m/2)} \leq \nu(m) \leq \frac{m}{9 \cdot \ln(m)\ln(2m)} .$$

We also may notice that $\nu(m)$ is about one percent of the number of 2-primes with the same bit length.

When N is not 2, by using ring homomorphisms, we can prove a result similar to Theorem 2.5.3. Let N have the prime factorization:

$$N = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}.$$

Here, the p_i's are distinct prime numbers and $e_i > 0$ are integers. For each i, let $F_i = Z/(p_i)$ be the residue field. Then, for each i, there is a ring homomorphism $\varphi_i : R = Z/(N) \rightarrow F_i$. Furthermore, φ_i induces a homomorphism between the polynomial rings [17, 18]:

$$\bar{\varphi}_i : R[x] \rightarrow F_i[x].$$

We first state a simple lemma:

Lemma 2.5.4 *Let $A = \{a_0, a_1, \cdots, \cdots\}$ be a periodic N-adic l-sequence generated by an N-FCSR with the prime connection number $q = 2p + 1$. If $3|N$ and $p \geq 2$, then there exists at least one a_i such that $a_i (mod\ 3) = 0$.*

Proof: Since A is an l-sequence, by Theorem 2.3.1 there are constants $D \in Z/(N)$

and $C \in Z/(q)$ such that

$$a_i = D[CN^{-i}(mod\ q)](mod\ N),\ i \geq 0.$$

Since N^{-1} is primitive modulo q and $q \geq 5$, we have

$$\{CN^{-i}(mod\ q) : i = 0, 1, \cdots, q-1\} = \{1, 2, 3, 4, \cdots, q-1\}.$$

Hence there is an i such that $a_i = 3D(mod\ N)$. This implies that $a_i(mod\ 3) = 0$ because $3|N$. $\Box$

Theorem 2.5.5 *Let $A = \{a_0, a_1, \cdots, \cdots\}$ be a periodic N-adic l-sequence generated by an N-FCSR with the prime connection number $q = 2p+1$. Let $\lambda(A)$ be the linear complexity of A over $R = Z/(N)$, p_k a prime factor of N. If (1) p is an odd prime; (2) p_k is primitive modulo p; and (3) $q > N^2$, then $p \leq \lambda(A) \leq p+1$.*

Proof: By Corollary 2.5.2, we only need to show that $\lambda(A) \geq p$. Suppose $\lambda(A) < p$. This implies that A has a recurrence relation with length less than p, or, equivalently, the generating function of A over R can be represented as

$$\alpha(A, x) = \sum_{i=0}^{\infty} a_i x^i = \frac{f(x)}{g(x)}$$

with $deg(g(x)) < p$.

Applying the homomorphism $\varphi_k : R \to F_k$ to the sequence A, we have the induced sequence over F_k: $\bar{A} = \{\varphi_k(a_i) = \bar{a}_i\}$. Since $\bar{a}_i + \bar{a}_{p+i} \equiv (N-1)(mod\ p_k) \equiv -1(mod\ p_k)$, over F_k the generating function of the sequence $\bar{A}$ can be expressed as

$$\begin{aligned}
\alpha(\bar{A}) &= \frac{\sum_{i=0}^{2p-1} \bar{a}_i x^i}{1 - x^{2p}} \\
&= \frac{\sum_{i=0}^{p-1} \bar{a}_i x^i + \sum_{i=0}^{p-1} (N - 1 - \bar{a}_i) x^{i+p}}{1 - x^{2p}} \\
&= \frac{(\sum_{i=0}^{p-1} x^i)((1-x)(\bar{a}_0 + \bar{a}_1 x + \cdots + \bar{a}_{p-1} x^{p-1}) + x^p)}{1 - x^{2p}} \\
&= \frac{(1-x)(\bar{a}_0 + \bar{a}_1 x + \cdots + \bar{a}_{p-1} x^{p-1}) - x^p}{(1-x)(1+x^p)}.
\end{aligned}$$

Let

$$\bar{h}(x) = (1-x)(\bar{a}_0 + \bar{a}_1 x + \cdots + \bar{a}_{p-1} x^{p-1}) - x^p,$$

$$\begin{aligned}
\bar{r}(x) &= (1-x)(1+x^p) = (1-x)(1+x)(1-x+x^2-\cdots+x^{p-1}),\\
\bar{f}(x) &= \bar{\varphi}_k(f(x)),\\
\bar{g}(x) &= \bar{\varphi}_k(g(x)).
\end{aligned}$$

Hence, $\alpha(\bar{A}) = \bar{h}(x)/\bar{r}(x) = \bar{f}(x)/\bar{g}(x)$. Since the degree of $\bar{g}(x)$ is strictly less than p, $gcd(\bar{h}(x), \bar{r}(x))$ has degree strictly greater than 1. We show that in fact $deg(gcd(\bar{h}(x), \bar{r}(x))) \leq 1$, a contradiction. Let $\bar{d}(x) = (1 - x + x^2 - \cdots + x^{p-1})$ over F_k. Note that $\bar{h}(1) \neq 0$. Then $gcd(\bar{h}(x), \bar{r}(x)) = gcd(\bar{h}(x), (1+x)\bar{d}(x))$. We show that $\bar{d}(x)$ is irreducible in $F_k[x]$.

First, consider the polynomial $d^*(x) = 1 + x + x^2 + \cdots + x^{p-1}$ over F_k. Since p_k is primitive modulo p, the only p_k-cyclotomic cosets modulo p are $C_0 = \{0\}$ and $C_1 = \{1, 2, \cdots, p-1\}$. Hence, $d^*(x)$ is the minimal polynomial of θ, where θ is a primitive root $x^p - 1$ over an extension field of F_k, and thus $d^*(x)$ is irreducible [28, 7.5, p.197]. Note that $\bar{d}(x) = d^*(-x)$. It follows that $\bar{d}(x)$ is irreducible.

Next we show that $\bar{d}(x) \not| \bar{h}(x)$. Note that $\bar{d}(-\theta) = d^*(\theta) = 0$. Suppose $\bar{d}(x) | \bar{h}(x)$. Then $\bar{h}(-\theta) = 0$. From the equation $\theta^p = 1$, we have

$$\begin{aligned}
0 &= \bar{h}(-\theta)\\
&= (1+\theta)(\bar{a}_0 - \bar{a}_1\theta + \bar{a}_2\theta^2 - \cdots - \bar{a}_{p-2}\theta^{p-2} + \bar{a}_{p-1}\theta^{p-1}) - (-\theta)^p\\
&= (\bar{a}_0 + \bar{a}_{p-1} + 1) + (-\bar{a}_1 + \bar{a}_0)\theta + (\bar{a}_2 - \bar{a}_1)\theta^2 + (-\bar{a}_3 + \bar{a}_2)\theta^3 + \cdots\\
&\quad +(-\bar{a}_{p-2} + \bar{a}_{p-3})\theta^{p-2} + (\bar{a}_{p-1} - \bar{a}_{p-2})\theta^{p-1}
\end{aligned}$$

Since $d^*(x)$ is the minimal polynomial of θ, the polynomial

$$\begin{aligned}
w(x) &= (\bar{a}_0 + \bar{a}_{p-1} + 1) + (-\bar{a}_1 + \bar{a}_0)x + (\bar{a}_2 - \bar{a}_1)x^2 + (-\bar{a}_3 + \bar{a}_2)x^3 + \cdots\\
&\quad +(-\bar{a}_{p-2} + \bar{a}_{p-3})x^{p-2} + (\bar{a}_{p-1} - \bar{a}_{p-2})x^{p-1}
\end{aligned}$$

is a multiple of $d^*(x)$. That is, $w(x) = 0$ or $w(x) = \beta \cdot d^*(x)$ for some nonzero element $\beta \in F_k$. If $w(x) = 0$, it follows that $\bar{a}_0 = \bar{a}_1 = \cdots = \bar{a}_{p-1}$ and $\bar{a}_0 + \bar{a}_{p-1} + 1 = 0$. We shall show that this is impossible. Suppose $\bar{a}_0 = \bar{a}_1 = \cdots = \bar{a}_{p-1} = \bar{a}$. Then $\bar{a} \neq 0$ and

$$\begin{aligned}
\bar{h}(x) &= (1-x)(\bar{a}_0 + \bar{a}_1 x + \cdots + \bar{a}_{p-1}x^{p-1}) - x^p\\
&= \bar{a}(1-x)(1 + x + x^2 + \cdots + x^{p-1}) - x^p
\end{aligned}$$

$$\begin{aligned} &= \bar{a}(1-x^p) - x^p \\ &= \bar{a} - (\bar{a}+1)x^p. \end{aligned}$$

On the other hand, we have that $\bar{h}(x) = \bar{d}(x)(\bar{c}+x)$ for some element $\bar{c} \in F_k$. That is,

$$\begin{aligned} \bar{h}(x) &= (\bar{c}+x)(1-x+x^2-x^3+\cdots-x^{p-2}+x^{p-1}) \\ &= \bar{c}+(1-\bar{c})x+(\bar{c}-1)x^2+\cdots+(\bar{c}-1)x^{p-1}+x^p \\ &= \bar{a}-(\bar{a}+1)x^p. \end{aligned}$$

By comparing the coefficients over F_k, we have $\bar{c} = \bar{a} = 1$ and $\bar{h}(x) = 1+x^p = 1-2x^p$. This is impossible unless $p_k = 3$. If $p_k = 3$, then $\alpha(\bar{A}) = 1/(1-x)$. This implies that $\bar{A} = \{\bar{a}_i = 1 : i \geq 0\}$. Since $p_k = 3$ and p_k is a prime factor of N, by Lemma 2.5.4 there is some i such that $\bar{a}_i = 0$. This contradiction shows that k cannot be 3, and then $\bar{d}(x)$ cannot divide $\bar{h}(x)$.

We now consider the case $w(x) = \beta \cdot d^*(x)$ with $\beta \in F_k$ and $\beta \neq 0$. By comparing the coefficients of the polynomials, we have a system of linear equations in variables $(\bar{a}_0, \bar{a}_1, \cdots, \bar{a}_{p-2}, \bar{a}_{p-1})$ over F_k.

$$\begin{aligned} \bar{a}_0 + \bar{a}_{p-1} + 1 &= \beta \\ -\bar{a}_1 + \bar{a}_0 &= \beta \\ \bar{a}_2 - \bar{a}_1 &= \beta \\ -\bar{a}_3 + \bar{a}_2 &= \beta \\ &\vdots \\ -\bar{a}_{p-2} + \bar{a}_{p-3} &= \beta \\ \bar{a}_{p-1} - \bar{a}_{p-2} &= \beta \end{aligned}$$

The coefficient matrix is:

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ & & & \cdots & & \\ & & & \cdots & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}$$

If $p_k \neq 2$, the determinant is non-zero. Hence the coefficient matrix is non-singular and the system has the unique solution:

$$\bar{a}_{2i} = \frac{\beta - 1}{2}, \qquad \bar{a}_{2i+1} = \frac{-\beta - 1}{2}$$

If $p_k = 2$, the matrix has rank $p - 1$. The system has two solutions, $(010101\cdots)$ and $(101010\cdots)$. Therefore, in both cases $\bar{A}$ has the pattern $(xyxy\cdots)$, where x and y are two elements in F_k. Since the prime connection number $q > N^2$ and A is an l-sequence, the fractions $-1/q$, $-N/q$ and $-N^2/q$ represent three shifts of the sequence A. It follows that A contains the pairs $(10), (01)$ and (00). The presence of the first two pairs imply that $x \neq y$ and the last pair shows that $x = y = 0$. This is impossible.

Therefore, $gcd(\bar{h}(x), \bar{d}(x)) = 1$ and $(1 + x)$ is the only possible common factor of $\bar{h}(x)$ and $\bar{r}(x)$. Hence, we have shown that $deg(gcd(\bar{h}(x), \bar{r}(x))) \leq 1$, and this contradicts the previous result that $deg(gcd(\bar{h}(x), \bar{r}(x))) > 1$. This contradiction shows that $deg(g(x)) \geq p$, and then $\lambda(A) \geq p$. □

We now are interested in triples of integers (q, p, N) satisfying the conditions: (1) q is prime; (2) q = 2p + 1 and p is odd and prime; (3) N is primitive modulo q; (4) there is a prime factor of N, denoted as $v(N, p)$, such that $v(N, p)$ is primitive modulo p over the finite field $Z/(v(N, p))$. Table 2.6 lists all the such triples with $q < 10000$ and $N \leq 30$.

By Theorem 2.5.5 and Table 2.6, we see that there are many N-adic sequences whose linear complexities are significantly larger than the lengths of the corresponding N-FCSRs. For example, choose $q = 8699, p = 4349$ and $N = 10$. Then any decimal sequence generated by a 10-FCSR with connection number $q = 8699$ has linear complexity at least $p = 4349$, but the length of the 10-FCSR is only 4.

Theorem 2.5.5 raises some deep questions in number theory. For instance, we are

Table 2.5: Prime Chain of Length 6

p_1	p_2	p_3	p_4	p_5	p_6
89	179	359	719	1439	2879
63419	126839	253679	507359	1014719	2029439
127139	254279	508559	1017119	2034239	4068479
405269	810539	1621079	3242159	6484319	12968639
810809	1621619	3243239	6486479	12972959	25945919

concerned with pairs of primes (p, q) such that $q = 2p + 1$, or in general, a chain of primes $(p_1, p_2, \cdots, p_t)$ such that $p_{i+1} = 2p_i + 1, 1 \leq i \leq t - 1$. Let us call this a *prime chain* of length t. Now one interesting question is whether there exists a prime chain of length t for any $t > 1$. We have checked this question for numbers up to 1,000,000. The maximal length of prime chains we have found is 6. However it seems hard to show that length of a prime chain never exceeds 6. Note that if a prime chain starts at p and the last decimal digit of p is 1, 3, or 7, then the chain has length less than or equal to 3. Hence, if a prime chain exists of length greater than 3, then all the prime numbers in the chain have the last digit 9. Table 2.5 presents all the prime chains of length 6 with starting prime <1,000,000. Another question is whether there are infinitely many prime pairs (p, q) with $q = 2p + 1$. This question is similar to the problem of arithmetic progressions of prime numbers [38].

Table 2.6: List of Triples (q, p, N)

q=2p+1	p	N primitive modulo q	v(N,p)
7	3	3	3
11	5	2,6,8	2,3
23	11	7,10,11,14,20,21	7,2,11
47	23	5,10,11,15,19,20,22,23,30	5,11,19,23
59	29	2,6,8,10,11,14,18,24,30	2,3,11
83	41	13,14,19,22	13,7,19,11
107	53	2,5,6,8,15,18,20,21,22,24,26,28	2,5,3
167	83	5,10,13,15,20,26,30	5,2,13
179	89	6,7,18,21,23,24,26,28,30	3,7,23,13
227	113	5,6,15,17,18,20,24	5,3,17
263	131	10,14,20,28,29,30	2,29
347	173	2,5,6,7,8,15,17,18,19,20,21,22,24,26,28	2,5,3,7,17,19,11
359	179	7,14,21,26,28	7,2
467	233	5,6,11,15,18,20,24	5,3,11
479	239	13,19,26	13,19
503	251	19,29	19,29
563	281	6,15,18,22,24,26	3,11,13
587	293	2,5,6,8,11,13,14,15,18,19,20,23,24	2,5,3,11,13,7,19,23
719	359	19	19
839	419	11,17,22,26	11,17,2
863	431	7,13,14,21,26,28	7,13
887	443	5,10,15,20,23,29,30	5,2,23,29
983	491	10,20,22,26,29,30	2,29
1019	509	2,6,7,8,10,13,18,21,22,24,28,30	2,3,7,13
1187	593	5,6,7,15,18,20,21,22,24,26,28	5,3,7,11,13
1283	641	6,15,18,21,23,24	3,23
1307	653	2,5,6,8,14,15,18,20,24,29	2,5,3,29
1319	659	26,29	2,29
1367	683	5,7,10,11,13,14,15,17,20,21,22,26,28,30	5,7,11,13,17
1439	719	11,17,22,23	11,17,23
1487	743	5,10,13,15,20,26,29,30	5,13,29
1523	761	7,11,21,28	7,11
1619	809	6,18,21,22,24,30	3,11
1907	953	5,6,11,15,18,19,20,24	5,3,11,19
2027	1013	5,6,7,15,18,20,21,24,28,29	5,3,7,29
2039	1019	7,13,14,21,22,26,28	7,13,2
2099	1049	6,14,18,23,24,30	3,7,23
2207	1103	5,7,10,14,15,20,21,28,30	5,7
2447	1223	5,10,15,20,23,30	5,23
2459	1229	2,6,8,10,13,17,18,21,22,24,28,30	2,3,13,17,11

Table 2.6 Continued			
q=2p+1	p	N primitive modulo q	v(N,P)
2579	1289	11,13	11,13
2819	1409	6,13,18,19,24,30	3,13,19
2879	1439	7,14,17,21,23,28,29	7,17,23,29
2903	1451	10,20,30	2
2963	1481	6,15,18,21,24	3
2999	1499	17,19,23	17,19,23
3023	1511	11,22	11
3167	1583	5,10,15,20,30	5
3203	1601	6,7,15,17,18,21,22,23,24,28	3,7,17,11,23
3467	1733	2,5,6,8,15,18,20,21,22,24,26,28	2,5,3,11
3779	1889	6,14,18,19,22,24,30	3,7,19,11
3803	1901	2,6,8,15,17,18,20,21,22,24,28	2,3,17
3863	1931	10,13,20,26,30	2,13
3947	1973	2,6,8,14,15,18,20,24	2,3,7
4007	2003	5,10,15,20,30	5
4079	2039	11,22	11
4127	2063	5,10,15,19,20,30	5,19
4139	2069	2,6,8,10,13,18,21,24,28,29,30	2,3,13,29
4259	2129	6,18,22,24,30	3,11
4283	2141	2,6,8,11,14,15,18,20,24	2,3,11,7
4547	2273	5,6,7,11,15,18,19,20,21,23,24,26,28	5,3,7,11,19,23,13
4679	2339	22,29	2,29
4787	2393	5,6,14,15,18,20,22,24,29	5,3,7,11,29
4799	2399	11,13,19,22,26,29	11,13,19,29
4919	2459	13,17,26	13,17,2
5087	2543	5,10,15,20,23,30	5,23
5099	2549	2,6,8,10,14,18,19,22,23,24,26,30	2,3,19,23
5387	2693	2,5,6,7,8,13,15,18,20,21,22,24,28	2,5,3,7,13
5399	2699	7,11,14,21,22,28	7,11,2
5483	2741	2,6,8,15,18,20,21,24,26,28,29	2,3,13,29
5507	2753	5,6,15,18,20,24,29	5,3,29
5639	2819	14,17,23,28	2,17,23
5807	2903	5,10,15,20,30	5
5879	2939	22,29	2,29
5927	2963	5,10,11,15,17,20,22,30	5,2,11,17
5939	2969	6,13,17,18,22,24,30	3,13,17,11
6047	3023	5,10,13,15,17,20,26,30	5,13,17
6599	3299	26	2
6659	3329	6,11,18,21,23,24,30	3,11,23
6719	3359	11,13,22,23,26	11,13,23

Table 2.6 Continued			
q=2p+1	p	N primitive modulo q	v(N,P)
6779	3389	6,18,24,30	3
6827	3413	2,5,6,8,13,15,18,19,20,21,22,24,28	2,5,3,13,19
6899	3449	6,7,18,21,22,24,28,29,30	3,7,11,29
6983	3491	10,11,14,20,22,26,28,30	2,11
7079	3539	7,14,17,19,21,26,28	7,2,17,19
7187	3593	5,6,11,13,15,18,19,20,24	5,3,11,13,19
7247	3623	5,7,10,11,14,15,20,21,22,23,28,29,30	5,7,11,23,29
7523	3761	6,15,18,22,24	3,11
7559	3779	17,19,26	17,19,2
7607	3803	5,10,13,15,20,26,30	5,2,13
7643	3821	6,14,15,18,19,24	3,7,19
7703	3851	10,11,20,22,26,30	2,11
7727	3863	5,10,13,15,20,26,30	5,13
7823	3911	23	23
8039	4019	11,13,22,26	11,13,2
8147	4073	5,6,14,15,18,20,24	5,3,7
8423	4211	7,14,21,28	7
8543	4271	13,26,29	13,29
8699	4349	2,6,8,10,13,14,17,18,24,30	2,3,13,17
8747	4373	2,5,6,7,8,13,15,18,19,20,21,22,24,28	2,5,3,7,13,19,11
8783	4391	17	17
8819	4409	6,13,14,18,24,30	3,13,7
8963	4481	6,15,18,24,29	3,29
9467	4733	5,15,19,20	5,19
9587	4793	5,6,7,15,18,20,21,22,24,28,29	5,3,7,11,29
9839	4919	13,19,26,29	13,19,29
9887	4943	11,22	11

Chapter 3

The Synthesis of N-FCSRs

The main objective of this chapter is to develop a cryptographical analysis on N-adic key streams based on N-FCSRs. The core of the analysis is a rational approximation algorithm. For an N-adic sequence A, the synthesis problem is to find a minimal length N-FCSR that generates the entire sequence by only processing a part of A. The synthesis problem is solved by two steps: (1) compute the reduced rational representation of A; (2) use the rational representation to explicitly construct an N-FCSR that generates A. The approach used here is different from that used by Klapper and Goresky [23] for p-FCSRs. They employed lattice theory and de Weger's rational approximation algorithm for p-adic numbers [10]. This approach does not work for general N-FCSRs because the algebraic structures of general N-adic numbers are weaker. Also the rational approximation algorithm presented here is totally different from that developed by Bach [1]. Bach's algorithm is designed only for 'add-with-carry' generators that are special cases of N-FCSRs. Noticed that for N-adic numbers, the addition with carry prevents the convergence of direct modifications of the original Berlekamp-Massey algorithm [32]. We have found a way to overcome the difficulty caused by the addition-carry.

