Register Synthesis for Algebraic Feedback Shift Registers Based on Non-Primes

Andrew Klapper and Jinzhong Xu*

Abstract

In this paper, we describe a solution to the register synthesis problem for a class of sequence generators known as *Algebraic Feedback Shift Registers* (or AFSRs). These registers are based on the algebra of π-adic numbers, where π is an element in a ring R, and produce sequences of elements in $R/(\pi)$. 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 these classes. This paper extends the analyses of feedback with carry shift registers and algebraic feedback shift registers by Goresky, Klapper, and Xu [4, 5, 11].

Key words: Feedback shift register, pseudorandom generator, stream cipher, register synthesis, N-adic numbers.

1 Introduction

In the design of stream ciphers, finite state devices for the generation of infinite sequences play two roles. First, they