In Section 1, we introduce notation and provide preliminary results. In Section 2, we describe the rational approximation algorithm, and in Section 3, we prove the correctness and convergence of the algorithm. We then solve the register synthesis problem for N-FCSRs. The implementation details and the algorithm improvements are also discussed in this section. In Section 4, we present an algorithm that constructs

an N-FCSR for any reduced rational number u/q with $gcd(q, N) = 1$. Combining this with the rational approximation algorithm completely solves the N-FCSR register synthesis problem. In Section 6, we introduce an N-adic complexity (or N-adic span) for finite length sequences. Essentially, the N-adic span of a sequence is the length of the minimal length N-FCSRs that generates the given sequence. The N-adic span is different from the linear span that is based on LFSRs. As seen in Section 2.5, there are varieties of sequences that have small N-adic span, but large linear span. Cryptographically, this means that there are key streams that may be secure against LFSR-based attacks, but not secure against N-FCSR-based attacks. One interesting question of N-adic span is the distribution among all sequences of a fixed length. In this section, we prove that more than half of the sequences of a length n will have N-adic span greater then or equal to $n/2$.

3.1 Preliminaries

To analyze eventually periodic N-adic sequences, we have used integers, rational numbers and N-adic numbers as algebraic models. For further analysis, we need to introduce certain measures on these algebraic objects. Recall that $S = \{s : 0 \leq s \leq N-1\}$.

Definition 3.1.1 *For any integer $x \neq 0$, the index with respect to the base N, $ind_N(x)$, is defined as follows:*

(1) $ind_N(x) = t$ if $x > 0$, $x = a_0 + a_1N + \cdots + a_tN^t$ with $a_i \in S$ and $a_t \neq 0$;

(2) $ind_N(x) = ind_N(-x)$ if $x < 0$;

(3) $ind_N(0) = -\infty$.

Note that we have $ind_N(x) = ind_N(|x|)$, hence we may assume $x > 0$ when we compute the index. We may view $ind_N(x)$ as the floor of the logarithm of $|x|$ with respect to the base N. We have the following arithmetic properties of the index function.

Proposition 3.1.1 *For any two integers x and y, $ind_N(xy) \leq ind_N(x)+ind_N(y)+1$. In particular, if $a \in S, y > 0$, then $ind_N(ay) \leq ind_N(y) + 1$. Also, $ind_N(x + y) \leq max(ind_N(x), ind_N(y)) + 1$.*

Proof: If one of x, y is zero, it is true when we take the convention $a+(-\infty) = -\infty$ for any finite integer a. So for the multiplication we assume both x and y to be non-zero and positive. Let $s = ind_N(x),\ t = ind_N(y)$. We then have two expansions:

$$\begin{aligned} x &= a_0 + a_1 N + \cdots a_s N^s \ , \\ y &= b_0 + b_1 N + \cdots b_t N^t \ . \end{aligned}$$

It follows that $xy \leq (N^{s+1}-1)(N^{t+1}-1) = N^{s+t+2} - N^{s+1} - N^{t+1} + 1$, and then $ind_N(xy) \leq s+t+1 = ind_N(x) + ind_N(y) + 1$.

For the addition, we have that $|x+y| \leq |x| + |y| \leq (N^{s+1}-1) + (N^{t+1}-1)$ and then $ind_N(x+y) = ind_N(|x+y|) \leq (s+1)$ or $(t+1)$. $\Box$

Note that the inequalities can be equalities for both multiplication and addition. For instance, let $x = 7$, $y = 1$ and $z = 15$. Thus we have $ind_2(x) = 2, ind_2(y) = 0$ and $ind_2(z) = 3$, but $ind_2(x+y) = 3 = max(ind_N(x), ind_N(y)) + 1$ and $ind_2(xz) = 6 = ind_N(x) + ind_N(z) + 1$.

For any pair of integers (h, r) with $r \neq 0$, we define the following function induced by the index function. We assume the second component in any integer pair to be non-zero unless the contrary is stated explicitly.

Definition 3.1.2 *For $N > 1$, $\Phi_N(h, r) = max\{ind_N(h), ind_N(r)\}$.*

Proposition 3.1.2 *For any pairs (h_1, r_1), (h_2, r_2) and any $k \geq 0$, we have :*

(1) $\Phi_N(h_1 + h_2, r_1 + r_2) \leq max\{\Phi_N(h_1, r_1), \Phi_N(h_2, r_2)\} + 1$.

(2) $\Phi_N(h_1 r_2 + r_1 h_2, r_1 r_2) \leq \Phi_N(h_1, r_1) + \Phi_N(h_2, r_2) + 2$.

(3) $\Phi_N(N^k(h_1, r_1)) = k + \Phi_N(h_1, r_1)$.

Proof: By Definition 3.1.1 and Proposition 3.1.1, it is possible to show (1), (2) and (3). We present the proof of (2) as an example. It follows from the definition and Proposition 3.1.1 that

$$\begin{aligned} \Phi(h_1 r_2 + r_1 h_2, r_1 r_2) &= max\{ind_N(h_1 r_2 + r_1 h_2), ind_N(r_1 r_2)\} \\ &\leq max\{ind_N(h_1 r_2), ind_N(r_1 h_2), ind_N(r_1) + ind_N(r_2)\} + 1 \\ &\leq max\{\Phi(h_1, r_1) + \Phi(h_2, r_2), \Phi(h_1, r_1) + \Phi(h_2, r_2)\} + 2 \\ &= \Phi(h_1, r_1) + \Phi(h_2, r_2) + 2. \ \Box. \end{aligned}$$

Note that the inequalities in (1) and (2) also can be equalities. For instance, let N be 2 and $(h_1, r_1) = (6, 7), (h_2, r_2) = (9, 15)$. Thus, $\Phi(h_1, r_1) = 2$ and $\Phi(h_2, r_2) = 3$, but $\Phi(h_1 r_2 + r_1 h_2, r_1 r_2) = \Phi(153, 105) = 7 = \Phi(h_1, r_1) + \Phi(h_2, r_2) + 2$.

Recall that in the original Berlekamp-Massey algorithm, the degree of a polynomial and the maximal degree of a pair of polynomials act as norm functions. They are important in the proof of correctness and fast convergence of the algorithm. Here we replace them by the index function and the function Φ. Also note that in the Berlekamp-Massey algorithm the arithmetic operations are taken in the ring of polynomials over a field and the addition has no carry. Here we are doing integer arithmetic, so both multiplication and addition create carries and then may increase the values of *ind* and Φ.

By Theorem 2.2.5, any eventually periodic N-adic sequence $A = (a_0, a_1, \cdots)$ can be represented as a rational number $\alpha(A) = u/q$ with $gcd(N, q) = 1$. By removing the greatest common factor, we may further assume that $q > 0$ and $gcd(u, q) = 1$, i.e., the rational representation is reduced. Therefore, we have a unique integer pair (u, q) associated with the sequence A. The Φ-value of this pair is important. We state the following definition.

Definition 3.1.3 *Let u/q be the reduced rational representation of an N-adic sequence A. Then the N-adic span (or N-adic complexity) is defined as $\lambda_N(A) = \Phi(u, q)$.*

Comparing with the linear span, we note that the N-adic span of a sequence A may not be exactly equal to the minimal length of an N-FCSR that generates A. Also, the physical size of an N-FCSR is determined by both the length of the register and the memory size. However, the quantity $\lambda_N(A)$ is the dominating parameter in both the algorithm design and the register construction. It can be proved that the number of bits required for the register and $\lambda_N(A)$ only differs by a $\log(\lambda_N(A))$ term. We will return to this problem after presenting and proving the rational approximation algorithm.

3.2 The Rational Approximation Algorithm

We are concerned with finite N-adic sequences as inputs. In cryptanalysis, these sequences may be small parts of N-adic key streams. For a given input sequence, the output of the rational approximation algorithm will be a pair of integers (h, r) whose corresponding rational number h/r is the rational representation of the key stream.

Let A be an N-adic key stream and $\alpha = h/q$ be its associated rational number. Assume the input sequence is a (consecutive) part of the key stream. To compute α, we iteratively construct integer pairs (h_i, r_i). Each pair (h_i, r_i) is constructed such that the N-adic expansion of the rational number h_i/r_i matches the input sequence up to i-th position. When a new symbol is input, we find a new pair (h_{i+1}, r_{i+1}) if necessary by forming a linear combination of (h_i, r_i) and an earlier pair. This is done so that the new pair approximates more symbols.

To make this idea work, there are two things we must be concerned with: (1) pairs $\{(h_i, r_i)\}$ must converge to a pair (h', r') such that $h'/r' = h/r$; and (2) the number of iterations leading to convergence must be as small as possible. Note that from stage i to stage $i+1$, if the expansion of h_{i+1}/r_{i+1} matches one more symbol than that of h_i/r_i does, but the size $\Phi_N(h_{i+1}, r_{i+1})$ increases at least one, then the convergent speed will be very slow and the size of the pair that the algorithm converges to will be very large. Also note that in general a linear combination of integer pairs (h_1, r_1) and (h_2, r_2) may increase in size (Φ-value) because N-adic integer addition has carries. This is one of many difficulties of the algorithm.

We overcome this difficulty as follows: when we form a linear combination of (h_i, r_i) and an earlier pair to construct (h_{i+1}, r_{i+1}), we make the pair (h_{i+1}, r_{i+1}) approximate at least three more symbols, but the size of the new pair increase at most two. This idea works, but it adds complexity to the proof of the algorithm.

We first present the pseudocode of the rational approximation algorithm for any $N > 1$. Then in the next section we prove its correctness and fast convergence. The implementation details and improvements for small N are discussed later.

Let the input sequence be $\{a_i : 0 \le a_i \le N-1, 0 \le i \le L\}$ and $\alpha = \sum_{i=0}^{L} a_i N^i$.

Rational Approximation Algorithm for N-FCSR

Pre-work:

(1) shift the sequence: $\alpha \leftarrow 1 + N\alpha$.

(2) set $(h_0, r_0) = (0, 1), (h_1, r_1) = (a_0 + a_1 N + a_2 N^2, 1 + N^3), m = 0, i = 1$.

loop: {

if($i \geq L + 1$) quit the loop.

if($(h_i - r_i\alpha) \equiv 0 \ mod(N^{i+1})$) $h_{i+1} = h_i, r_{r+1} = r_i, i = i + 1$

else {

if($\exists s \neq 0$ with $|s| \leq \lfloor N^2/2 \rfloor$ and $N^{i+3} \mid s(h_i - r_i\alpha)$)

set $(h_{i+1}, r_{i+1}) = s(h_i, r_i)$

else {

find s, t such that

(a) $(s, t) \neq (0, 0)$

(b) $|s|, |t| \leq \lfloor N^2/2 \rfloor$

(c) $N^{i+3} \mid s(h_i - r_i\alpha) + tN^{i-m}(h_m - r_m\alpha)$

set $(h_{i+1}, r_{i+1}) = s(h_i, r_i) + tN^{i-m}(h_m, r_m)$

} end of 'else'

if (1) $\Phi_N(h_{i+1}, r_{i+1}) > \Phi_N(h_i, r_i)$ and

(2) $\Phi_N(h_i, r_i) \leq i - m + \Phi_N(h_m, r_m)$ and

(3) $t \neq 0$

then set $m = i$

} end of 'ELSE'

} end of loop

Post work:

(1) find $\alpha = u/q$ from the equation $1 + N(u/q) = u^*/q^*$.

(2) reduce the output to u^*/q^* with $gcd(u^*, q^*) = 1$.

Figure 3.1: Rational Approximation Algorithm for N-FCSRs

Remarks: (1) The purpose of replacing α by $1 + N\alpha$ (i.e., shifting α) is to assure that the shifted sequence has its first element relatively prime to N.

(2) For $N = 3$, the bounds on the integers s, t inside the *loop* need to change to $[-5, 5]$. The reason of this change is explained later.

(3) Furthermore, when N is 2 or 3, we can modify the algorithm so that it is more efficient. The modified algorithms will be given after we prove the algorithm above.

We prove that the algorithm is correct and works similarly to the Berlekamp-Massey algorithm. Comparing with the original Berlekamp-Massey algorithm, the critical modification is that whenever a discrepancy occurs, 3 more elements in the sequence are processed (see line 4-5 and 7-11 inside the loop). The goal of the processing is to find a new rational number whose N-adic expansion matches all the processed elements and whose Φ-value is bounded and can be estimated by the previous values. Note that integer addition and multiplication have carries and N may be composite. The former makes the original B-M algorithm fails, and the latter makes Klapper and Goresky's lattice based approach fails.

3.3 Proof of the Rational Approximation Algorithm

Several lemmas are needed to prove the correctness and convergence of the algorithm. In the sequel, we state and prove the lemmas and explain why the algorithm works. First of all we have to ensure that the algorithm can make progress correctly at each index i.

Lemma 3.3.1 *The integer s under the second condition 'if' can be computed and checked efficiently.*

Proof: Suppose the integer s is a solution of the equation: $xb \equiv 0 \ (mod\ N^3)$, where $b = (h_i - r_i\alpha)/N^i \ mod\ N^3$. Let $d = gcd(b, N^3), b = b_0 d, N^3 = dL$. We then have: $xb_0 \equiv 0 \ (mod\ L)$. Equivalently, $x \equiv 0 \ (mod\ L)$ since $gcd(b_0, L) = 1$. Hence we only need to check if there is an integer k such that $|kL| \leq N^2/2$. This is true if and only if $L \leq N^2/2$. $\Box$

Lemma 3.3.2 *The integers s and t exist except when $N = 3$. For $N = 3$, the bounds need to be enlarged to be $[-5, 5]$. Further more when $N = 3$, we can choose s, t such that $|s| \neq 5$ or $|t| \neq 5$.*

Proof: For integers x and y, there are unique $w_0, w_1, w_2 \in S$ such that

$$x(b_i + b_{i+1}N + b_{i+2}N^2) + y(c_m + c_{m+1}N + c_{m+2}N^2) \equiv w_0 + w_1N + w_2N^2 (mod\ N^3).$$

Here,

$$b_i + b_{i+1}N + b_{i+2}N^2 = (h_i - r_i\alpha)/N^i\ mod\ N^3,$$

and

$$c_m + c_{m+1}N + c_{m+2}N^2 = (h_m - r_m\alpha)/N^m\ mod\ N^3.$$

Let $\Psi(x, y) = (w_0, w_1, w_2)$. We need to find suitable integers s and t such that $\Psi(s, t) = (0, 0, 0)$. First note that there are only N^3 distinct triples (w_0, w_1, w_2). There are several cases that need to be analyzed.

If $N = 3$, we choose x, y such that $-2 \leq x, y \leq 3$. Then we have $6^2 = 36$ distinct pairs (x, y), but only $N^3 = 27$ distinct triples (w_0, w_1, w_2). This shows that there are two different pairs $(x_1, y_1), (x_2, y_2)$ that map into the same number $w_0 + w_1N + w_2N^2\ (mod\ N^3)$. Then $(s, t) = (x_1, y_1) - (x_2, y_2) \neq (0, 0)$ is a non-zero pair such that $-5 \leq s, t \leq 5$ and $\Psi(s, t) = (0, 0, 0)$. Suppose $\Psi(s, t) = (0, 0, 0)$, but $|s| = |t| = 5$. We then have

$$N^3 | 5[(b_i + b_{i+1}N + b_{i+2}N^2) \pm (c_m + c_{m+1}N + c_{m+2}N^2)]$$

Since $gcd(N, 5) = 1$, this implies that

$$N^3 | [(b_i + b_{i+1}N + b_{i+2}N^2) \pm (c_m + c_{m+1}N + c_{m+2}N^2)]$$

and then we can choose $s, t \in \{1, -1\}$.

In the case when $N = 2k$ is even, we consider the map $\Psi(x, y)$ for $-N^2/4 \leq x, y \leq N^2/4$. Note that $\lfloor N^2/4 \rfloor = k^2$ and we have $(2k^2 + 1)^2 > N^3 = 8k^3$. Now similarly we have $(s, t) = (x_1, y_1) - (x_2, y_2) \neq (0, 0)$ such that $-\lfloor N^2/2 \rfloor \leq s, t \leq \lfloor N^2/2 \rfloor$ and $\Psi(s, t) = (0, 0, 0)$.

In the case when $N = 2k + 1$ with $k \geq 2$ is odd, we have $\lfloor N^2/4 \rfloor = k^2 + k$. We consider integers x, y such that $-(k^2 + k) \leq x, y \leq (k^2 + k)$. This gives us

$(2(k^2+k)+1)^2$ different pairs (x,y). Note that $k>1$, we have the inequality

$$(2(k^2+k)+1)^2-(2k+1)^3=2k[2k(k^2-1)-1]>0$$

It follows that the number of distinct such pairs (x,y) is greater the number of distinct triples (w_0,w_1,w_2). Again we get a pair (s,t) such that $-2(k^2+k)\leq s,t\leq 2(k^2+k)$ and $\Psi(s,t)=(0,0,0)$. Since $\lfloor N^2/2\rfloor=2(k^2+k)$, we are done. □

Notice that there may exist more than one such pair of (s,t), and we may select one such that $ind_N(s), ind_N(t)$ are minimized so that $\Phi(h_{i+1},r_{i+1})$ can be minimized. To find s and t efficiently, we may pre-compute a table. The number of entries in the table is at most N^4. When N is not very large, we may simply conduct an exhaustive search. Note that number s and t are not independent. When N is large, we can use the following lemma.

Lemma 3.3.3 *The numbers s and t can be computed in time $O(N^2\log^3 N)$.*

Proof: Let $a=\bar{a}_0+\bar{a}_1N+\bar{a}_2N^2$, $b=\bar{b}_0+\bar{b}_1N+\bar{b}_2N^2$ with $\bar{a}_0\neq 0,\bar{b}_0\neq 0$ and $0\leq\bar{a}_i,\bar{b}_i\leq N-1$. By Lemma 3.3.2, there exist s and t such that $sa+tb\equiv 0\ (mod\ N^3)$ and $|s|,|t|\leq N^2$. We now compute s and t by reducing the search space.

Let $u=gcd(N^3,a,b)$ and $K=N^3/u$. Then we have the equations

$$a=a'u,\quad b=b'u,\qquad sa'\equiv -tb'\ (mod\ K).$$

Let $d=gcd(K,a')$, $a'=a''d$ and $K=K_1d$. Then $gcd(d,b')=1$. This implies that $d|t$, and then $t=t_1d$ and $sa''=-t_1b'\ (mod\ K_1)$. Since $gcd(a'',K_1)=1$, there is an inverse v of a'' in $Z/(K_1)^*$ such that $v''a''+cK_1=1$ for some integer c. This gives us the equation

$$s\equiv -t_1b'v\ (mod\ K_1).$$

We then search for t_1 such that this equation holds and $|t_1d|,|s|\leq N^2$. Since greatest common divisor and division can be computed in time $O(\log^3 N)$ and $|t_1|$ is in the range $[0,N^2/d]$, s and t can be computed in time $O(N^2\log^3 N/d^2)$. □

Note that since $|b_1v\ (mod\ K_1)|<K_1\leq N^3$, we can build a table for any fixed K_1 which is a factor of N^3. Let $\tau(N^3)$ be an upper bound on the number of factors of N^3. Hence the number of tables is bounded by $\tau(N^3)$.

Before proceeding with the correctness proof, we explain why we need pre-work and post work in the algorithm. The purpose of pre-work is so the first pair (h_m, r_m) satisfies (a) $(h_m - \alpha r_m) \equiv 0 \ (mod\ N^m)$ but not $0 \ (mod\ N^{m+1})$, (b) There is no integer s with $|s| \leq N^2/2$ so that $N^3 | s(a_m + a_{m+1}N + a_{m+2}N^2)$. Here we assume a_m is the first non-zero bit and $h_m = 0$. By shifting the sequence by setting $\alpha \leftarrow 1 + N\alpha$, we have $a_0 = 1$. Then the pair $(0, 1)$ satisfies both (a) and (b) for the modified input sequence. Furthermore, the algorithm is designed to make both (a) and (b) true for indices i at which the combination updating (updating by using s and t) occurs. The reason to enforce these conditions is to guarantee that $s \neq 0$ and $t \neq 0$ whenever they are used, and then $r_i \neq 0$ for each pair (h_i, r_i) in the algorithm. We need the following definition.

Definition 3.3.1 *We define an index to be a turning point as follows:*

(1) The initial index m is a turning point;

(2) If m_1 is a turning point, $m_2 > m_1$ is the turning point following m_1 if

(a) $(h_{m_2} - \alpha r_{m_2}) \not\equiv 0 \ (mod\ N^{m_2+1})$

(b) there is no integer $s \neq 0$ such that

$$|s| \leq \lfloor N^2/2 \rfloor, \ \ N^{m_2+3} | s(h_{m_2} - \alpha r_{m_2})$$

(c) $\Phi_N(h_{m_2+1}, r_{m_2+1}) > \Phi_N(h_{m_2}, r_{m_2})$

(d) $\Phi_N(h_{m_2}, r_{m_2}) \leq (m_2 - m_1) + \Phi_N(h_{m_1}, r_{m_1})$

(e) m_2 is the smallest integer satisfying (a), (b), (c) and (d).

Note that 2(a) and 2(b) hold initially by the pre-work. Also note that an index i is a turning point if it is either the initial index m or it is one obtained from updating $m = i$ in the algorithm.

Lemma 3.3.4 *If $(h_{i+1}, r_{i+1}) = s(h_i, r_i) + tN^{i-m}(h_m, r_m)$ in the algorithm, then*

$$\Phi_N(h_{i+1}, r_{i+1}) \leq max(\Phi_N(h_i, r_i), i - m + \Phi_N(h_m, r_m)) + 2$$

Proof: Let $\mu = max(\Phi_N(h_i, r_i), i - m + \Phi_N(h_m, r_m))$ and $h_{i+1} = sh_i + tN^{i-m}h_m$. Note that $ind_N(h_{i+1}) = ind_N(|h_{i+1}|)$. When $N \neq 3$, we have that $|s|, |t| \leq \lfloor N^2/2 \rfloor \leq N^2/2$,

and then

$$|h_{i+1}| \leq (N^2/2)(N^{\mu+1}-1) + (N^2/2)(N^{\mu+1}-1) < N^{\mu+3}-1.$$

Hence $ind_N(z) \leq (\mu+2)$. We can do a similar computation for $r_{i+1} = sr_i + tN^{i-m}r_m$ and then the conclusion follows by the definition of Φ.

For the case $N = 3$, by Lemma 3.3.2, s, t are chosen such that $|s|, |t| \leq 5$, but not $|s| = |t| = 5$. In other words, one of $|s|$ and $|t|$ is less than 5. It follows that

$$|h_{i+1}| \leq 5(N^{\mu+1}-1) + 4(N^{\mu+1}-1) = 9(N^{\mu+1}-1) < N^{\mu+3}-1$$

The same argument is true for r_{i+1}, and this completes the proof.□

We say the algorithm is convergent at an index i if $h_i/r_i = \alpha = u/q$. The following observation assures that if (h_i, r_i) approximates α up to the i-th position, but $\Phi_N(h_i, r_i)$ is much smaller than i, then the algorithm is convergent at i.

Lemma 3.3.5 *If $i > \Phi_N(u,q) + \Phi_N(h_i, r_i) + 2$, then $h_i/r_i = u/q$.*

Proof: Note that $h_i/r_i - u/q = bN^i/qr_i$ for some b. If $b \neq 0$, then $\Phi_N(bN^i, qr_i) \geq i$. By Proposition 3.1.2, $\Phi_N(bN^i, qr_i) = \Phi_N(h_iq - r_iu, r_iq) \leq \Phi_N(r,q) + \Phi_N(h_i, r_i) + 2$. This is a contradiction, so $b = 0$. □

We must show that if the algorithm is not convergent at some index, then there is a larger index that is a turning point. Before updating (h_i, r_i) to (h_{i+1}, r_{i+1}) by using s and t, there are two cases that may cause the algorithm to converge. The first case is when the pair (h_i, r_i) to (h_{i+1}, r_{i+1}) is repeatedly updated by just increasing the index. In this case $\Phi_N(h_i, r_i)$ stays unchanged, but it approximates more and more bits. The above lemma says that if the number of these updatings is large enough, the algorithm converges. The second case is when (h_i, r_i) to (h_{i+1}, r_{i+1}) is updated by multiplying by an integer s. Then $\Phi_N(h_{i+1}, r_{i+1}) \leq \Phi_N(h_i, r_i)+2$. On the other hand, $h_{i+1} - \alpha r_{i+1} = s(h_i - \alpha r_i) \equiv 0 \ (mod\ N^{i+3})$. This shows that the Φ-value increases by at most 2, but it approximates at least 3 more bits. This implies that if this case repeats enough times, the algorithm converges. For convenience, we introduce some terminology.

In the algorithm, at an index i, if $h_i - \alpha r_i \equiv 0 \ (mod N^i)$ but not $(mod\ N^{i+1})$, then (h_{i+1}, r_{i+1}) is obtained either by multiplying by an integer s or as a combination

of (h_i, r_i) and (h_m, r_m) by using a pair of integers s and t. We call either such an i an *updating index*, with the former a *type 1 updating*, and the latter a *type 2 updating*. If a type 2 updating occurs under the condition $\Phi_N(h_i, r_i) \leq i - m + \Phi_N(h_m, r_m)$ and $\Phi_N(h_{i+1}, r_{i+1}) > \Phi_N(h_i, r_i)$, it is called a *turn-updating*. That is, i is the turning point next to m.

Before proceeding, recall that each pair (h_i, r_i) in the algorithm corresponds to a rational number h_i/r_i. Therefore we must show that r_i is never zero, otherwise the output from the algorithm would not correspond to an N-FCSR. We use the following three lemmas to prove this fact. We say two pairs (h_1, r_1) and (h_2, r_2) are Z-linearly independent if $x(h_1, r_1) + y(h_2, r_2) = (0, 0)$ implies that $x = y = 0$.

Lemma 3.3.6 *Let i be a type 2 updating index and (s, t) be the pair used in the combination. Then neither s nor t is zero.*

Proof: Recall that $N^{i+3} | s(h_i - \alpha r_i) + t(h_m - \alpha r_m) N^{i-m}$. If $s = 0$, then $N^{i+3} | t(h_m - \alpha r_m) N^{i-m}$ and $N^{m+3} | t(h_m - \alpha r_m)$. This is impossible because m is a turning-point. If $t = 0$, then $N^{i+3} | s(h_i - \alpha r_i)$. This is also impossible because i is a type 2 updating index. □

Lemma 3.3.7 *Let m be a turning point. For any index $i \geq (m+1)$ before the next turning point, (h_m, r_m) and (h_i, r_i) are Z-linearly independent. At any updating index i, $(h_{i+1}, r_{i+1}) \neq (0, 0)$.*

Proof: The proof is by induction. Note that at the initial stage, we have $(h_m, r_m) = (0, 1)$ and $(h_{m+1}, r_{m+1}) = (a_m N^m + a_{m+1} N^{m+1} + a_{m+2} N^{m+2}, 1 + N^{m+3})$ with $a_m \neq 0$. This shows that (h_m, r_m) and (h_{m+1}, r_{m+1}) are Z-linearly independent.

Suppose (h_m, r_m) and (h_i, r_i) are Z-linearly independent and i is an updating index. If i is a type 1 updating index, we have $(h_{i+1}, r_{i+1}) = s(h_i, r_i)$ with $s \neq 0$. So (h_m, r_m) and (h_{i+1}, r_{i+1}) are still Z-linearly independent. If i is a type 2 updating index, there are $s \neq 0$ and $t \neq 0$ such that

$$(h_{i+1}, r_{i+1}) = s(h_i, r_i) + t N^{i-m} (h_m, r_m).$$

Suppose there are x and y such that $x(h_{i+1}, r_{i+1}) + y(h_m, r_m) = (0, 0)$. Then

$$xs(h_i, r_i) + (xtN^{i-m} + y)(h_m, r_m) = (0, 0).$$

This implies that $xs = 0$ and $xtN^{i-m} + y = 0$. Since $s \neq 0$, it follows that $x = 0$, so $y = 0$. This shows that (h_m, r_m) and (h_{i+1}, r_{i+1}) are Z-linearly independent. In particular, $(h_{i+1}, r_{i+1}) \neq (0, 0)$. A similar argument shows that if i is a type 2 updating index, then (h_i, r_i) and (h_{i+1}, r_{i+1}) are Z-linearly independent. Since a new turning point is obtained only by type 2 updating, it follows that if i is a turning-point and (h_i, r_i) and (h_{i+1}, r_{i+1}) are updated into the new (h_m, r_m) and (h_{m+1}, r_{m+1}), then they are still Z-linearly independent. This completes the proof. □

Lemma 3.3.8 *At any updating index i, we have $ind_N(h_i) < i$.*

Proof: The proof is again by induction on i. Note that at the initial stage, $h_m = 0$. Hence $ind_N(h_m) < m$. Now suppose the lemma is true for every updating index $k \leq i$. We prove it is true for the next updating index. By the inductive hypothesis we have $ind_N(h_i) < i$ and $ind_N(h_m) < m$. Since $h_{i+1} = sh_i$ or $h_{i+1} = sh_i + tN^{i-m}h_m$, we have $ind_N(h_i) < i$ and $ind_N(N^{i-m}h_m) < i$. It follows that $|h_{i+1}| \leq N^2(N^{i+1} - 1) < N^{i+3}$ and $ind_N(h_{i+1}) < i+3$. On the other hand, we at least have $h_{i+1} = h_{i+2} = h_{i+3}$, hence we have $ind_N(h_{i+3}) < i + 3$. Since the next updating index $j \geq i + 3$ and $h_j = h_{i+1}$, this shows that at the next updating index j, $ind_N(h_j) = ind_N(h_{i+3}) < i + 3 \leq j$. □

Theorem 3.3.9 *For every j, $r_j \neq 0$.*

Proof: This is true initially. Since it is true for type 1 updating, we only need to consider the type 2 updating case with $j = i + 1$, i an updating index. We have that

$$(h_{i+1}, r_{i+1}) = s(h_i, r_i) + tN^{i-m}(h_m, r_m).$$

Suppose $r_{i+1} = 0$. Then $h_{i+1} = h_{i+1} - r_{i+1}\alpha \equiv 0 \ (mod\ N^{i+3})$. If $h_{i+1} \neq 0$, then $ind_N(h_{i+1}) \geq i + 3$. By Lemma 3.3.8, we have $ind_N(h_i) < i$ and, $ind_N(N^{i-m}(h_m, r_m)) < (i - m) + m = i$. It follows that $ind_N(h_{i+1}) \leq i + 2 < i + 3$. This contradiction shows that $h_{i+1} = 0$. Therefore $(h_{i+1}, r_{i+1}) = (0, 0)$. This contradicts Lemma 3.3.7. □

We know that at each step the pair (h_i, r_i) represents a rational number whose N-adic expansion approximates α up to the $(i - 1)$-th term. Next we show that if

the algorithm is not convergent up to index i, then there is such an index i that is a turning point. First we prove the following.

Lemma 3.3.10 *Let m be a turning point and i be an updating index before the next turning point. If $\Phi_N(h_i, r_i) \leq i - m + \Phi_N(h_m, r_m)$, then for every $j > i$ before the next turning point, $\Phi_N(h_j, r_j) \leq j - m + \Phi_N(h_m, r_m)$.*

Proof: We only need to consider the next updating index j after i. We have (1) $(h_{i+1}, r_{i+1}) = s(h_i, r_i)$ or (2) $(h_{i+1}, r_{i+1}) = s(h_i, r_i) + tN^{i-m}(h_m, r_m)$. Recall that an updating occurs if and only if $h_j - r_j\alpha \equiv 0 \ (mod\ N^i)$ but $\not\equiv 0 \ (mod\ N^{i+1})$. Note that $(h_{i+1}, r_{i+1}) = \cdots = (h_j, r_j)$ and $j - i \geq 3$.

In case (1), we have

$$\begin{aligned} \Phi_N(h_j, r_j) &= \Phi_N(h_{i+1}, r_{i+1}) \\ &\leq \Phi_N(h_i, r_i) + 2 \\ &\leq i - m + \Phi_N(h_m, r_m) + 2 \\ &\leq (j - m) + \Phi_N(h_m, r_m) \ . \end{aligned}$$

In case (2), we have $\Phi_N(h_{i+1}, r_{i+1}) \leq \Phi_N(h_i, r_i)$ because i is a type 2 updating index, but not a turning point. By the assumption, $\Phi_N(h_i, r_i) \leq i - m + \Phi_N(h_m, r_m)$. Therefore

$$\begin{aligned} \Phi_N(h_j, r_j) &= \Phi_N(h_{i+1}, r_{i+1}) \\ &\leq \Phi_N(h_i, r_i) \\ &\leq i - m + \Phi_N(h_m, r_m) \\ &\leq j - m + \Phi_N(h_m, r_m) \ . \end{aligned}$$

This completes the proof. □

An index $k > m$ is called *normal* if $\Phi_N(h_k, r_k) \leq (k - m) + \Phi_N(h_m, r_m)$. Thus we see that once a normal updating is reached, all further updating indices are normal until the next turning point.

Lemma 3.3.11 *Let m be a turning point, and let $\delta = \Phi_N(h_{m+1}, r_{m+1}) - \Phi_N(h_m, r_m)$. Then either the algorithm converges with no more turning points, or it will first reach*

a normal index k with no more than $\delta - 2$ updating, and then reach another turning point.

Proof: First note that at any updating index i, $h_{i+1} - \alpha r_{i+1} \equiv 0 \ (mod\ N^{i+3})$. In particular, since m is obtained by a type 2 updating, $h_{m+1} - \alpha r_{m+1} \equiv 0 \ (mod\ N^{m+3})$.

Let $i_{-1} = m = i_0 < i_1 < i_2 < \cdots < i_t$ be consecutive updating indices. Suppose they satisfy the conditions

$$\Phi_N(h_{i_j}, r_{i_j}) > i_j - m + \Phi_N(h_m, r_m), \quad 0 \le j \le t \ .$$

In other words $i_j, (j \ge 0)$, are all not normal. Let $u_j = i_j - i_{j-1}, 0 \le j \le t$. Then $u_j \ge 3$ and

$$i_t - m = \sum_{j=0}^{t} u_j \ge 3(1+t).$$

On the other hand, we have

$$(h_{i_j+1}, r_{i_j+1}) = \cdots = (h_{i_{j+1}}, r_{i_{j+1}}), \qquad 0 \le j \le t-1,$$

and

$$\Phi_N(h_{i_j+1}, r_{i_j+1}) \le max(\Phi_N(h_{i_j}, r_{i_j}), i_j - m + \Phi_N(h_m, r_m)) + 2.$$

Therefore $\Phi_N(h_{i_j+1}, r_{i_j+1}) \le \Phi_N(h_{i_j}, r_{i_j}) + 2$, $0 \le j \le t-1$. This implies that

$$\Phi_N(h_{i_t}, r_{i_t}) \le 2t + \Phi_N(h_{m+1}, r_{m+1}).$$

It follows that

$$\begin{aligned} & i_t - m + \Phi_N(h_m, r_m) - \Phi_N(h_{i_t}, r_{i_t}) \\ & \ge 3(t+1) + \Phi_N(h_m, r_m) - (\Phi_N(h_{m+1}, r_{m+1}) + 2t) \\ & = 3 + t - \delta. \end{aligned}$$

This shows that if $t = \delta - 3$, then i_t would be a normal index, so $t < \delta - 2$.

Once a normal index k is reached, by Lemma 3.3.10 all the updating indices $i \ge k$ are normal. Furthermore, either there is another turning point or the algorithm converges. □

If the difference $i - \Phi_N(h_i, r_i)$ is large enough, then by Lemma 3.3.5, the algorithm converges. Thus we want to bound $\Phi_N(h_i, r_i)$. In order to bound $\Phi_N(h_i, r_i)$, we have

to carefully estimate the increase at each updating. Let m and m_1 be consecutive turning points. We define the increase from m to m_1 as

$$\beta_{m_1} = \Phi_N(h_{m_1}, r_{m_1}) - \Phi_N(h_{m+1}, r_{m+1}).$$

At start, we set $\beta_m = 0$. Let k_m be the number of turning points less than m.

Lemma 3.3.12 *At any turning point m,*

$$\Phi_N(h_{m+1}, r_{m+1}) \leq (m+3) + 2k_m + \sum_{j \leq m} \beta_j - \Phi_N(h_m, r_m)$$

and

$$2k_m + \sum_{j \leq m} \beta_j \quad \leq \frac{2m}{3} \ .$$

Proof: Initially we have $k_m = 0$ and $\beta_m = 0$. Note that $\Phi_N(h_{m+1}, r_{m+1}) = m + 3$ and $\Phi_N(h_m, r_m) = 0$. Hence the lemma is true at the first turning point. Suppose it is true at turning point m and m_1 is the next turning point. Let $w + 1$ be the total number of updatings occurring up to m_1. Then we have

$$m_1 = m + u_0 + u_1 + \cdots + u_w,$$

with $u_i \geq 3$ the difference between the i-th and $(i+1)$-th updating. Since m_1 is a turning point, there exist s and t such that

$$(h_{m_1+1}, r_{m_1+1}) = s(h_{m_1}, r_{m_1}) + tN^{m_1-m}(h_m, r_m).$$

By the induction and the fact that $-\Phi_N(h_{m+1}, r_{m+1}) = \beta_{m_1} - \Phi_N(h_{m_1}, r_{m_1})$, we have

$$\begin{aligned}
\Phi_N(h_{m_1+1}, r_{m_1+1}) &\leq (m_1 - m) + 2 + \Phi_N(h_m, r_m) \\
&\leq (m_1 - m) + 2 + (m+3) + 2k_m + \sum_{j \leq m} \beta_j - \Phi_N(h_{m+1}, r_{m+1}) \\
&= (m_1 + 3) + 2(k_m + 1) + \sum_{j \leq m} \beta_j + \beta_{m_1} - \Phi_N(h_{m_1}, r_{m_1}) \\
&= (m_1 + 3) + 2k_{m_1} + \sum_{j \leq m_1} \beta_j - \Phi_N(h_{m_1}, r_{m_1}).
\end{aligned}$$

It is left for us to show the second estimate. It is true at the initial turning point. We assume that at the turning point m

$$3(k_m + \sum_{j \leq m} \beta_j/2) \leq m.$$

Note that $\beta_{m_1} = \Phi_N(h_{m_1}, r_{m_1}) - \Phi_N(h_{m+1}, r_{m+1}) \leq 2w$ and

$$\begin{aligned} m_1 &= m + u_0 + u_1 + \cdots + u_w \\ &\geq 3(k_m + \sum_{j \leq m} \beta_j/2) + u_0 + u_1 + \cdots + u_w \\ &\geq 3(k_m + \sum_{j \leq m} \beta_j/2) + 3 + 3w \\ &\geq 3(k_m + \sum_{j \leq m} \beta_j/2) + 3 + 3\beta_{m_1}/2 \\ &= 3(k_{m_1} + \sum_{j \leq m_1} \beta_j/2). \end{aligned}$$

Equivalently, we have the desired result

$$2k_{m_1} + \sum_{j \leq m_1} \beta_j \leq \frac{2m_1}{3} .$$

This completes the proof of Lemma 3.3.12. □

Let us define λ_m to be the smallest $\Phi_N(h, r)$ among all the pairs (h, r) with $h - \alpha r \equiv 0 \ (mod\ N^m)$. For the eventually periodic sequence $A = \{a_i \in S\}$ and $\alpha = \sum_{i=0}^{\infty} a_i N^i = u/q$ with u and q relatively prime, we define $\lambda(A) = \Phi_N(u, q)$.

Lemma 3.3.13 *Let m be a turning point and let (h, r) be associated with λ_{m+1}. Then we have*

$$\lambda_{m+1} \geq (m - 2) - \Phi_N(h_m, r_m) .$$

Proof: We have $h/r - h_m/r_m = (hr_m - rh_m)/rr_m$ with $hr_m - rh_m = bN^m \neq 0$ for some b. Then $\Phi_N(hr_m - rh_m, rr_m) \geq m$. On the other hand, we have

$$\Phi_N(hr_m - rh_m, rr_m) \leq \Phi_N(h, r) + \Phi_N(h_m, r_m) + 2.$$

Consequently, $\lambda_{m+1} = \Phi_N(h, r) \geq (m - 2) - \Phi_N(h_m, r_m)$ □

Lemma 3.3.14 *At any turning point m,*

$$\Phi_N(h_{m+1}, r_{m+1}) - \lambda_{m+1} \leq 5 + 2k_m + \sum_{j \leq m} \beta_j$$

Proof: By Lemma 3.3.12, we have that

$$\Phi_N(h_{m+1}, r_{m+1}) \leq (m+3) + 2k_m + \sum_{j \leq m} \beta_j - \Phi_N(h_m, r_m).$$

On the other hand, by Lemma 3.3.13, we have that

$$\lambda_{m+1} \geq (m-2) - \Phi_N(h_m, r_m).$$

Then we have that

$$\begin{aligned} \Phi_N(h_{m+1}, r_{m+1}) - \lambda_{m+1} &\leq (m+3) + 2k_m + \sum_{j \leq m} \beta_j - \Phi_N(h_m, r_m) - (m - 2 - \Phi_N(h_m, r_m)) \\ &= 5 + 2k_m + \sum_{j \leq m} \beta_j. \end{aligned}$$

This proves Lemma 3.3.14. □

We now are ready to state and prove the main theorem about the convergence of the algorithm. We assume the pre-work has been performed. That is, the first non-zero term in A is 1.

Theorem 3.3.15 *Let m be a turning point and $\alpha(A) = u/q$ with u and q relatively prime. Then when $m > 6(3 + \lambda)$, the algorithm is convergent at $m+1$. That is,*

$$\frac{h_{m+1}}{r_{m+1}} = \frac{u}{q}.$$

Therefore u/q can be recovered by removing the greatest common divisor of h_{m+1} and r_{m+1}.

Proof: We have

$$\frac{h_{m+1}q - r_{m+1}u}{r_{m+1}q} = \frac{bN^{m+1}}{r_{m+1}q}$$

for some integer b. If $b \neq 0$, we consider the Φ-value of the pair $(h_{m+1}q - r_{m+1}u, r_{m+1}q)$ $= (bN^{m+1}, r_{m+1}q)$. First we have the inequalities

$$\begin{aligned} \Phi_N(h_{m+1}q - r_{m+1}u, r_{m+1}q) &\leq 2 + \Phi_N(h_{m+1}, r_{m+1}) + \Phi_N(u, q) \\ &\leq 2 + 5 + 2k_m + \sum_{j \leq m} \beta_j + \lambda_{m+1} + \lambda \\ &\leq 7 + \frac{2m}{3} + 2\lambda. \end{aligned}$$

On the other hand, we have $\Phi_N(bN^{m+1}, r_{m+1}q) \geq (m+1)$. Now we see that when $m > 6(3+\lambda)$, the following inequality holds

$$m + 1 \leq 7 + \frac{2m}{3} + 2\lambda < m + 1.$$

This inequality is a contradiction. Thus $b = 0$ and the conclusion follows. □

We have shown that the algorithm converges in at most $6(3+\lambda)$ iterations. Note that in each iteration i, the computation time used depends on the value of $\Phi_N(h_i, r_i)$ that is always less than or equal to i. Since s and t can be computed in time $O(N^2 \log^3(N))$ and $|s|, |t| < N^2$, the time complexity of each iteration is $O(12(3+\lambda) + N^2 \log^3(N))$. Therefore we have the following

Corollary 3.3.16 *The time complexity of the rational approximation algorithm is*

$$O(6\lambda(12\lambda + N^2 \log^3(N))).$$

3.4 Register Construction

For any given eventually periodic sequence A over S, the rational approximation algorithm finds a rational representation $\alpha(A, N) = u/q$ with $gcd(u, q) = 1$ and $gcd(N, q) = 1$. The objective of this section is to explicitly construct an N-FCSR from $\alpha = u/q$ which generates A. As described in Section 2.1, in order to build an N-FCSR, we must have (1) register length r; (2) register taps: q_i $(0 \leq i \leq r)$; (3) initial memory M; and (4) initial loading $(a_0, a_1, \cdots, a_{r-1})$. The next theorem describes an algorithm which computes all these data.

Theorem 3.4.1 *Let $\alpha = u/q$ be a reduced rational number with integers $q > 0$ and $gcd(N, q) = 1$. Then we can construct an N-FCSR with length $r = ind_N(q + N)$ that generates an eventually periodic N-adic sequence whose corresponding N-adic number is $\alpha = u/q$.*

Proof: Consider the N-adic expansion of $q + N$:

$$q + N = b + q_1 N + q_2 N^2 + q_3 N^3 + \cdots + q_r N^r,$$

with, $0 \leq b, q_i < N, q_r \neq 0$ and $gcd(N, b) = 1$. Let $q_0 = -(N-b)$. Since $gcd(N, q) = 1$, $gcd(N, q_0) = 1$.

By Theorem 2.2.5, we need to find integers M and $\{a_0, a_1, ...a_{r-1}\}$ with $a_i \in S$ such that

$$\alpha = \frac{u}{q} = \frac{\sum_{n=0}^{r-1}(\sum_{i=0}^{n} q_i a_{n-i})N^n - MN^r}{q}. \qquad (3.5.1)$$

Step 1 Expand u in powers of N as follows

$$u = \sum_{i=0}^{r-1} b_i N^i + wN^r$$

with $b_i \in S$ and $w \in Z$.

Step 2 Compute $a_0 \in S$ such that $q_0 a_0 = b_0 + l_0 N$. For $0 < n \leq r-1$, do:

(1) $h_n = b_n - l_{n-1}$.

(2) compute $c_n \in S$ and $v_n \in Z$ by division such that

$$\sum_{i=1}^{n} q_i a_{n-i} - h_n = v_n N - c_n.$$

(3) compute $a_n \in S$ and $z_n \in Z$ such that $c_n = a_n q_0 + z_n N$.

(4) $l_n = v_n + z_n$.

Step 3 Set $M = l_r - w$ and output $a_i, 0 \leq i \leq r-1$.

To prove correctness, we consider the following equations:

$$\begin{aligned}
\sum_{n=0}^{r-1}(\sum_{i=0}^{n} q_i a_{n-i})N^n &= \sum_{n=0}^{r-1}(l_n N + h_n)N^n \\
&= \sum_{n=0}^{r-1}(l_n N - l_{n-1} + b_n)N^n \\
&= \sum_{i=0}^{r-1}(l_n N - l_{n-1})N^n + \sum_{n=0}^{r-1} b_n N^n \\
&= u - (w - l_r)N^r.
\end{aligned}$$

Hence $\sum_{n=0}^{r-1}(\sum_{i=0}^{n} q_i a_{n-i})N^n - (l_r - w)N^r = u$. □

Remark: If we start with an N-adic sequence $A = (a_0, a_1, a_2, \cdots)$, in Step 2 we do not need to compute $a_0, a_1, \cdots, a_{r-1}$.

Regarding register hardware implementation, the total size of an N-FCSR is the register length plus the register memory size. Assume a *hardware memory unit* can hold an integer between 0 and $N-1$. Then each a_i and each q_i can be stored in one such unit. To store the register memory M, we need $ind_N(M)+1$ units for $|M|$ and an extra unit for sign. Notice that the size of register memory varies when the register changes states. Fortunately, by Proposition 2.2.1, the register memory size is always bounded.

Corollary 3.4.2 *Let $\alpha = u/q$ be a reduced rational number with integers $q > 0$ and $gcd(N, q) = 1$. Let $q = q_0 + q_1 N + q_2 N^2 + \cdots + q_r N^r$ be the expansion of q with $-N < q_0 < 0$, $0 \le q_1, q_2, \cdots, q_{r-1} < N$, $0 < q_r < N$. Let w be the Hamming weight of $q - q_0$, the number of nonzero q_i, $1 \le i \le r$. Then an N-FCSR can be constructed with memory size:*

$$s \le max\{ind_N((w+1)(N-1)) + 2, ind_N(u) - r + 2\}.$$

Proof: Let M_0 be the initial memory, and M the memory at an arbitrary time. By the proof of Theorem 3.4.1 and equation (3.5.1), we have that

$$u = \sum_{n=0}^{r-1} (\sum_{i=0}^{n} q_i a_{n-i}) N^n - M_0 N^r.$$

Note that

$$\begin{aligned} M_0 N^r &= \sum_{n=0}^{r-1} \sum_{i=0}^{n} q_i a_{n-i} N^n - u \\ &= \sum_{i=0}^{r-1} q_i \sum_{n=i}^{r-1} a_{n-i} N^n - u. \end{aligned}$$

Let $t = \sum_{i=0} |q_i|$. We then have

$$|M_0 N^r| \le \sum_{i=0}^{r-1} |q_i| \sum_{n=i}^{r-1} |a_{n-i}| N^n + |u|$$

$$\begin{aligned} &\leq t\ (N-1)\sum_{n=0}^{r-1} N^n + |u| \\ &\leq t(N^r-1) + |u| \\ &\leq tN^r + |u|. \end{aligned}$$

It follows that

$$\begin{aligned} ind_N(M_0) &\leq ind_N(tN^r + |u|) - r \\ &\leq max\{r + ind_N(t), ind_N(u)\} + 1 - r \\ &= max\{ind_N(t) + 1, ind_N(u) - r + 1\}. \end{aligned}$$

By Proposition 2.2.1, if the register enters a period, then $|M| \leq w(N-1)$. Therefore, the memory size s is less than or equal to $max\{ind_N(M_0), w(N-1)\} + 1$. Note that since $t \leq (w+1)(N-1)$, we have that

$$\begin{aligned} s &\leq max\{ind_N(M_0), w(N-1)\} + 1 \\ &\leq max\{ind_N((w+1)(N-1)) + 1, ind_N(u) - r + 1, w(N-1)\} + 1 \\ &\leq max\{ind_N((w+1)(N-1)) + 2, ind_N(u) - r + 2\}. \end{aligned}$$

This completes the proof. □

Recall that for an eventually periodic N-adic sequence $A = (a_i : i \geq 0)$, there is a unique rational representation $\alpha = \alpha(A) = \sum_{i=0}^{\infty} a_i N^i = u/q$ with $q > 0$, $gcd(N, q) = 1$ and $gcd(u, q) = 1$. Also recall that $\lambda = \lambda(A) = \Phi(u, q)$. Note that in the above corollary, $r = ind_N(q + N)$. Hence $w \leq r \leq ind_N(q) + 1 \leq \lambda + 1$. We then have $(w+1)(N-1) \leq (\lambda+2)N$. By Proposition 3.1.1, $ind_N(\lambda+2)N = ind_N(\lambda+2) + 1$, it follows that $ind_N(r(w+1)(N-1)) \leq 2 \cdot ind_N(\lambda+2) + 2$. By Corollary 3.4.2, we have proved the following:

Theorem 3.4.3 *For an eventually periodic N-adic sequence $A = (a_i : i \geq 0)$, let $\lambda = \lambda(A)$ and $\alpha = \sum_{i=0}^{\infty} = u/q$ with $gcd(N, q) = 1$. Then an N-FCSR that outputs A can be constructed with size bounded by*

$$\lambda + ind_N(\lambda + 2) + 4.$$

As noticed before, the register setting with $q_0 = -1$ simplifies the register operations. If we are willing to sacrifice the register size by at most two, we can always make q_0 be -1.

Corollary 3.4.4 *Let $q = q_0 + q_1 N + \cdots + q_r N^r$ be the connection number with $q_0 < 0$ and $gcd(q_0, N) = 1$. Then there is a positive integer $x < N$ such that*

$$xq = -1 + c_1 N + c_2 N^2 + \cdots + c_u N^u, \ c_i \in S, c_u \neq 0, u \leq r + 1.$$

Proof: By elementary number theory there is an integer x such that $0 < x < N$ and $q_0 x = -1 + bN$ with $b < 0$. Then the conclusion follows. □

This corollary shows that the number of register taps increases by at most one. Since u is replaced by xu, the register memory size may also increase by one. Therefore, the total register size may increase by two.

To conclude this section, let us look an example. Let N = 10 and the input sequence be:

$$A = (2, 7, 9, 8, 5, 4, 9, 9, 3, 3, 7, 4, 5, 7, 7, 0, 6, 4, 1, 2, 8, 1, 2, 2, 6, 0, 9, 5, 5, 0).$$

We have implemented the rational approximation algorithm without doing shifting: $\alpha \to 1 + N\alpha$. With the input sequence, the program is stabilized at the 10-th input symbol. It outputs an integer pair:

$$(h, r) = (-689925600, 14713990200).$$

The GCD of h and r is 13267800. After removing the GCD, the pair is reduced to: $(u, q) = (-52, 1109)$. Therefore, the connection number is $q = 1109$, a prime number. By applying the register constructing algorithm, we have that $q = -1 + 10 + 10^2 + 10^3$. The register has three taps: $(1, 1, 1)$, the initial loading cells: $(a_2, a_1, a_0) = (9, 7, 2)$, and the initial memory $M = 0$. By running the constructed register, we generate the entire sequence which has period 1108:

2 7 9 8 5 4 9 9 3 3 7 4 5 7 7 0 6 4 1 2 8 1 2 2 6 0 9 5 5 0 2 8 0 1 0 2 3 5 0 9 4 4 8 7 0
7 5 3 6 5 5 7 8 1 8 8 8 5 3 8 7 9 5 3 9 8 1 0 1 3 4 8 5 8 2 7 8 8 4 2 6 3 2 2 8 2 3 4 0 8
2 1 2 6 9 7 3 1 3 8 2 4 5 2 2 0 5 7 2 5 5 3 4 3 1 9 3 4 7 5 7 0 4 2 7 3 3 4 1 9 4 5 9 9 4
4 9 8 2 1 3 7 1 2 1 5 8 4 8 1 5 5 2 3 1 7 1 0 9 0 0 0 1 1 2 4 7 3 5 6 5 7 9 2 0 3 6 9 8 4

3 7 5 6 9 1 8 9 9 7 7 5 1 5 2 9 6 8 4 0 4 9 3 7 0 2 0 3 5 8 6 0 6 3 0 0 4 4 8 6 9 4 1 6 2
0 9 1 1 2 5 8 5 9 3 9 2 6 8 7 2 9 9 1 1 3 6 0 0 7 7 4 9 1 6 7 5 9 2 8 0 2 1 4 7 2 4 4 1 0
6 7 3 7 8 9 5 4 0 1 6 7 4 8 0 4 3 8 5 7 1 5 4 1 1 7 9 7 4 2 5 2 0 8 0 9 7 7 4 0 3 8 1 3 3
8 4 6 9 0 7 7 5 0 4 0 5 9 4 9 3 8 1 4 4 0 9 3 3 6 3 3 3 0 7 0 8 5 4 8 8 1 9 9 0 0 1 2 3 6
1 1 9 1 2 3 7 2 3 3 9 5 8 3 8 0 3 2 6 1 0 8 9 7 5 3 7 6 7 1 6 5 3 5 4 3 3 1 8 2 2 3 8 3 5
7 6 9 3 0 4 8 2 5 6 4 6 7 8 2 9 0 3 3 7 3 4 5 3 3 2 9 4 6 0 2 9 1 3 4 9 6 0 7 4 2 4 1 8 3
3 5 2 1 9 2 3 5 1 0 7 8 5 1 6 3 1 1 6 8 5 0 5 1 7 3 2 3 9 4 7 1 4 3 9 6 9 5 2 8 6 7 2 7 7
7 2 8 8 9 6 5 2 5 3 1 0 5 6 1 3 1 6 0 8 4 3 6 4 4 5 4 4 4 3 2 0 6 8 4 9 2 7 9 9 6 6 3 7 7
8 3 0 3 7 0 1 9 0 1 1 3 5 9 7 2 0 1 4 5 0 0 6 6 2 5 4 2 2 9 3 5 8 7 1 8 7 7 3 9 0 4 4 9 7
1 9 8 9 7 6 4 9 0 5 5 1 2 9 2 4 6 3 4 4 2 1 8 1 1 1 4 6 1 2 0 4 6 0 1 8 9 8 6 5 1 4 1 7 2
1 1 5 7 3 6 7 7 1 7 6 5 9 1 7 8 7 3 0 2 6 8 6 1 7 5 4 7 7 9 4 2 7 4 4 6 5 6 8 0 6 5 2 4 2
9 5 7 2 6 6 5 8 0 5 4 0 0 5 5 0 1 7 8 6 2 8 7 8 4 1 5 1 8 4 4 7 6 8 2 8 9 0 9 9 9 8 8 7 5
2 6 4 3 4 2 0 7 9 6 3 0 1 5 6 2 4 3 0 8 1 0 0 2 2 4 8 4 7 0 3 1 5 9 5 0 6 2 9 7 9 6 4 1 3
9 3 6 9 9 5 5 1 3 0 5 8 3 7 9 0 8 8 7 4 1 4 0 6 0 7 3 1 2 7 0 0 8 8 6 3 9 9 2 2 5 0 8 3 2
4 0 7 1 9 7 8 5 2 7 5 5 8 9 3 2 6 2 1 0 4 5 9 8 3 2 5 1 9 5 6 1 4 2 8 4 5 8 8 2 0 2 5 7 4
7 9 1 9 0 2 2 5 9 6 1 8 6 6 1 5 3 0 9 2 2 4 9 5 9 4 0 5 0 6 1 8 5 5 9 0 6 6 3 6 6 6 9 2 9
1 4 5 1 1 8 0 0 9 9 8 7 6 3 8 8 0 8 7 6 2 7 6 6 0 4 1 6 1 9 6 7 3 8 9 1 0 2 4 6 2 3 2 8 3
4 6 4 5 6 6 8 1 7 7 6 1 6 4 2 3 0 6 9 5 1 7 4 3 5 3 2 1 7 0 9 6 6 2 6 5 4 6 6 7 0 5 3 9 7
0 8 6 5 0 3 9 2 5 7 5 8 1 6 6 4 7 8 0 7 6 4 8 9 2 1 4 8 3 6 8 8 3 1 4 9 4 8 2 6 7 6 0 5 2
8 5 6 0 3 0 4 7 1 3 2 7 2 2 2 7 1 1 0 3 4 7 4 6 8 9 4 3 8 6 8 3 9 1 5 6 3 5 5 4 5 5 5 6 7
9 3 1 5 0 7 2 0 0 3 3 6 2 2 1 6 9 6 2 9 8 0 9 8 8 6 4 0.

3.5 Distribution of N-adic complexities

Recall that an eventually periodic N-adic sequence is determined by a single period which is a finite sequence. There are various methods that can generate eventually periodic N-adic sequences. Our question is: for a random N-adic sequence of length n, what is the probability that the sequence can be generated by a small N-FCSR? To answer this question, we first introduce the following definitions.

Definition 3.5.1 *For an N-adic sequence $S_n = (a_0, ...a_{n-1})$ of length n, we say an integer pair (h, r) with $gcd(r, N) = 1$ is an approximation of S_n if*

$$h - r(a_0 + a_1 N + a_2 N^2 + ... + a_{n-1} N^{n-1}) \equiv 0 \ mod(N^n).$$

Since r is relatively prime to N, the rational number h/r can be expanded as an N-adic number. Therefore for an approximation (h,r), the first n bits of the N-adic expansion of the rational number h/r are just $a_0, a_1, a_2, ..., a_{n-1}$. Recall that $\Phi(h,r) = max\{ind_N(|h|), ind_N(|r|)\}$.

Definition 3.5.2 *The N-adic complexity of S_n, $\lambda(S_n)$ is the smallest $\Phi(h,r)$, where (h,r) is an approximation of S_n.*

Theorem: For $n \geq 2$, define

$$\mathcal{A} = \{S_n | \lambda(S_n) < \lfloor (n-2)/2 \rfloor\} \text{ and } \mathcal{B} = \{S_n | \lambda(S_n) \geq \lfloor (n-2)/2 \rfloor\}.$$

Then $|\mathcal{B}| \geq (N-1)N^{n-1}$.

Proof: Note that $\mathcal{A}$ does not intersect $\mathcal{B}$, hence $|\mathcal{A}| + |\mathcal{B}| = N^n$. Therefore we only need to show that $|\mathcal{A}| \leq N^{n-1}$.

For every j with $0 < j < N$, we can define a map from $\mathcal{A}$ into $\mathcal{B}$. Let $\phi_j(a) = a+j \ (mod\ N)$ for all $a : 0 \leq a < N$. Then ϕ_j is a bijection on the set $\{0, 1, 2, \cdots, N-1\}$ and it has no fixed points.

For any $S_n = a_0 a_1 ... a_{n-1} \in \mathcal{A}$, we have an integer pair (h,r) such that $\lambda(S_n) = \Phi(h,r)$ and

$$h - r(a_0 + a_1 N + a_2 N^2 + ... + a_{n-1} N^{n-1}) = bN^n.$$

for some $b \in Z$.

Define the map φ_j by

$$\varphi_j(a_0 a_1 \cdots a_{n-2} a_{n-1}) = a_0 a_1 ... a_{n-2} \phi_j(a_{n-1}).$$

Let j be fixed and let (u,q) be the integer pair corresponding to $\varphi_j(\bar{S}_n)$. Let $\lambda_j = \lambda(\varphi_j(\bar{S}_n)) = \Phi(u,q)$. We then have $ur - hq = N^{n-1}c$ for some non-zero integer c.

Since $ind_N(ur - hq) \leq \Phi(h,r) + \Phi(u,q) + 2$, we have

$$\Phi(h,r) + \Phi(u,q) \geq (n-1) - 2.$$

Let $n = 2k$ be even. Then $\lfloor (n-2)/2 \rfloor = k-1$. It follows that

$$\begin{aligned} \lambda_j &\geq (n-1) - 2 - \Phi(h,r) \\ &\geq (2k-1) - 2 - (k-1) + 1 \\ &\geq (k-1) \\ &= \lfloor (n-2)/2 \rfloor. \end{aligned}$$

If $n = 2k+1$ is odd, then $\lfloor (n-2)/2 \rfloor = k-1$, but

$$\begin{aligned} \lambda_j &\geq (n-1) - 2 - \Phi(h,r) \\ &\geq 2k - 2 - (k-1) + 1 \\ &\geq \lfloor (n-2)/2 \rfloor. \end{aligned}$$

This shows that for every $S_n \in \mathcal{A}$, $\varphi_j(S_n)$ is in $\mathcal{B}$, and for every fixed j, φ_j is one-to-one. Consequently, $\varphi_j(\mathcal{A}) \subset \mathcal{B} : 1 \leq j \leq N-1$.

Next we show that all the $\varphi_j(\mathcal{A})$ are disjoint. Suppose not. Then there exist two distinct indices j and k and two sequences $S_n = a_0 a_1 \cdots a_{n-1}$ and $S_n^{'} = b_0 b_1 \cdots b_{n-1} \in \mathcal{A}$ such that $\varphi_j(S_n) = \varphi_k(S_n^{'})$. That is,

$$a_0 a_1 \cdots a_{n-2} \phi_j(a_{n-1}) = b_0 b_1 \cdots b_{n-2} \phi_k(b_{n-1}).$$

This implies that $a_i = b_i, 0 \leq i \leq (n-2)$ and $a_{n-1} + j \equiv b_{n-1} + k \ (mod\ N)$. Note $S_n, S_n^{'}$ are both in $\mathcal{A}$. If $a_{n-1} \neq b_{n-1}$, we have two sequences:

$$\begin{aligned} S_n &= a_0 a_1 a_2 \cdots a_{n-2} a_{n-1} \\ S_n^{'} &= a_0 a_1 a_2 \cdots a_{n-2} b_{n-1} \end{aligned}$$

with the last symbol different. An argument similar to the above shows that $\lambda(S_n^{'}) \geq \lfloor (n-2)/2 \rfloor$ and then $S_n^{'} \notin \mathcal{A}$. This is impossible. Hence $S_n = S_n^{'}$. Consequently, $(j-k) \equiv 0 \ (mod\ N)$ and $j = k$. This contradiction implies that the sets $\varphi_j(\mathcal{A})$, $1 \leq j \leq N-1$ are disjoint. It follows that

$$|\mathcal{B}| \geq \sum_{j=1}^{N-1} |\varphi_j(\mathcal{A})| = (N-1)|\mathcal{A}|.$$

Table 3.1: Distribution of 2-Adic Complexities of Length ≤ 8

size/2-adic	0	1	2	3	4	5	6	7	total
1	2								2
2	3	1							4
3	3	4	1						8
4	3	8	4	1					16
5	3	8	16	4	1				32
6	3	8	34	14	4	1			64
7	3	8	38	60	14	4	1		128
8	3	8	38	130	58	14	4	1	256

Since $|\mathcal{A}| + |\mathcal{B}| = N^n$, this shows that $|\mathcal{A}| \leq N^{n-1}$ and $|\mathcal{B}| \geq (N-1)N^n$. $\square$

The result just proved says that if we randomly generate an N-adic sequence S of length n, the probability of getting one with N-adic complexity less than $(n-2)/2$ is less than or equal to $1/N$.

The following table displays the distribution of 2-adic complexities of binary sequences of length up to 8. The table was generated by an exhaustive computation.

Comparing this table with the linear complexity profile described by Rueppel in [35], there exist some similar characteristics. For instance, the number of sequences of length n with N-adic complexity equal to $\lceil n - 2/2 \rceil$ is greater than or equal to the half of the total. However, we are not able to prove this result yet.

3.6 Conclusions

We have described feedback with carry shift registers over $Z/(N)$ for any $N > 1$ (N-FCSR). Efficient algorithms have been developed and proved to solve the register synthesis problem for N-FCSRs. Cryptographically they provide a way to analyze sequences over $Z/(N)$ similarly to the way the Berlekamp-Massey algorithm and LFSRs can be used. Consequently any sequence over $Z/(N)$ used as a key stream in a stream cipher must have high N-adic complexity. It is an interesting open problem to find efficient devices that can generate sequences with large N-adic complexity.

Chapter 4

Algebraic Feedback Shift Registers

In this chapter, we first briefly discuss *algebraic feedback shift registers*(AFSRs) [24]. This class of registers is based on the algebra of π-adic numbers, where π is an element in a ring R, and produce sequences of elements in $R/(\pi)$. They generalize linear feedback shift registers over finite fields and feedback with carry shift registers over the rational integers. The main goal of this chapter is to present a solution to the register synthesis problem for certain AFSRs. We give several cases where the register synthesis problem can be solved by an efficient algorithm. Consequently, any keystreams over $R/(\pi)$ used in stream ciphers must be unable to be generated by a small register in this class. This extends the analysis of N-FCSRs developed in the previous chapters. In Section 1, after reviewing some concepts in commutative algebra, we give the definition of AFSRs. Then we state the main properties and characteristics of AFSRs. In Section 2, we describe a set of numerical conditions and a general rational approximation procedure. We prove that the procedure can be used to efficiently compute the rational representation of eventually periodic sequences over $R/(\pi)$ if the set of conditions are satisfied. In the remaining sections we present cases over which the set of conditions hold, and then the efficient rational approximation algorithms exist. These cases include AFSRs over the rational integers, polynomial rings over finite fields and certain rings whose fraction fields are quadratic number fields.

4.1 Algebraic Feedback Shift Registers

Let R be an integral domain which is a commutative ring with no zero divisors [17, 18]. Let F be its field of fractions. Let $\pi \in R$. The principal ideal generated by π is denoted $I = (\pi)$. We assume throughout that the quotient $K = R/(\pi)$ is finite, called the *residue ring of* (R, π). Since π is not necessary prime, the residue ring may have zero divisors. In general, K is isomorphic to a direct sum of finite many Galois fields.

Let S be a complete set of representatives for K in R. That is, for every element $a \in K$ there is a unique element $\alpha \in S$ that reduces to a modulo π. Then the set of power series

$$\sum_{i=0}^{\infty} a_i \pi^i, \quad a_i \in S, \tag{4.1}$$

forms a ring, $\hat{R}$. If $\cap_{i=0}^{\infty} I^n = (0)$ (that is, R is separable with respect to the I-adic topology), then there is an embedding of R in $\hat{R}$. We assume this throughout.

There is a well defined notion of the reduction of an element $\alpha \in \hat{R}$ modulo π. If α is

$$\alpha = \sum_{i=0}^{\infty} a_i \pi^i,$$

then the *reduction of* α *modulo* π is a_0. We also refer to

$$\sum_{i=0}^{\infty} a_{i+1} \pi^i$$

as the *integral quotient* of α by π, denoted $\mathrm{quo}(\alpha, \pi)$. Thus in general

$$\alpha = (\alpha \bmod \pi) + \pi \mathrm{quo}(\alpha, \pi).$$

Note that if $\alpha \in R$, then $\mathrm{quo}(\alpha, \pi) \in \mathrm{R}$.

Now let T be a second (possibly the same) complete set of representatives for K in R.

Definition 4.1.1 *An* algebraic feedback shift register *(or AFSR) over* (R, π, S, T) *of length* r *is specified by* $r+1$ *elements* $q_0, q_1, \cdots, q_r \in T$ *called the* taps, *with* q_0 *invertible modulo* π. *It is an automaton each of whose states consists of* r *elements* $a_0, a_1, \cdots, a_{r-1} \in S$ *and an element* $m \in R$ *(the* extra memory *or* carry*). The state is updated by the following steps.*

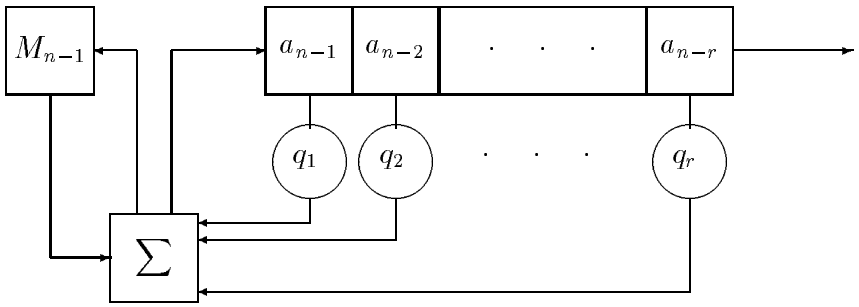

Figure 4.1: An AFSR Architectureafter

1. *Compute*
$$\tau = \sum_{i=1}^{r} q_i a_{n-i} + m.$$

2. *Find* $a_n \in S$ *such that* $q_0 a_r \equiv \tau \bmod \pi$.

3. *Replace* $(a_{n-r}, \cdots, a_{n-1})$ *by* $(a_{n-r+1}, \cdots, a_n)$ *and replace* m *by* $\mathrm{quo}(\tau - \mathrm{q_0 a_n}, \pi)$.

A diagram of an AFSR is given in Figure 4.1.

Such a device outputs an infinite sequence by repeatedly outputting the last element a_0 and changing states.

Example 1 Let $R = \mathbf{Z}$, the integer domain; $\pi = N > 1$, an integer; and $S = \{0, 1, 2, \cdots, N-1\}$. Then any AFSR under this setting is nothing but an N-FCSR defined in Chapter 2.

Example 2 Let R be a polynomial ring over the Galois field $GF(2)$ and $\pi = 1+x+x^2$, an irreducible polynomial. We then have the residue field $K = R/(\pi) \cong G(4)$ and a complete representative set $S = \{0, 1, x, 1+x\}$. Let $r = 3$, $(a_0, a_1, a_2) = (1, x, 0)$, $(q_0, q_1, q_2, q_3) = (1, x, 0, 1)$, and the initial memory $M = 0$.

Table 4.1 displays the state changes and outputs of the first 17 iterations of an AFSR over R with connection element $q = 1 + x\pi + \pi^3$, initial state $(1, x, 0)$ and initial memory $M = 0$. We see that the output enters a periodic part at the seventh iteration.

Table 4.1: Output of an x-AFSR

memory	q_1	q_2	q_3	output
	x	0	1	
0	1	x	1	1
0	1+x	1	x	x
1	1+x	1+x	1	1
1	1	1+x	1+x	1+x
0	0	1	1+x	1+x
0	1+x	0	1	1
1	**0**	**1+x**	**0**	**0**
0	1	0	1+x	1+x
0	1	1	0	0
0	x	1	1	1
1	x	x	1	1
1	1+x	x	x	x
1	x	1+x	x	x
1	0	x	1+x	1+x
0	1+x	0	x	x
1	1+x	1+x	0	0
1	**0**	**1+x**	**0**	**0**

Example 3 Let R be a polynomial ring over a Galois field F. Let $\pi = x$. Then $R/(\pi)$ is isomorphic to F. If we choose $S = T = F$ (a complete set of representatives of the residue field), then any AFSR over (R, π, S, T) with initial memory zero is nothing but a linear feedback shift register over F.

Example 4 Let $R = \mathbf{Z}[i]$, the Gaussian domain, and $\pi = 1 + i$. Then π is a prime element because the norm is 2. For an arbitrary element $w \in R$, $w = a + bi$ for some integers a and b. Note that $w = a + bi = (a - b) + b(1 + i) \equiv (a + b) \bmod \pi$ and $2 = (1 + i)(1 - i) \equiv 0 \bmod \pi$. It follows that $R/(\pi) \cong G(2)$. Then we can choose $S = \{0, 1\}$. Let $(a_2, a_1, a_0) = (1, 1, 1)$, $(q_0, q_1, q_2, q_3) = (1, 1, 0, 1)$ and the initial memory $m = 0$. By using the relation that $2 = \pi(2 - \pi)$, we can execute the register and produce a binary sequence. Table 4.2 displays the state changes and outputs for the index $j = 0, 1, 2, 3, 4, 5, 6$.

An AFSR is a finite state device provided that the extra memory takes on only finitely many values throughout an infinite execution. If this is the case, then the output is an eventually periodic sequence. In general, the extra memory may take

Table 4.2: Output of a π-AFSR

memory	q_1	q_2	q_3	output
	1	0	1	
0	1	1	1	1
$2-\pi$	0	1	1	1
$1-\pi$	1	0	1	1
$1-\pi$	1	1	0	0
$1-\pi$	0	1	1	1
$1-\pi$	1	0	1	1
$1-\pi$	1	1	0	0

infinitely many values. But in many interesting cases (such as in the examples above), memory values are confined within a finite set. In this dissertation, we are primarily concerned with AFSRs over polynomial rings over finite fields and AFSRs over rings whose fraction fields are finite extensions of the rational numbers (*number fields*).

We first consider AFSRs over polynomial rings.

Proposition 4.1.1 *For any AFSR with R a polynomial ring over a finite field and π being a polynomial (not necessary irreducible) with the degree $d > 0$. Let $t = max\{deg(s) : s \in S \cup T\}$. Suppose at a state the register has memory M of degree u. Then the following facts hold.*

(1) If $u \leq 2t$, then this remains true in all later states;

(2) If $u > 2t$, then the degree in the next state decreases at least by one.

Proof: Let M_{r-1} be the memory polynomial at the current state and M_r be that at the next state. By definition,

$$\begin{aligned} \sigma &= \sum_{i=0}^{r-1} a_i q_{r-i} + M_{r-1} \\ &= M_r \pi + a_r. \end{aligned}$$

If $u \leq 2t$, then the degree of σ is less than or equal to $2t$. Suppose $deg(M_r) > 2t$. Since $a_r \in S$ and $deg(a_r) \leq t$, the degree of $M_r\pi + a_r$ is $deg(M_r) + d > 2t$, a contradiction.

If $u > 2t$, then the degree of σ is u. Suppose $deg(M_r) \geq u$. We then have $deg(M_r\pi + a_r) = deg(M_r) + d > u$, a contradiction. $\Box$

The rings whose fraction fields are number fields are particularly important in both pure and computational algebraic number theory. The following conditions states when the memory of an AFSR over such rings is bounded.

Proposition 4.1.2 *[24] Suppose F is a finite extension of the rational numbers. If for every embedding of F in the complex numbers we have $|\pi| > 1$, then the memory in the infinite execution of any AFSR over F takes on only finitely many values. If there is an embedding of F in the complex numbers such that $|\pi| < 1$, then there is an AFSR whose memory grows unboundedly from some initial state.*

For an AFSR with taps $q_0, \cdots, q_r$, we call the following element

$$q = q_0 + q_1\pi + q_2\pi^2 + \cdots + q_r\pi^r$$

in R the *connection number*. We associate with any infinite sequence $A = (a_0, a_1, \cdots)$ over S the π-adic number

$$\alpha = \alpha(A, \pi) = \sum_{i=0}^{\infty} a_i\pi^i.$$

The following facts are similar to facts for N-FCSRs and can be proved similarly [24].

1. Suppose A is the output from an AFSR with connection number $q = q_0 + q_1\pi + \cdots + q_r\pi^r$ and initial extra memory m. Then the associated π-adic number is

$$\alpha = \frac{\sum_{n=0}^{r-1}(\sum_{i=0}^{n} q_i a_{n-i})\pi^n - m\pi^r}{q}. \tag{4.2}$$

2. Adding b to the memory adds $-b\pi/q$ to the output.

3. For any $u, q \in R$, with $q \not\equiv 0 \bmod \pi$, there is at most one AFSR over R, π, and S with connection element q, whose output corresponds to u/q.

4. Given a connection element

$$q = -q_0 + \sum_{i=1}^{r} q_i\pi^i$$

with $q_0, \cdots, q_r \in T$, and $u \in R$, there is an AFSR over R with output sequence A such that $\alpha(A, \pi) = u/q$. Furthermore, there is an efficient algorithm for constructing this AFSR.

4.2 Rational Approximation

The register synthesis problem for AFSRs is similar to that for N-FCSR. For a given sequence over S, given only a small prefix of the sequence, we would like to construct an AFSR of minimal size such that the register can generate the entire sequence. For an eventually periodic sequence $A = (a_i)$ over S with period L, there is an index $i_0 \geq 0$ such that $a_{i+L} = a_i$ for $i \geq i_0$. Let α be the π-adic number associated with A. Then we have that

$$\begin{aligned} \alpha &= \sum_{i=0}^{\infty} a_i \pi^i \\ &= \sum_{i=0}^{i_0-1} a_i \pi^i + \sum_{i \geq i_0} a_i \pi^i. \end{aligned}$$

Therefore,

$$\alpha = \sum_{i=0}^{i_0-1} a_i \pi^i + \pi^{i_0} \frac{\sum_{i=i_0}^{i_0+L-1} \pi^{i-i_0}}{1-\pi^L}.$$

This shows that an eventually periodic sequence has a rational representation. In order to construct an AFSR to generate an eventually periodic sequence A, we have to find the rational representation first. Thus the register synthesis problem for AFSRs can be solved if the following (loosely defined) problem can be solved.

Rational Approximation
Instance: A prefix of a sequence A.
Problem: Find elements $q_0, q_1, \cdots, q_r \in T$ and $u \in R$ such that

$$\alpha(A, \pi) = \frac{u}{-q_0 + \sum_{i=1}^{r} q_i \pi^i} = \frac{u}{q}.$$

We say this problem is loosely defined because there are many u and q_is that satisfy this equality, and it is not stated what minimality condition they should satisfy so that the resulting AFSR is minimal. Nor is it stated how large a prefix of the sequence should suffice for success. Minimality should mean that r is small, but it also depends on the representation of the extra memory. It would be attractive to represent the memory as a polynomial in π with coefficients in S, or perhaps in $S \cup -S$, but this is only possible in certain cases (for example, when R is the integers and $S = \{0, 1, \cdots, \pi - 1\}$). In what follows we use algebraically defined notions of minimality.

In many specific implementations of AFSRs reasonable bounds can be given relating these minimality notions and minimality for the specific implementations.

We may hope to improve the size of the generator by removing common factors from u and q. Such common factors can be found, for example, if R is a Euclidean domain. Such rings, however, are rare. For example, there is a small finite list of Euclidean domains R whose fraction fields are quadratic extensions of the rationals [6]. Furthermore, some multiples of q may have smaller π-adic expansions than that of q. An example is given in Section 4.5. Even if it were known that the minimal register arises when u and q are relatively prime, R might have an infinite unit group. In this case multiplying some u and q by the same unit will decrease the size of the register (there are infinitely many pairs (vu, vq) with v a unit, but only finitely many registers of any given size). Thus it must be stressed that the algorithm given here will give *some* generator of the given sequence (under suitable conditions), but not necessarily the *minimal* one.

Regarding the required size of the prefix, an algorithm solving the rational approximation problem will result in an effective register synthesis algorithm if any prefix whose length is polynomial in the size of the smallest AFSR that outputs A results in a correct rational representation of $\alpha(A, \pi)$. In the algorithm we present, the size of the required prefix is in fact always linear in the size of the smallest AFSR.

In this section we give a set of conditions on R under which a rational approximation algorithm exists. The algorithm is a modification of the Berlekamp-Massey algorithm (BM-algorithm) [32]. Recall that the idea of the BM-algorithm is to maintain a best rational approximation up to the jth term of the prefix at stage j. When a new symbol is processed, if the current best approximation no longer works (i.e., a "discrepancy" occurs), a combination of the current best approximation and a previous one results in a new best approximation. In the case of the Berlekamp-Massey algorithm, these are approximations to power series by rational functions (quotients of polynomials). A critical fact that makes the algorithm work is that the degree of the sum of two polynomials is at most the maximum of the degrees of the two polynomials. When there is a carry in addition and multiplication, this is false for reasonable analogues of degree. In order to control the growth of the approximations it is necessary to produce a new approximation that works for several new terms at once. To make this effective we need a measure of the size of elements of R that

increases in a controlled way when we perform various algebraic operations. Thus we assume we have a function $\phi_{R,\pi} : R \to \mathbf{Z} \cup \{-\infty\}$ satisfying the following properties.

Property 1: There are non-negative integers b and c such that

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

Here we use the convention that $-\infty + a = -\infty$ for every a. Such a function $\phi_{R,\pi}$ is called an *index function*. From it we define a function $\Phi_{R,\pi}$ on $R \times R$ by $\Phi_{R,\pi}(x, y) = \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y)\}$ for any $x, y \in R$.

Proposition 4.2.1 *For any two pairs* (h_1, r_1), $(h_2, r_2) \in R \times R$ *and integer* $k > 0$,

1. $\Phi_{R,\pi}(h_1 + h_2, r_1 + r_2) \leq \max\{\Phi_{R,\pi}(h_1, r_1), \Phi_{R,\pi}(h_2, r_2)\} + c$;
2. $\Phi_{R,\pi}(h_1 r_2 - r_1 h_2, r_1 r_2) \leq \Phi_{R,\pi}(h_1, r_1) + \Phi_{R,\pi}(h_2, r_2) + b + c$;
3. $\Phi_{R,\pi}(\pi^k(h_1, r_1)) = k + \Phi_{R,\pi}(h_1, r_1)$.

Suppose some AFSR over R and π has connection number $q = \sum_{i=0}^{r} q_i \pi^r$ with $q_i \in T$, and produces an output sequence whose associated π-adic number is $\alpha = u/q$, where u is given be equation (4.2). Then it follows from Property 1 that

$$\phi(q) \leq r + c \lceil \log(r) \rceil + e$$

and

$$\phi(u) \leq r + c + \max\{2c \lceil \log(r-1) \rceil + e + f + b, \phi(m)\},$$

where $e = \max\{\phi(x) : x \in T\}$, $f = \max\{\phi(x) : x \in S\}$, and m is the initial memory. In most cases $\phi(m)$ is a measure of the amount of memory required to store the memory. If this is the case, then $\Phi(u, q)$ is at most linear in the size of the AFSR. Thus if we can bound the execution time of a rational approximation algorithm in

terms of $\Phi(u, q)$, then we will have also bounded the execution time in terms of the size of the AFSR.

We also assume we have a finite subset $P_{R,\pi}$ of R such that the following properties hold.

Property 2 There are integers $B > C \geq 0$ such that

1. for every $s \in R$ if $s \in P_{R,\pi}$ then π^B does not divide s;
2. for any $h_1, h_2 \in R$, there exist $s, t \in P_{R,\pi}$ such that $\pi^B | sh_1 + th_2$;
3. for any $h_1, h_2 \in R$ and any $s, t \in P_{R,\pi}$, we have

$$\phi_{R,\pi}(sh_1 + th_2) \leq \max\{\phi_{R,\pi}(h_1), \phi_{R,\pi}(h_2)\} + C.$$

It follows that for any two pairs $(h_1, r_1), (h_2, r_2)$ and any $s, t \in P_{R,\pi}$, we have

$$\Phi_{R,\pi}(s(h_1, r_1) + t(h_2, r_2)) \leq \max\{\Phi_{R,\pi}(h_1, r_1), \Phi_{R,\pi}(h_2, r_2)\} + C.$$

Such a set $P_{R,\pi}$ is called an *interpolation set.* When there is no risk of ambiguity we drop the subscripts and simply write $\phi = \phi_{R,\pi}$, etc. With these definitions and properties, the rational approximation algorithm is given in Figure 4.2. In the remainder of this section we show that this algorithm solves the Rational Approximation Problem for any ring that has a function ϕ and predicate P satisfying Properties 1 and 2. In subsequent sections we give several examples of such rings.

At the start of the algorithm, we replace α by $1 + \pi\alpha$. The purpose is to guarantee that $(h_0 - \alpha r_0) \equiv 0$ modulo π^0 but not modulo π^1, and that there is no element $s \in R$ with $s \in P$ such that $\pi^B | s(h_0 - \alpha r_0)$.

We now proceed to show that the algorithm outputs a correct rational representation of α when enough bits are given. We start with some definitions that will make the explanation simpler.

At an index i, if $h_i - \alpha r_i \equiv 0 \ (mod \pi^i)$ but $h_i - \alpha r_i \not\equiv \ (mod\ \pi^{i+1})$, then (h_{i+1}, r_{i+1}) is obtained either by multiplying by an element $s \in R$ or as a combination of $s(h_i, r_i) + t(h_m, r_m)$. We call either such an i an updating index, with the former a *type 1*

Rational_Approximation

begin
input $A = \{a_i \in S, 0 \leq i \leq k\}$
$\alpha \longleftarrow 1 + \pi \sum_{i=0}^{k} a_i \pi^i$
$(h_0, r_0) \longleftarrow (0, 1)$
$(h_1, r_1) \longleftarrow (a_0 + \cdots + a_{B-1}\pi^{B-1}, 1 + \pi^B)$
$m \longleftarrow 0$
for $(i = m + 1$ to $k - 1)$
  if $((h_i - r_i\alpha) \not\equiv 0 \ mod(\pi^{i+1}))$ {
    if ($\exists s \neq 0 \in P \bigwedge(\pi^{i+B} \mid s(h_i - r_i\alpha)))$
      $(h_{i+1}, r_{i+1}) \longleftarrow s(h_i, r_i)$
    else {
      Find $s, t \in P$, not both zero, with
        $\pi^{i+B} \mid s(h_i - r_i\alpha) + t\pi^{i-m}(h_m - r_m\alpha)$
      $(h_{i+1}, r_{i+1}) \longleftarrow s(h_i, r_i) + t\pi^{i-m}(h_m, r_m)$
    }
    if ($\Phi(h_{i+1}, r_{i+1}) > \Phi(h_i, r_i)$ and $\Phi(h_i, r_i) \leq i - m + \Phi(h_m, r_m)$
      and $t \neq 0$)
      $m \longleftarrow i$
  }
output u, q with $1 + \pi(u/q) = h_k/r_k$
end

Figure 4.2: Rational Approximation Algorithm for AFSRs

updating, and the latter a *type 2 updating*. If a type 2 updating occurs under the condition $\Phi(h_i, r_i) \leq i - m + \Phi(h_m, r_m)$ and $\Phi(h_{i+1}, r_{i+1}) > \Phi(h_i, r_i)$, it is called a *turning-point*. We also call $i = 0$ a turning-point.

Recall that each pair (h_i, r_i) in the algorithm corresponds to a fraction $h_i/r_i \in Q(R)$. Therefore we must show that r_i is never zero. Otherwise the output from the algorithm would not correspond to an AFSR. We use the following three lemmas to prove this fact. We say two pairs $(h_1, r_1), (h_2, r_2)$ are R-linearly independent if $x(h_1, r_1) + y(h_2, r_2) = (0, 0)$ implies that $x = y = 0$.

Lemma 4.2.2 *Let m be a turning point, $i > m$ be a type 2 updating index, and (s, t) be the pair used in the combination. Then neither s nor t is zero.*

Proof: Recall that by Property 2 there exist s and t satisfying $\pi^{i+B}|s(h_i - \alpha r_i) + t(h_m - \alpha r_m)\pi^{i-m}$. If $s = 0$, then $\pi^{i+B}|t(h_m - \alpha r_m)\pi^{i-m}$, so $\pi^{m+B}|t(h_m - \alpha r_m)$. This is impossible because m is a turning-point. If $t = 0$, then $\pi^{i+B}|s(h_i - \alpha r_i)$. This is also impossible because i is a type 2 updating index. □

Lemma 4.2.3 *Let m be a turning point. For any index $i \geq (m+1)$ before the next turning point, (h_m, r_m) and (h_i, r_i) are R-linearly independent. At any updating index i, $(h_{i+1}, r_{i+1}) \neq (0,0)$.*

Proof: The proof is by induction. Note that at the initial stage $m = 0$, $(h_m, r_m) = (0,1)$, and $(h_{m+1}, r_{m+1}) = (a_0 + \cdots + a_{B-1}\pi^{B-1}, 1 + \pi^B)$ with $a_0 \neq 0$. This shows that (h_m, r_m) and (h_{m+1}, r_{m+1}) are R-linearly independent.

Suppose (h_m, r_m) and (h_i, r_i) are R-linearly independent and i is an updating index. If i is a type 1 updating index, we have $(h_{i+1}, r_{i+1}) = s(h_i, r_i)$ with $s \neq 0$. So (h_m, r_m) and (h_{i+1}, r_{i+1}) are still R-linearly independent. If i is a type 2 updating index, there are $s \neq 0$ and $t \neq 0$ such that

$$(h_{i+1}, r_{i+1}) = s(h_i, r_i) + t\pi^{i-m}(h_m, r_m).$$

Suppose there are $x, y \in R$ such that $x(h_{i+1}, r_{i+1}) + y(h_m, r_m) = (0,0)$. Then

$$xs(h_i, r_i) + (xt\pi^{i-m} + y)(h_m, r_m) = (0,0).$$

This implies that $xs = 0$ and $xt\pi^{i-m} + y = 0$. Since $s \neq 0$, it follows that $x = 0$, so $y = 0$. This shows that (h_m, r_m) and (h_{i+1}, r_{i+1}) are R-linearly independent. In particular, $(h_{i+1}, r_{i+1}) \neq (0,0)$. A similar argument shows that if i is a type 2 updating index, then (h_i, r_i) and (h_{i+1}, r_{i+1}) are R-linearly independent. Since a new turning point is obtained only by a type 2 updating, it follows that if i is a turning-point, so (h_i, r_i) and (h_{i+1}, r_{i+1}) are replaced by (h_m, r_m) and (h_{m+1}, r_{m+1}), then the new pairs are still R-linearly independent. This completes the proof. □

Lemma 4.2.4 *At any updating index i we have $\phi(h_i) < i$.*

Proof: The proof is again by induction on i. At the initial stage $h_m = 0$. Hence $\phi(h_m) < m$. Now we suppose $\phi(h_k) < k$ for every updating index $k \leq i$. We prove

the same is true for the next updating index. By the inductive hypothesis we have $\phi(h_i) < i$ and $\phi(h_m) < m$. Since $h_{i+1} = sh_i$ or $h_{i+1} = sh_i + t\pi^{i-m}h_m, \phi(h_i) < i$, and $\phi(\pi^{i-m}h_m) < i$, it follows from Property 2 that $\phi(h_{i+1}) \leq \max\{\phi(h_i), \phi(\pi^{i-m}h_m)\} + C < i + B$. On the other hand, we at least have $h_{i+1} = h_{i+2} = \cdots = h_{i+B}$, hence we have $\phi(h_{i+B}) < i + B$. Since the next updating index $j \geq i + B$ and $h_j = h_{i+1}$, this shows that at the next updating index j, $\phi(h_j) = \phi(h_{i+1}) < i + B \leq j$. □

Theorem 4.2.5 *For every j, $r_j \neq 0$.*

Proof: We prove this by induction. It is true initially. Since it is true for a type 1 updating, we only need to consider the type 2 updating case with $j = i + 1$, where i is an updating index. We have

$$(h_{i+1}, r_{i+1}) = s(h_i, r_i) + t\pi^{i-m}(h_m, r_m).$$

Suppose $r_{i+1} = 0$. Then $h_{i+1} = h_{i+1} - r_{i+1}\alpha \equiv 0 \ (mod\ \pi^{i+B})$. If $h_{i+1} \neq 0$, then $\phi(h_{i+1}) \geq i + B$. By Lemma 4.2.4, we have $\phi(h_i) < i$ and, $\phi(\pi^{i-m}(h_m, r_m)) < (i - m) + m = i$. It follows that $\phi(h_{i+1}) \leq i + C < i + B$. This contradiction shows that $h_{i+1} = 0$. Therefore $(h_{i+1}, r_{i+1}) = (0, 0)$. This contradicts Lemma 4.2.3. □

We say the algorithm is *convergent at index i* if $h_i/r_i = \alpha$. We should point out again that $\Phi(h_i, r_i)$ may not be minimized when the algorithm converges. The following lemma shows that if the pair (h_i, r_i) approximates α up to the i-th position, but $\Phi(h_i, r_i)$ is much smaller than i, then the algorithm is convergent at i. We may assume $\alpha = u/q$ with $\Phi(u, q)$ minimal among in the set of $\Phi(h, r)$ with $\alpha = h/r$.

Lemma 4.2.6 *If $i > \Phi(u, q) + \Phi(h_i, r_i) + b + c$, then $h_i/r_i = u/q$.*

Proof: Note that $h_i/r_i - u/q = x\pi^i/qr_i$ for some $x \in R$. If $x \neq 0$, then by Property 1 $\Phi(x\pi^i, qr_i) \geq i$. On the other hand, by Property 2 we have $\Phi(x\pi^i, qr_i) = \Phi(h_iq - r_iu, r_iq) \leq \Phi(r, q) + \Phi(h_i, r_i) + b + c$. When $i > \Phi(r, q) + \Phi(h_i, r_i) + b + c$ which is a contradiction. Hence $x = 0$ and $h_i/r_i = u/q = \alpha$. □

Besides updating (h_i, r_i) to (h_{i+1}, r_{i+1}) with a type 2 updating, there are two cases when this can be true. The first is when (h_{i+1}, r_{i+1}) equals (h_i, r_i), so $\Phi(h_i, r_i)$ is unchanged. Lemma 4.2.6 implies that if the number of such updatings is large enough,

the algorithm must have converged. The second case is when (h_{i+1}, r_{i+1}) is formed by a type 1 updating. In this case, it follows from Property 2 that $\Phi(h_{i+1}, r_{i+1}) \leq \Phi(h_i, r_i) + C$. However, $h_{i+1} - \alpha r_{i+1} = s(h_i - \alpha r_i) \equiv 0 \ (mod\ \pi^{i+B})$. This shows that the Φ-value increases by at most C, but it approximates at least B more symbols. If this case repeats enough, then the hypotheses of Lemma 4.2.6 must hold, so the algorithm must have converged.

Next we show that if the algorithm is not convergent at an index i then it eventually converges or reaches a turning point.

Lemma 4.2.7 *Let m be a turning point and let i be an updating index before the next turning point. If $\Phi(h_i, r_i) \leq i - m + \Phi(h_m, r_m)$, then $\Phi(h_j, r_j) \leq j - m + \Phi(h_m, r_m)$ for every $j > i$ before the next turning point.*

Proof: We only need to consider the next updating index j after i. We have (1) $(h_{i+1}, r_{i+1}) = s(h_i, r_i)$ or (2) $(h_{i+1}, r_{i+1}) = s(h_i, r_i) + t\pi^{i-m}(h_m, r_m)$. Recall that an updating occurs if and only if $h_j - r_j\alpha \equiv 0 \ (mod\ \pi^i)$ but $\not\equiv 0 \ (mod\ \pi^{i+1})$. Note that $(h_{i+1}, r_{i+1}) = \cdots = (h_j, r_j)$ and $j - i \geq B$, so $i + C < j$. In case (1) we have

$$\begin{aligned} \Phi(h_j, r_j) &= \Phi(h_{i+1}, r_{i+1}) \\ &\leq \Phi(h_i, r_i) + C \\ &\leq i - m + \Phi(h_m, r_m) + C \\ &\leq (j - m) + \Phi(h_m, r_m). \end{aligned}$$

In case (2) we have $\Phi(h_{i+1}, r_{i+1}) \leq \Phi(h_i, r_i)$ because i is a type 2 updating index but not a turning point. Therefore

$$\begin{aligned} \Phi(h_j, r_j) &= \Phi(h_{i+1}, r_{i+1}) \\ &\leq \Phi(h_i, r_i) \\ &\leq i - m + \Phi(h_m, r_m) \\ &\leq j - m + \Phi(h_m, r_m) \end{aligned}$$

remains true. □

An index $k > m$ is called *normal* if $\Phi(h_k, r_k) \leq (k - m) + \Phi(h_m, r_m)$. Thus the above lemma shows that once a normal updating is reached, all further updating indices are normal at least until the next turning point.

Lemma 4.2.8 *Let m be a turning point and $\delta = \Phi(h_{m+1}, r_{m+1}) - \Phi(h_m, r_m)$. Then either the algorithm converges with no more turning points, or it will first reach a normal index k with no more than $(\delta - C)/(B - C)$ updatings, and then reach another turning point.*

Proof: First note that, since m is obtained by a type 2 updating, $h_{m+1} - \alpha r_{m+1} \equiv 0 \ (mod\ \pi^{m+B})$.

Let $m = i_0 < i_1 < i_2 < \cdots < i_t$ be consecutive updating indices. Suppose that for $0 \leq j \leq t$ we have

$$\Phi(h_{i_j}, r_{i_j}) > i_j - m + \Phi(h_m, r_m).$$

In other words the i_j for $j \geq 1$ are all not normal. Let $u_j = i_j - i_{j-1}$ for $1 \leq j \leq t$. Then $u_j \geq B$ and $i_t - m = \sum_{j=1}^{t} u_j \geq Bt$. On the other hand, for $1 \leq j \leq t-1$ we have

$$(h_{i_j+1}, r_{i_j+1}) = \cdots = (h_{i_{j+1}}, r_{i_{j+1}}),$$

and

$$\Phi(h_{i_j+1}, r_{i_j+1}) \leq \max(\Phi(h_{i_j}, r_{i_j}), i_j - m + \Phi(h_m, r_m)) + C.$$

Therefore

$$\begin{aligned} (h_{i_{j+1}}, r_{i_{j+1}}) &= \Phi(h_{i_j+1}, r_{i_j+1}) \\ &\leq \Phi(h_{i_j}, r_{i_j}) + C. \end{aligned}$$

This implies that

$$\begin{aligned} \Phi(h_{i_t}, r_{i_t}) &\leq C(t-1) + \Phi(h_{i_1}, r_{i_1}) \\ &= C(t-1) + \Phi(h_{m+1}, r_{m+1}). \end{aligned}$$

It follows that

$$\begin{aligned} i_t - m + \Phi(h_m, r_m) - \Phi(h_{i_t}, r_{i_t}) &\geq Bt + \Phi(h_m, r_m) - (\Phi(h_{m+1}, r_{m+1}) + C(t-1)) \\ &= C + (B - C)t - \delta. \end{aligned}$$

This shows that if $t \geq (\delta - C)/(B - C)$, then i_t is a normal index. So $t < (\delta - C)/(B - C)$. Once a normal index k is reached, by Lemma 4.2.7 all the updating

indices $i \geq k$ are normal. It follows from Lemma 4.2.6 that either there is another turning point or the algorithm converges. □

If the difference $i - \Phi(h_i, r_i)$ is large enough, then by Lemma 4.2.6, the algorithm converges. Thus to bound the number of iterations the algorithm takes to converge, we must bound $\Phi(h_i, r_i)$. To bound $\Phi(h_i, r_i)$, we must carefully estimate the increase at each updating. Let m and m_1 be consecutive turning points. We define *the increase from m to m_1* as

$$\beta_{m_1} = \Phi(h_{m_1}, r_{m_1}) - \Phi(h_{m+1}, r_{m+1}).$$

At the start point, we can set $\beta_m = 0$. Let k_m be the number of turning points less than m. Let $D = B + c\lceil \log(B) \rceil + g$, where $g = \max\{\phi(x), \phi(1) : x \in S\}$. We then have that $\Phi(h_1, r_1) \leq F$. We now are ready to prove the next lemma.

Lemma 4.2.9 *At any turning point m*

$$\Phi(h_{m+1}, r_{m+1}) \leq (m + B) + Ck_m + \sum_{j \leq m} \beta_j - \Phi(h_m, r_m) + D$$

and

$$Ck_m + \sum_{j \leq m} \beta_j \leq \frac{Cm}{B}.$$

Proof: The proof is by induction. For the base case, $m = 0$, we have $k_0 = 0$, $\beta_0 = 0$, $\Phi(h_1, r_1) \leq D < D + B$, and $\Phi(h_0, r_0) = 0$. Thus the lemma is true at the first turning point.

Suppose the lemma is true at a turning point m and m_1 is the next turning point. Let $w + 1$ be the total number of updatings occurring up to m_1. Then we have

$$m_1 = m + u_0 + u_1 + \cdots + u_w,$$

with $u_i \geq B$ the difference between the i-th and $(i + 1)$-st updatings. Since m_1 is a turning point, there exist s and t such that

$$(h_{m_1+1}, r_{m_1+1}) = s(h_{m_1}, r_{m_1}) + t\pi^{m_1 - m}(h_m, r_m).$$

By induction and the fact that $-\Phi(h_{m+1}, r_{m+1}) = \beta_{m_1} - \Phi(h_{m_1}, r_{m_1})$, we have

$$\begin{aligned}
\Phi(h_{m_1+1}, r_{m_1+1}) &\leq (m_1 - m) + C + \Phi(h_m, r_m) \\
&\leq (m_1 - m) + C + (m + B) + Ck_m + \sum_{j \leq m} \beta_j - \Phi(h_{m+1}, r_{m+1}) + D \\
&= (m_1 + B) + C(k_m + 1) + \sum_{j \leq m} \beta_j + \beta_{m_1} - \Phi(h_{m_1}, r_{m_1}) + D \\
&= (m_1 + B) + Ck_{m_1} + \sum_{j \leq m_1} \beta_j - \Phi(h_{m_1}, r_{m_1}).
\end{aligned}$$

It remains to show the second inequality. It is true at the initial turning point. We assume that at a turning point m

$$BCk_m + B(\sum_{j \leq m} \beta_j) \leq Cm \ .$$

We have $\beta_{m_1} = \Phi(h_{m_1}, r_{m_1}) - \Phi(h_{m+1}, r_{m+1}) \leq Cw$ and

$$\begin{aligned}
Cm_1 &= Cm + C(u_0 + u_1 + \cdots + u_w) \\
&\geq BCk_m + B(\sum_{j \leq m} \beta_j) + C(u_0 + u_1 + \cdots + u_w) \\
&\geq BCk_m + B(\sum_{j \leq m} \beta_j) + BC(w + 1) \\
&\geq BCk_m + B(\sum_{j \leq m} \beta_j) + BC + B\beta_{m_1} \\
&= BCk_{m_1} + B(\sum_{j \leq m_1} \beta_j).
\end{aligned}$$

Equivalently, we have the desired result

$$Ck_{m_1} + \sum_{j \leq m_1} \beta_j \leq \frac{Cm_1}{B},$$

which completes the proof. □

Let λ_m be the smallest $\Phi(h, r)$ with $h - \alpha r = 0 \pmod{\pi^m}$. For the eventually periodic sequence $A = a_0, a_1, \cdots$, with $a_i \in S$, and $\alpha = \sum_{i=0}^{\infty} a_i \pi^i = u/q$ with $\Phi(u, q)$ minimal, we define $\lambda(A) = \Phi(u, q)$.

Lemma 4.2.10 *If m is a turning point, then*

$$\lambda_{m+1} \geq (m - b - c) - \Phi(h_m, r_m).$$

Proof: Let $h - \alpha r \equiv 0 \pmod{\pi^{m+1}}$ and $\lambda_{m+1} = \Phi(h, r)$. Then $(h, r) \neq (h_m, r_m)$. We have

$$\begin{aligned} \frac{h}{r} - \frac{h_m}{r_m} &= \frac{hr_m - rh_m}{rr_m} \\ &= \frac{x\pi^m}{rr_m} \end{aligned}$$

for some $x \neq 0 \in R$. Therefore $\Phi(hr_m - rh_m, rr_m) \geq m$. On the other hand, we have

$$\Phi(hr_m - rh_m, rr_m) \leq \Phi(h, r) + \Phi(h_m, r_m) + b + c.$$

Consequently, $\lambda_{m+1} = \Phi(h, r) \geq (m - b - c) - \Phi(h_m, r_m)$. □

We now are ready to state and prove the main theorem on the convergence of the algorithm.

Theorem 4.2.11 *Let i be any index and $\alpha(A) = u/q$ with $\Phi(u, q)$ minimal. Then when*

$$i > \frac{B(2(b + c) + B) + D}{B - C} + \frac{2B}{B - C}\lambda,$$

the algorithm is convergent at i. That is,

$$\frac{h_i}{r_i} = \frac{u}{q}.$$

Proof: By Lemma 4.2.6 it suffices to show that $i > b + c + \lambda + \Phi(h_i, r_i)$. Let m be the last turning point before i, let $t = i - m - 1$, and let w be the number of updatings between m and i. Thus $w \leq t/B$. Then

$$\begin{aligned} b + c + \lambda + \Phi(h_i, r_i) &\leq b + c + \lambda + \Phi(h_{m+1}, r_{m+1}) + Cw \\ &\leq b + c + \lambda + m + B + \frac{Cm}{B} - \Phi(h_m, r_m) + Cw + D \\ &\leq b + c + \lambda + B + \frac{Cm}{B} + \lambda_{m+1} + b + c + Cw + D \\ &\leq 2(b + c) + B + 2\lambda + \frac{Cm}{B} + \frac{Ct}{B} + D \\ &= 2(b + c) + B + 2\lambda + \frac{C}{B}(i - 1) + D, \end{aligned}$$

where the second line follows from Lemma 4.2.9 and the third line follows from Lemma 4.2.10. It follows that $b + c + \lambda + \Phi(h_i, r_i) < i$ if

$$2(b + c) + B + 2\lambda + D \leq \frac{B - C}{B}(i - 1).$$

This is equivalent to the hypotheses on i in the statement of the theorem. □

4.3 Rational Approximation over Z

We now apply the general rational approximation algorithm to N-FCSRs. We view any N-FCSR as an AFSR over the ring $R = \mathbf{Z}$, the ordinary integers. In this case $\pi > 1$ is an integer (possibly composite). Let $S = \{a : 0 \leq a \leq \pi - 1\}$. If $x \neq 0$ and $|x| = a_0 + a_1\pi + \cdots + a_t\pi^t$ with $a_i \in S$ and $a_t \neq 0$, then we define $\phi_{\mathbf{Z},\pi}(x) = t$. Equivalently, $\phi_{\mathbf{Z},\pi}(x) = t$ if $\pi^t \leq |x| < \pi^{t+1}$. Then by Proposition 3.1.1, Property 1 holds with $b = 1$ and $c = 1$. We also define

$$x \in P_{\mathbf{Z},\pi} \text{ if } |x| \leq \begin{cases} \lfloor \pi^2/2 \rfloor & \text{if } \pi \geq 4 \\ 5 & \text{if } \pi = 3. \end{cases}$$

Then by Lemma 3.3.1 and 3.3.2, Property 2 holds with $B = 3$ and $C = 2$. It follows that the Rational Approximation Algorithm converges in $22 + 6\lambda$ steps.

Suppose m is the initial memory of an AFSR over $\mathbf{Z}$, π. If $\phi_{\mathbf{Z},\pi}(m) \leq k$, then the m can be represented by $k + 1$ elements of S plus one sign bit. Thus, by the discussion following Proposition 4.2.1, the number of symbols of the output sequence needed to synthesize an equivalent AFSR is at most linear in the size of the smallest AFSR that generates the sequence.

4.4 Rational Approximation for AFSRs over Polynomial Rings

Let F be a finite field and R=F[x] be the polynomial ring over F. Let π be any polynomial in R and $d > 0$ be the degree of π. Note that the residue ring K = R/(π) has finitely many elements and it may have zero divisors. Let S = $\{a \in R : deg(a) < d\}$ be a complete representative set of K. Over these settings, we can

construct AFSRs and output sequences are sequences over S. In particular, if π is an irreducible polynomial, the output sequences can be interpreted as sequences over an extension of field F.

For any polynomial $g(x)$, by polynomial division $g(x) = g'(x)\pi + s(x)$ with $deg(g') < deg(g)$ and $s(x) \in$ S. Hence, a polynomial $f(x) \in$ R can be uniquely expanded in terms of powers of π with coefficients in S. If $f(x) \neq 0$ and $f(x) = a_0 + a_1\pi + \cdots + a_t\pi^t$ with $a_i \in$ S and $a_t \neq 0$, we then define $\phi_{\mathrm{R},\pi}(f(x)) = t = \lfloor deg(f)/d \rfloor$, which we call the degree of $f(x)$ relative to π. Note that summation of two polynomials does not increase degrees. That is, $\phi_{\mathrm{R},\pi}(f(x)+g(x)) \leq \max\{\phi_{\mathrm{R},\pi}(f(x)), \phi_{\mathrm{R},\pi}(g(x))\}$. For multiplication of two polynomials $f(x)$ and $g(x)$, we have that

$$\phi_{\mathrm{R},\pi}(f(x)g(x)) \leq \phi_{\mathrm{R},\pi}(f(x)) + \phi_{\mathrm{R},\pi}(g(x)) + 1.$$

This implies that Property 1 holds with $b = 1$ and $c = 0$. We also define

$$f(x) \in P_{\mathrm{R},\pi} \text{ if } \deg(f(x)) \leq d.$$

In order to construct a rational approximation algorithm under the setting above, we must find the constants B and C such that Property 2 holds.

Proposition 4.4.1 *Let $a(x), b(x), c(x), d(x) \in$ S with $a(x) \neq 0$ and $b(x) \neq 0$. Then there are polynomials $u(x), v(x) \in P_{\mathrm{R},\pi}$ not both zero such that*

$$\pi^2 | u(x)(a(x) + c(x)\pi) + v(x)(b(x) + d(x)\pi).$$

Proof: For any pair $(u(x), v(x)) \in$ S $\times$ S, let

$$w(x) \equiv u(x)(a(x) + c(x)\pi) + v(x)(b(x) + d(x)\pi) \bmod (\pi^2) \text{ with } \phi_{\mathrm{R},\pi}(\mathrm{w}) \leq 1.$$

Let $\Omega = |F|$, the size of the field. Then we have at most Ω^{2d} different such $w(x)$. Note that $|\mathrm{S} \times \mathrm{S} - (0,0)| = \Omega^{2(d+1)} - 1$. This implies that there are at least two different nonzero pairs $(u_1(x), v_1(x)), (u_2(x), v_2(x))$ such that

$$(u_1(x) - u_2(x))(a(x) + c(x)\pi) + (v_1(x) - v_2(x))(b(x) + d(x)\pi) \equiv 0 \bmod (\pi^2).$$

This completes the proof. □

Proposition 4.4.2 *For any* $f(x), g(x) \in \mathrm{R}$ *and* $u(x), v(x) \in P_{\mathrm{R},\pi}$, *we have*

$$\phi_{\mathrm{R},\pi}(u(x)f(x) + v(x)g(x)) \leq \max\{\phi_{\mathrm{R},\pi}(f(x)), \phi_{\mathrm{R},\pi}(g(x))\} + 1.$$

Proof: Any polynomial in $P_{\mathrm{R},\pi}$ has degree at most d. Then, $deg(u(x)f(x) + v(x)g(x)) \leq \max\{deg(f), deg(g)\} + d$. Therefore,

$$\begin{aligned} \phi_{\mathrm{R},\pi}(uf + vg) &\leq \left\lfloor \frac{\max\{deg(f), deg(g)\}}{d} \right\rfloor + 1 \\ &\leq \max\{\left\lfloor \frac{deg(f)}{d} \right\rfloor, \left\lfloor \frac{deg(g)}{d} \right\rfloor\} + 1 \\ &= \max\{\phi_{\mathrm{R},\pi}(f), \phi_{\mathrm{R},\pi}(g)\} + 1. \end{aligned}$$

This proves the result. □

By Proposition 4.4.1 and 4.4.2, we see that Property 2 holds with $B = 2$ and $C = 1$. This implies that the rational approximation algorithm converges in $13 + 4\lambda$ steps.

One special case is when $\pi = x$. In this case, S equals F. If q_0 is chosen to be -1 and the initial extra memory is chosen to be 0, then the resulting AFSR is just a regular LFSR over F. We see that we can choose $b = 0$, $c = 0$, $B = 1$, and $C = 0$ so that Property 1 and 2 holds. This leads to the BM-algorithm, which converges in $2\lambda + 3$ steps.

4.5 Rational Approximation for Ramified Extensions

Let Q be a ring, $\tau \in Q$, S a complete set of residues modulo τ, and suppose we have an index function $\phi_{Q,\tau}$ and interpolation set $P_{Q,\tau}$ with respect to τ. Let b, c, B, and C be the constants in Properties 1 and 2 with respect to $\phi_{Q,\tau}$ and $P_{Q,\tau}$.

Let d be a positive integer and $\epsilon = \pm 1$. Assume that the polynomial $X^d - \epsilon\tau$ is irreducible over Q, and π is a root of this polynomial. In this section we consider the case when

$$R = Q[\pi] = \left\{ \sum_{i=0}^{d-1} a_i \pi^i : a_i \in Q \right\}.$$

We have $R/(\pi) = Q/(\tau)$, so S is a complete set of representatives for R modulo π as well.

For any $x = \sum_{i=0}^{d-1} a_i \pi^i, a_i \in Q$, we define

$$\phi_{R,\pi}(x) = \max\{d\phi_{Q,\tau}(a_i) + i : 0 \leq i \leq d-1\}.$$

This is well defined because, by the irreducibility of $X^d - \epsilon\tau$, this representation of x is unique.

Proposition 4.5.1 *For any $x \in R$ and non-zero integer k, $\phi_{R,\pi}(\pi^k x) = k + \phi_{R,\pi}(x)$.*

Proof: If $x = \sum_{i=0}^{d-1} a_i \pi^i$, let $w = \max\{\phi_{Q,\tau}(a_i) : 0 \leq i \leq d-1\}$ and let j be the largest index such that $w = \phi_{Q,\tau}(a_j)$. Then $\phi_{R,\pi}(x) = dw + j$. Letting $\pi x = x' = \sum_{i=0}^{d-1} a'_i \pi^i$, with $a'_i \in Q$, we have $a'_0 = \epsilon a_{d-1}\tau$ and $a'_i = a_{i-1}$ for $1 \leq i \leq d-1$. Let $w' = \max\{\phi_{Q,\tau}(a'_i) : 0 \leq i \leq d-1\}$ and let j' be the largest index such that $w' = \phi_{Q,\tau}(a'_j)$. If $j < d-1$, then $w' = w$ and $j' = j+1$. Hence $\phi_{R,\pi}(x') = dw' + j' = dw + j + 1 = \phi_{R,\pi}(x) + 1$. If $j = d-1$, then $w' = w+1$, $j' = 0$, and $\phi_{R,\pi}(x') = dw' + j' = dw + d = (dw + j - 1) + 1 = \phi_{R,\pi}(x) + 1$. □

Proposition 4.5.2 *For any $x, y \in R$, $\phi_{R,\pi}(x \pm y) \leq \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y)\} + cd$.*

Proof: Let

$$x \stackrel{\text{def}}{=} \sum_{i=0}^{d-1} a_i \pi^i \text{ and } y = \sum_{i=0}^{d-1} b_i \pi^i,$$

with $a_i, b_i \in Z$. Then

$$\begin{aligned} z &= x \pm y \\ &= \sum_{i=0}^{d-1} (a_i \pm b_i)\pi^i \\ &= \sum_{i=0}^{d-1} c_i \pi^i, \end{aligned}$$

with $c_i = a_i \pm b_i$. Since

$$\phi_{Q,\tau}(c_i) \leq \max\{\phi_{Q,\tau}(a_i), \phi_{Q,\tau}(b_i) : 0 \leq i \leq d-1\} + c,$$

it follows that

$$\begin{aligned}\phi_{R,\pi}(z) &= \max\{d\phi_{Q,\tau}(c_i)+i : 0 \le i \le d-1\} \\ &\le \max\{d\phi_{Q,\tau}(a_i)+i, d\phi_{Q,\tau}(b_i)+i : 0 \le i \le d-1\} + cd \\ &= \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y) : 0 \le i \le d-1\} + cd.\end{aligned}$$

This proves the proposition. □

Proposition 4.5.3 *Let $b' = cd\lceil\log(d)\rceil + bd$. Then for any $x, y \in R$, $\phi_{R,\pi}(xy) \le \phi_{R,\pi}(x) + \phi_{R,\pi}(y) + b'$.*

Proof: As before let $x = \sum_{i=0}^{d-1} a_i\pi^i$ and $y = \sum_{i=0}^{d-1} b_i\pi^i$ with $a_i, b_i \in Q$. Then $z = xy = \sum_{i=0}^{d-1} c_i\pi^i$ with

$$c_i = \left(\sum_{l=0}^{i} a_l b_{i-l}\right) + \epsilon\tau\left(\sum_{l=i+1}^{d-1} a_l b_{d+i-l}\right).$$

Thus

$$\begin{aligned}\phi(R,\pi(xy)) &= \max\left\{d\phi_{Q,\tau}\left(\sum_{l=0}^{i} a_l b_{i-l}\right) + \epsilon\tau\left(\sum_{l=i+1}^{d-1} a_l b_{d+i-l}\right) + i : 0 \le i \le d-1\right\} \\ &\le \max\{d\max\{\phi_{Q,\tau}(a_l b_{i-l}), 1+\phi_{Q,\tau}(a_l b_{d+i-l})\} + i\} + cd\lceil\log(d)\rceil \\ &\le \max\{d\max\{\phi_{Q,\tau}(a_l)+\phi_{Q,\tau}(b_{i-l}), 1+\phi_{Q,\tau}(a_l)+\phi_{Q,\tau}(b_{d+i-l})\} + i\} \\ &\quad + cd\lceil\log(d)\rceil + bd \\ &\le \phi_{R,\pi}(x) + \phi_{R,\pi}(y) + b',\end{aligned}$$

which proves the proposition. □

We now have proved that Property 1 holds with $c' = cd$ and $b' = cd\lceil\log(d)\rceil + bd$. Let $e = \max\{\phi_{Q,\tau}(x) : x \in S\}$ and let k satisfy

$$k - c\lceil\log(k)\rceil \ge e + \frac{b'+c'}{d} + 1. \tag{4.3}$$

Let $B' = 2d(k+1)$ and $C' = d(k + c\lceil\log(k)\rceil + e) + d - 1 + b' + c'$. Then it follows from equation (4.3) that $B' > C'$. For any $x = \sum_{i=0}^{d-1} a_i\pi^i \in R$, let $x \in P_0$ if $\phi_{Q,\tau}(a_i) \le k + c\lceil\log(k)\rceil + e$ for every i. Let $P_{R,\pi} = \{u - v : u, v \in P_0\}$. The following two propositions prove Property 2 with respect to $\Phi_{R,\pi}$ and $P_{R,\pi}$.

Proposition 4.5.4 *For any $x, y \in R$, there are $s, t \in P_{R,\pi}$ such that $\pi^{B'} | sx + ty$.*

Proof: For any $u, v \in R$, there is a unique representation:

$$ux + vy \equiv w_0 + w_1\pi + \cdots + w_{d-1}\pi^{B'-1} \pmod{\pi^{B'}},$$

with $w_i \in S$ for $0 \leq i \leq d-1$. If $N = |S|$, then there are $N^{B'} = N^{2d(k+1)}$ distinct representations in this form.

Let $z = \sum_{j=0}^{k} z_j \tau^j$ with $z_j \in S$. Then $\phi_{Q,\tau}(z) \leq k + c\lceil \log(k) \rceil + e$. Also, $\phi_{Q,\tau}(\tau^{k+1}) \leq k + 1 + e \leq k + c\lceil \log(k) \rceil + e$. It follows that there are at least $N^{k+1} + 1$ choices for each coefficient a_i of $x \in P_0$. Thus

$$|P_0| \geq (N^{k+1} + 1)^d,$$

and there are at least $(N^{k+1} + 1)^{2d}$ choices for a pair $u, v \in P_0$. It follows that there are $u, v, u', v' \in P_0$ with $(u, v) \neq (u', v')$, and $ux + vy \equiv u'x + v'y \pmod{\pi^{B'}}$.

Therefore $s = u - u'$ and $t = v - v'$ satisfy the conclusions of the proposition. □

Proposition 4.5.5 *For any $h_1, h_2 \in R$ and any $s, t \in R$, let $h = sh_1 + th_2$. If $s, t \in P_{R,\pi}$, then*

$$\phi_{R,\pi}(h) \leq \max\{\phi_{R,\pi}(h_1), \phi_{R,\pi}(h_2)\} + C'.$$

Proof: Let $w = \max\{\phi_{R,\pi}(h_1), \phi_{R,\pi}(h_2)\}$. By Propositions 4.5.2 and 4.5.3, $\phi_{R,\pi}(sh_1 + th_2) \leq \max\{\phi_{R,\pi}(sh_1), \phi_{R,\pi}(th_2)\} + b'$. Since $s \in P_{R,\pi}$, we have $s = u - v$ for some $u, v \in P_0$. Thus

$$\begin{aligned} \phi_{R,\pi}(s) &\leq \max\{\phi_{R,\pi}(u), \phi_{R,\pi}(v)\} + c' \\ &\leq d(k + c\lceil \log(k) \rceil + e) + d - 1 + c'. \end{aligned}$$

The same bound holds for t. It follows that

$$\begin{aligned} \phi_{R,\pi}(sh_1 + th_2) &\leq d(k + c\lceil \log(k) \rceil + e) + d - 1 + c' + b' + \max\{\phi_{R,\pi}(h_1), \phi_{R,\pi}(h_2)\} \\ &= C' + \max\{\phi_{R,\pi}(h_1), \phi_{R,\pi}(h_2)\}, \end{aligned}$$

which completes the proof. □

It follows that there is a rational approximation algorithm for R, π. Suppose any element x of Q can be represented using at most $p\phi_{Q,\tau}(x)$ bits for some p. Then any element m of R can be represented using at most $p\phi_{R,\pi}(x)$ bits. Thus, by the discussion following Proposition 4.2.1, the number of symbols of the output sequence of an AFSR over R, π needed to synthesize an equivalent AFSR is at most linear in the size of the smallest AFSR that generates the sequence.

While the algorithm is guaranteed to find a rational representation for the given sequence, its Φ value may not be minimal. In fact it may be that multiplying both elements in a pair by the same element (thus leaving the corresponding rational element unchanged) decreases Φ. For example, suppose $\tau = 3$ and $d = 2$ so $\pi^2 = 3$. Let $x = 27 - 14\pi$, $y = 28 - 15\pi$, and $z = 1 + \pi$. Then $\phi_{R,\pi}(x) = \phi_{R,\pi}(y) = 6$. However, $zx = -15 + 13\pi$ and $zy = -17 + 13\pi$ so $\phi_{R,\pi}(zx) = \phi_{R,\pi}(zy) = 5$.

The constants b', c', B', and C' can sometimes be improved upon, giving an improvement in the estimate of the number of iterations sufficient for convergence of the algorithm. If $Q = \mathbf{Z}$ and $\tau > 0$, then we can take $b' = d(3 + f) - 1$ where f is the smallest integer satisfying $d < \tau^{f+1}$. This allows us to take $B' = 2(f + 4)d$ and $C' = B' - 2$. Sometimes we can further improve these constants. For example, if $d = 2$ and $\tau \geq 4$, then in our original version we have $b' = 6$, $c' = 2$, $B' = 30$, and $C' = 29$. The general bounds for $Q = \mathbf{Z}$ give $b' = 5$, $c' = 2$, $B' = 16$, and $C' = 14$. It is possible to improve the last two to $B" = 10$ and $C = 9$ by a different choice of the set P.

4.6 Rational Approximation for Quadratic Extensions

In this section we consider the case of a quadratic extension of a ring Q. Again let Q be a domain, $\tau \in Q$, S a complete set of residues modulo τ with $N = |S|$, and suppose we have an index function $\phi_{Q,\tau}$ and interpolation set $P_{Q,\tau}$ with respect to τ. Let b, c, B, and C be the constants in Properties 1 and 2 with respect to $\phi_{Q,\tau}$ and $P_{Q,\tau}$.

Let $m, g \in Q$ with $m^a = \tau$ for some $a \geq 1$. Let π be a root of the polynomial $X^2 - 2gmX + m^a$, and assume $\pi \notin Q$. In this section we consider whether there

is a rational approximation algorithm for $R = Q[\pi]$. If we let $\Delta = m^a - g^2m^2$, then $\pi = gm + \sqrt{-\Delta}$ and we also have $R = Q[\sqrt{-\Delta}]$. The norm from (the field of fractions of) R to (the field of fractions of) Q is given by $\Gamma(u + v\sqrt{-\Delta}) = u^2 + \Delta v^2$. In particular, $\Gamma(\pi) = \tau$. Let

$$\phi_{R,\pi}(x) = \phi_{Q,\tau}(\Gamma(x)).$$

It follows immediately that

$$\phi_{R,\pi}(xy) \leq \phi_{R,\pi}(x) + \phi_{R,\pi}(y) + b,$$

and

$$\phi_{R,\pi}(\pi^k x) = k + \phi_{R,\pi}(x).$$

However, the additivity condition for an index function does not in general hold. Therefore, we assume at this point that it does hold. That is, we assume that there is a c' such that for any $x_0, x_1, y_0, y_1 \in Q$

$$\phi_{Q,\tau}((x_0 + y_0)^2 + \Delta(x_1 + y_1)^2) \leq \max\{\phi_{Q,\tau}(x_0^2 + \Delta x_1^2), \phi_{Q,\tau}(y_0^2 + \Delta y_1^2) + c'. \quad (4.4)$$

At the end of this section we give examples of rings Q for which this condition holds. For now we show that if it holds, then the remaining conditions – the existence of a set $P_{R,\pi}$ satisfying Property 2 – for the existence of a rational approximation algorithm hold.

First we consider the case when $a \geq 2$. Let $e = \max\{\phi_{Q,\tau}(x) : x \in S\}$ and let $z = 2e + 3b + 3c + c' + \phi_{Q,\tau}(\Delta)$. Choose $r \in \mathbf{Z}$ large enough that $4r \geq 2a^2 - 5a + 2(a-1)z + 4(a-1)b\lceil\log(r)\rceil$. Then we can choose $k \in \mathbf{Z}$ so that

$$\frac{z + 2r + 2b\lceil\log(r)\rceil}{a} \leq k \leq \frac{4r - 2a + 5}{2(a-1)}. \quad (4.5)$$

It follows from equation (4.5) that

$$z + 2r + 2b\lceil\log(r)\rceil < (k+1)a + 1 \quad (4.6)$$

and

$$2a + 2k(a-1) - 1 \leq 4(r+1). \quad (4.7)$$

Let $C' = 2r + 2b\lceil\log(r)\rceil + z$. Let $P_0 = \{s = s_0 + s_1\sqrt{-\Delta} : s_0, s_1 \in Q \text{ and } \phi_{Q,\tau}(s_i) \leq r + b\lceil\log(r)\rceil + e\}$, and $P_{R,\pi} = \{s - s' : s, s' \in P_0\}$. It is immediate that $\phi_{R,\pi}(sh_1 +$

$th_2) \leq \max\{\phi_{R,\pi}(h_1), \phi_{R,\pi}(h_2)\} + C'$ for any $h_1, h_2 \in R$ and $s, t \in P_{R,\pi}$. Also, let $B' = (k+1)a + 1$. Then $B' > C'$ by equation (4.6).

As in the Section 4.5,

$$|\{(s,t) : s, t \in P_0\}| \geq (N^{r+1} + 1)^4.$$

To bound the number of residue classes modulo $\pi^{B'}$ we need a lemma.

Lemma 4.6.1 *For any $k \geq 0$, $\pi^{(k+1)a+1}$ divides $\tau^{a+k(t-1)}$.*

Proof: Let $d, e \in Q$. Then

$$\begin{aligned}(2gm - \pi)(md + \pi e) &= m(m(2gd + m^{a-2}e) - d\pi) \\ &= m(mf - \pi d)\end{aligned}$$

for some $f \in Q$.

We iterate this a times: For any $d, e \in Q$ there are $f, h \in Q$ such that $(2gm - \pi)^a(md + \pi e) = m^a(mf + \pi h) = \pi(2gm - \pi)(mf + \pi h)$. Thus $(2gm - \pi)^{a-1}(md + \pi e) = \pi(mf + \pi h)$. It follows that

$$\begin{aligned}(2gm - \pi)^{a+k(a-1)} &= (2gm - \pi)^{(k+1)(a-1)}(2gm - \pi) \\ &= \pi^{k+1}(mf + \pi h)\end{aligned}$$

for some $f, h \in Q$. Now we have

$$\begin{aligned}\tau^{a+k(a-1)} &= \pm\pi^{a+k(a-1)}(\pi - 2gm)^{a+k(a-1)} \\ &= \pm\pi^{a+k(a-1)}\pi^{k+1}(mf + \pi h) \\ &= \pm\pi^{(k+1)a+1}(mf + \pi h).\end{aligned}$$

This proves the lemma.□

Now let $x + \pi y \in R$, with $x, y \in Q$. We can write $x = x_0 + x_1\tau^{a+1+k(a-1)}$ and $y = y_0 + y_1\tau^{a+k(a-1)}$. It follows from Lemma 4.6.1 that $\pi^{B'} = \pi^{(k+1)a+1}$ divides both $\tau^{a+k(a-1)}$ and $\pi\tau^{a+k(a-1)-1}$. The number of distinct choices modulo $\pi^{B'}$ of x_0 and y_0 is $N^{2a+2k(a-1)-1}$. It follows from equation (4.7) that for any $u, v \in R$ there are $s, t, s', t' \in P_0$ such that $su + tv \equiv s'u + t'v \pmod{\pi^{B'}}$. Therefore $s - s', t - t'$ is a pair in $P_{R,\pi}$ satisfying the requirements of the second part of Property 2.

Now consider the case when $a = 1$. Then $\tau = \pi(2\tau - \pi) = \pi^2(4\tau - 2\pi - 1)$ so π^2 divides τ. In this case we can choose r so that $z + 2b\lceil \log(r) \rceil \leq 2r + 4$, $B' = 4r + 5$, and $C' = z + 2b\lceil \log(r) \rceil$ and a similar argument works. We have proved the following theorem.

Theorem 4.6.2 *If equation (4.4) holds, then there is a rational approximation algorithm for R with respect to π.*

4.6.1 Imaginary Quadratic Extensions of Z

In this subsection we assume $R = \mathbf{Z}[\pi]$ is an imaginary quadratic extension of the integers, with $\pi^2 - 2gm\pi + N = 0$ and $N = m^a$.

In this case Δ is a positive integer. We carry out the above construction with $Q = \mathbf{Z}$, $\tau = N$, and index function and interpolation set as in Section 4.3. It suffices to show equation (4.4) holds. Let $x = x_0 + x_1\sqrt{-\Delta}$ and $y = y_0 + y_1\sqrt{-\Delta}$ with $x_0, x_1, y_0, y_1 \in \mathbf{Z}$. We then have $\Gamma(x + y) = (x_0 + y_0)^2 + \Delta(x_1 + y_1)^2$. Notice that $(c + d)^2 \leq 2(c^2 + d^2)$ for any real numbers c and d. This implies that

$$\begin{aligned} \Gamma(x + y) &\leq 2(x_1^2 + y_1^2) + 2\Delta(x_2^2 + y_2^2) \\ &\leq 2\Gamma(x) + 2\Gamma(y). \end{aligned}$$

Let $w_1 = \phi_{R,\pi}(x) = \phi_{\mathbf{Z},N}(\Gamma(x))$ and $w_2 = \phi_{R,\pi}(y) = \phi_{\mathbf{Z},N}(\Gamma(y))$. Then we have

$$\Gamma(x) \leq N^{w_1+1} - 1,$$

$$\Gamma(y) \leq N^{w_2+1} - 1,$$

and

$$\Gamma(x + y) \leq 4(N^{max(w_1,w_2)+1} - 1).$$

Since $N \geq 2$, we have $\phi_{R,\pi}(x + y) = \phi_{\mathbf{Z},N}(\Gamma(x + y)) \leq \max\{w_1, w_2\} + 2$. We have proven the following corollary.

Corollary 4.6.3 *If $R = \mathbf{Z}[\pi]$ is an imaginary quadratic extension of the integers, with $\pi^2 - 2gm\pi + N = 0$ and $N = m^a$, then R has a rational approximation algorithm with respect to π.*

Any element $m = x_0 + x_1\sqrt{-\Delta}$ can be represented using $\phi_{\mathbf{Z},N}(x_0) + \phi_{\mathbf{Z},N}(x_1) \leq \phi_{\mathbf{Z},N}(x_0^2 + \Delta x_1^2) = \phi_{R,\pi}(m)$ elements of $\{0, 1, \cdots, N-1\}$. Thus, by the discussion following Proposition 4.2.1, the number of symbols of the output sequence of an AFSR over R, π needed to synthesize an equivalent AFSR is at most linear in the size of the smallest AFSR that generates the sequence.

4.6.2 Quadratic Extensions of $\mathbf{Z}[\sqrt{N}]$

In this subsection we let N be a positive integer which is not a perfect square, let $\tau^2 = N$, and let $Q = \mathbf{Z}[\tau]$. Let $\pi^2 - 2gm\pi + \tau = 0$ with $\tau = m^a$ and $g, m \in Q$, and let $R = Q[\pi]$. Thus $Q = \mathbf{Z} + \tau\mathbf{Z}$ and $R = Q + \pi Q$. Let $\Delta = m^a - g^2m^2 = \Delta_0 + \Delta_1\tau$ with $\Delta_0 > 0$, $\Delta_1 \neq 0$ in $\mathbf{Z}$, and $\Delta_0^2 > N\Delta_1^2$. That is, the norm (from the fraction field of Q to the rational numbers) of Δ is positive.

We use the index function and interpolation set defined in Section 4.5, with constants b, c, B, and C for Properties 1 and 2.

Lemma 4.6.4 *If $u \in Q$, then $2\phi_{Q,\tau}(u) - 2 \leq \phi_{Q,\tau}(u^2)$.*

Proof: Let $u = u_0 + u_1\tau$, so $u^2 = u_0^2 + Nu_1^2 + 2u_0u_1\tau$. We have $u_0^2 \leq u_0^2 + Nu_1^2$. Suppose $N^{k-1} < u_0 \leq N^k$ and $N^{l-1} < u_1 \leq N^l$. Then

$$\begin{aligned} 2\phi_{Q,\tau}(u) &= 2\max\{2\phi_{\mathbf{Z},N}(u_0), 2\phi_{\mathbf{Z},N}(u_1) + 1\} \\ &= 2\max\{2k, 2l+1\}. \end{aligned}$$

We also have $u_0^2 + Nu_1^2 > \max\{N^{2(k-1)}, N^{2(l-1)+1}\}$, so

$$\begin{aligned} \phi_{Q,\tau}(u^2) &\geq 2\phi_{\mathbf{Z},N}(u_0^2 + Nu_1^2) \\ &\geq 2\max\{2k-1, 2l\} \\ &= 2\phi_{Q,\tau}(u) - 2, \end{aligned}$$

which proves the lemma. □

Lemma 4.6.5 *Let $\Delta = \Delta_0 + \Delta_1\tau$ with $\Delta_0, \Delta_1 \in \mathbf{Z}$, $\Delta_0 > 0$, $\Delta_1 \neq 0$, and $\Delta_0^2 > N\Delta_1^2$. If $u, v \in Q$, then $2\phi_{Q,\tau}(u) \leq \phi_{Q,\tau}(u^2 + \Delta v^2) + 2$ and $2\phi_{Q,\tau}(v) \leq \phi_{Q,\tau}(u^2 + \Delta v^2) + 2$.*

Proof: Let $u = u_0 + \tau u_1$ and $v = v_0 + \tau v_1$ with $u_0, u_1, v_0, v_1 \in \mathbf{Z}$. Then

$$\begin{aligned} u^2 + \Delta v^2 &= u_0^2 + N u_1^2 + \Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1 \\ &\quad + (2u_0 u_1 + 2\Delta_0 v_0 v_1 + \Delta_1 v_0^2 + \Delta_1 N v_1^2)\tau. \end{aligned} \tag{4.8}$$

We have

$$\begin{aligned} \Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1 &= \Delta_0 (v_0 + \sqrt{N} v_1)^2 + 2 v_0 v_1 \sqrt{N}(\Delta_1 \sqrt{N} - \Delta_0) \quad (4.9) \\ &= \Delta_0 (v_0 - \sqrt{N} v_1)^2 + 2 v_0 v_1 \sqrt{N}(\Delta_1 \sqrt{N} + \Delta_0) \quad (4.10) \end{aligned}$$

Suppose that $\Delta_1\sqrt{N} - \Delta_0$ and $\Delta_1\sqrt{N} + \Delta_0$ have the same sign. Then $\Delta_1^2 N - \Delta_0^2 > 0$, which is false by hypothesis. Thus one is positive and one is negative. Whatever the sign of $v_0 v_1$ is, either expression (4.9) or expression (4.10) is nonnegative. It follows from equation (4.8) that

$$\begin{aligned} \phi_{Q,\tau}(u^2 + \Delta v^2) &\geq 2\phi_{\mathbf{Z},N}(u_0^2 + N u_1^2) \\ &\geq \max\{4\phi_{\mathbf{Z},N}(u_0) - 2, 4\phi_{\mathbf{Z},N}(u_1)\} \\ &= 2\phi_{Q,\tau}(u) - 2. \end{aligned}$$

It also follows that

$$\begin{aligned} \phi_{Q,\tau}(u^2 + \Delta v^2) &\geq 2\phi_{\mathbf{Z},N}(\Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1) \\ &\geq 2\max\{\phi_{\mathbf{Z},N}((v_0 \pm \sqrt{N} v_1)^2), \phi_{\mathbf{Z},N}(2\sqrt{N} v_0 v_1)\}. \end{aligned}$$

Let $m = \phi_{\mathbf{Z},N}(v_0)$ and $l = \phi_{\mathbf{Z},N}(v_1)$. If $l \geq m + 1$, then $\phi_{\mathbf{Z},N}((v_0 \pm \sqrt{N} v_1)^2) \geq N^{2l}$. If $m \geq l \geq m - 1$, then $\phi_{\mathbf{Z},N}(2\sqrt{N} v_0 v_1) \geq \max\{2m, 2l + 1\} - 1$. If $m - 2 \geq l$, then $\phi_{\mathbf{Z},N}((v_0 \pm \sqrt{N} v_1)^2) \geq N^{2m-1}$. In every case it follows that

$$\begin{aligned} \phi_{Q,\tau}(u^2 + \Delta v^2) &\geq 2(\max\{2\phi_{\mathbf{Z},N}(v_0), 2\phi_{\mathbf{Z},N}(v_1) + 1\} - 1) \\ &= 2\phi_{Q,\tau}(v) - 2. \end{aligned}$$

The lemma follows. □

Let $x = x_0 + \pi x_1$ and $y = y_0 + \pi y_1$. We have

$$\begin{aligned} \phi_{R,\pi}(x + y) &= \phi_{Q,\tau}((x_0 + y_0)^2 + \Delta(x_1 + y_1)^2) \\ &\leq \max\{\phi_{Q,\tau}((x_0 + y_0)^2), \phi_{Q,\tau}((x_1 + y_1)^2) + \phi_{Q,\tau}(\Delta) + b\} + c \\ &\leq \max\{2\phi_{Q,\tau}(x_0 + y_0), 2\phi_{Q,\tau}(x_1 + y_1) + \phi_{Q,\tau}(\Delta) + b\} + c + 4 \\ &\leq \max\{2\phi_{Q,\tau}(x_0), 2\phi_{Q,\tau}(y_0), 2\phi_{Q,\tau}(x_1) + \phi_{Q,\tau}(\Delta) + b, \\ &\quad 2\phi_{Q,\tau}(y_1) + \phi_{Q,\tau}(\Delta) + b\} + 3c + 4. \end{aligned}$$

By Lemma 4.6.5, both $2\phi_{Q,\tau}(x_0)$ and $2\phi_{Q,\tau}(x_1)$ are bounded by $\phi_{Q,\tau}(x_0^2 + \Delta x_1^2) + 2$, and similarly for y. It follows that

$$\begin{aligned}\phi_{R,\pi}(x+y) &\leq \max\{\phi_{Q,\tau}(x_0^2+\Delta x_1^2), \phi_{Q,\tau}(y_0^2+\Delta y_1^2)\} + b + 3c + 6 \\ &= \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y)\} + b + 3c + 6.\end{aligned}$$

We have proved the following.

Corollary 4.6.6 *Let N be a positive integer which is not a perfect square, let $\tau^2 = N$, and let $Q = \mathbf{Z}[\tau]$. Let $\pi^2 - 2gm\pi + \tau = 0$ with $\tau = m^a$ and $g, m \in Q$, and let $R = Q[\pi]$. If $m^a - g^2m^2 = \Delta_0 + \Delta_1\tau$ with $\Delta_0 > 0$, $\Delta_1 \neq 0$, and $\Delta_0^2 > N\Delta_1^2$, then R has a rational approximation algorithm with respect to π.*

Any element $m = x_0 + x_1\tau + x_2\sqrt{-\Delta} + x_3\tau\sqrt{-\Delta} \in R$, with $x_i \in \mathbf{Z}$, can be represented using $\sum_{i=0}^{3} \phi_{\mathbf{Z},N}$ elements, plus four sign bits. We have

$$\begin{aligned}\sum_{i=0}^{3} \phi_{\mathbf{Z},N} &\leq 4\max\{\phi_{\mathbf{Z},N}(x_i) : i = 0, \cdots, 3\} \\ &\leq 2\max\{\phi_{Q,\tau}(x_0 + x_1\tau), \phi_{Q,\tau}(x_2 + x_3\tau)\} \\ &\leq \phi_{Q,\tau}((x_0 + x_1\tau)^2 + \Delta(x_2 + x_3\tau)^2) + 2 \\ &= \phi_{R,\pi}(m) + 2.\end{aligned}$$

Thus, by the discussion following Proposition 4.2.1, the number of symbols of the output sequence of an AFSR over R, π needed to synthesize an equivalent AFSR is at most linear in the size of the smallest AFSR that generates the sequence.

Remarks:

(1) We have shown the existence of constants b, c, B, C, but have not attempted to optimize them. We know the algorithm converges after a linear number of iterations. In many cases the convergence may be more rapid than indicated by the results here.

(2) Rational approximation algorithms exist for extensions by roots of other quadratic polynomials. For instance, $\pi = 3 + \sqrt{-3}$ is a root of the equation $X^2 - 6X + 12 = 0$. Let $N = 12$. Then $N^4 = \pi^7(\pi - 6)$. In this case we can choose $b' = 1$. Since this is an imaginary quadratic extension, the additivity condition on the index function holds, in this case with $c' = 1$. We can also take $B' = 7$, and $C' = 6$ to establish a rational approximation algorithm. The task of completely characterizing those quadratic extensions for which there is a rational approximation algorithm remains.

4.7 Comments

For rational π-adic numbers over a domain R, a general rational approximation algorithm has been developed. This algorithm can be used to cryptanalyze sequences that can be generated by an AFSR over $R/(\pi)$. There are several ways to represent any such sequence as a π-adic number: by different choices of the complete set of representatives S, or by different choices of the ring R with given residue ring $R/(\pi)$. For each representation a cryptographic complexity is associated with the sequence. For secure use of such a sequence in stream ciphers, these complexities must be large to guarantee security against the rational approximation algorithms.

Since R is an integral domain, there is a quotient field $Q(R) = \{x/y : x, y \in R, y \neq 0\}$. However, R may not be a *greatest common divisor* (GCD) domain, and so a fraction may not have a reduced form, i.e., $x/y = u/q$ with $gcd(u, q) = 1$. Even if R is a GCD domain, we may not have an efficient way to compute the GCD. If R is a Euclidean domain, it is a GCD domain and a Euclidean algorithm may be used to compute the GCD. It is well known that only finitely many quadratic number fields are fraction fields of Euclidean domains.

We also know that if R is a Dedekind domain, then R being a GCD domain is equivalent to R being a *unique factorization domain* (UFD). Although we do not know if R being a Euclidean domain is equivalent to it being a UFD., there are many rings that are not U.F.D.

The complexities of algebraic structure of the underling ring project difficulties in the analysis of sequences. For instance, in general we can not assume that the generating fraction α has a reduced form. It remains a research project to systematically characterize integral domains over which AFSRs have efficient rational approximation algorithms. Deeper algebraic theory and tools may be needed in such investigations. For cases with rational approximation algorithms, the optimal implementation is another interesting issue.

Bibliography

[1] E. Bach. Efficient prediction of marsaglia-zaman random number generators. *IEEE Trans. Inform. Theory.*, (May), 1998.

[2] H. Beker and F.Piper. *Cipher Systems, The Protection of Communications.* John Wiley and Sons, 1982.

[3] E. R. Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, New York, 1968.

[4] I. F. Blake. Codes over certain rings. *Inform. Control*, 20:396–404, 1972.

[5] I. F. Blake. Codes over integer residue rings. *Inform. Control*, 29:295–300, 1975.

[6] Z. Borevich and I. Shafarevich. *Number Theory.* Academic Press, New York, 1966.

[7] B.Stenström. *Rings and Modules of Quotients*, volume 237 of *Lecture Notes in Mathematics.* Springer-Verlag, New York, 1971.

[8] G. Xiao C. Ding and W. Shan. *The Stability Theory of Steam Ciphers.* Springer-Verlag, 1991.

[9] H. Cohen. *A Course in Computational Algebraic Nnumber Theory.* Springer Verlag, New York, 1993.

[10] B. M. M. de Weger. Approximation lattices of p-adic numbers. *J. Num. Th.*, 24:70–88, 1986.

[11] D. E. Denning. *Cryptology and Data Security.* Addison-Wesley, 1990.

[12] C. Pomerance (editor). *Cryptology and Computational Number Theory*, volume 42 of *Proceedings of Symposia in Applied mathematics.* AMS, Providence, 1990.

[13] D. Gollmann and W.G. Chambers. Clocking controlled shift registers: A review. *IEEE J. on Selected Areas in Communications*, 7(4):525–533, 1989.

[14] S.W. Golomb. *Shift Register Sequences.* Holden-Day, San Francisco, Calif, 1967.

[15] M. Goresky and A. Klapper. Large period nearly debruijin fcsr sequences. *Advances in Cryptology-Eurocrypt 1995, Lecture Notes in Computer Science*, 921:263–273, 1995.

[16] C.G. Günther. Alternating step generators controlled by de bruijn sequences. *Advances in Cryptology-EUROCRYPT'87 Proceedings*, pages 5–14, 1991.

[17] N. Jacobson. *Basic Algebra I.* W.H. Freeman, San Francisco, 1974.

[18] N. Jacobson. *Basic Algebra II.* W.H. Freeman, San Francisco, 1980.

[19] E. L. Key. An analysis of the structure and complexity of nonlinear sequence generators. *IEEE Trans. Info. Theory*, IT-22(6):732–736, 1976.

[20] A. Klapper. Feedback with carry shift registers over finite fields, fast software encryption. *Lecture Notes in Computer Science*, 1008:170–178, 1995.

[21] A. Klapper and M. Goresky. 2-adic shift registers. fast software encryption. *Lecture Notes in Computer Science*, 809:174–178, 1994.

[22] A. Klapper and M. Goresky. Cryptanalysis based on 2-adic rational approximation. *Advances in Cryptology, Crypto '95, Lecture Notes in Computer Science*, 963:262–273, 1995.

[23] A. Klapper and M. Goresky. Feedback shift registers, 2-adic span, and combiners with memory. *Journal of Cryptology*, 10:111–147, 1997.

[24] A. Klapper and J. Xu. Algebraic feedback shift registers, to appear,. *Theoretical Computer Science*, 1998.

[25] D. Knuth. *The Art of Compuer Programming, Vol* 2. *Seminumerical Algorithms.* Addison-Wesley, Reading MA, 1981.

[26] N. Koblitz. *p-Adic Numbers, p-Adic Analysis, and Zeta Functions*, volume 58 of *Graduate Texts in Mathematics.* Springer Verlag, New York, 1984.

[27] N. Koblitz. *A Course in Number Theory and Cryptography*, volume 114 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 1987.

[28] F. J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam,New York, 1993.

[29] K. Mahler. *p-adic numbers and their functions*. 2nd Edition, Cambridge Univ. Press, 1967.

[30] G. Marsaglia and A. Zaman. A new class of random number generators. *Ann. Appl. Prob.*, 1:462–480, 1991.

[31] J. Massey and R. Rueppel. *Methods of, and Apparatus for, Transforming a Digital Data Sequence into an Encoded Form*, volume 4797922 of *U.S. Patent*. 1989.

[32] J. L. Massey. Shift-register synthesis and bch decoding. *IEEE, Trans. Info. Th.*, IT-15(1):122–127, 1969.

[33] W. Qi and Z. Dai. The trace representation of sequences and the structural analysis of the space of nonlinear filtered sequences over $z/(p^d)$. *Acta Mathematicae Applicatae Sinica*, 20(1):128–136, 1997.

[34] J.A. Reeds and N.J.A. Sloane. Shift-register synthesis (modulo m). *SIAM J. Comput.*, 14(3):505–513, August, 1985.

[35] R. Rueppel. *Analysis and Design of Stream Ciphers*. Springer Verlag, New York, 1986.

[36] B. Schneier. *Applied Cryptography*. 2nd Edition, John Wiley and Sons, New York, 1996.

[37] P. Shankar. On bch codes over arbitrary integer rings. *IEEE Trans. Inform. Theory.*, IT-25:480–483, 1977.

[38] W. Sierpinski. *250 Problems in Elementary Number Theory*. American Elsevier Publishing Company Inc., New York, 1970.

[39] D. R. Stinson. *Cryptography: Theory and Practice*. CRC, Boca Raton, London, Tokyo, 1995.

[40] A. Lee Y. Kim, C. Seo and J. Lim. On the linear span of fcsr. *submitted to Elsevier*, 1997.

[41] N. Zierler. Linear recurring sequences. *J. of SIAM*, 7(1):31–48, 1959.

[42] N. Zierler and W.H. Mills. Products of linear recurring sequences. *J. of Algebra*, 27(1):67–69, 1973.

Vita

Jinzhong Xu was born in Shanghai, P. R. China, on December 10, 1958. After finishing his study at Jiangsu Teacher's College, Suzhou, P. R. China (now it is named as Suzhou University), he received the degree of Bachelor of Science in January 1982. Right after the graduation, he was teaching mathematics in high school. In the fall of the same year, he entered Guangxi Normal University in Guilin, P. R. China and began to receive his graduate education. He got his degree of Master of Science from Nanjing University, P. R. China in November 1984. In October 1985, he was employed by Suzhou University, P. R. China, and was teaching at the Department of Mathematics. From 1987 to 1988, he went to University of Toronto, Toronto, Canada as a visiting scholar under an educational exchange program between Jiangsu Province, China and Ontario Province, Canada. From September 1988 to July 1991, he was teaching at Suzhou University as a mathematics lecturer. In August 1991, he entered the Graduate School of University of Kentucky. From the fall of 1991 to the summer of 1994, he was teaching mathematics at the University of Kentucky as a teaching assistant. From the fall of 1994 to the summer of 1995, he received the Dissertation Year Fellowship, awarded by University of Kentucky. He received his Ph.D. degree in mathematics from the College of Arts and Science of University of Kentucky in May 1997. Based on his mathematics researches, his book "Flat Covers of Modules" was published by Springer-Verlag in November 1996. From the Fall of 1995 to the Spring of 1999, he was a research assistant and a teaching assistant at the department of computer science. Since 1999, he has been employed by Assessment Technologies, Inc. as a senior researcher.

Signature of Student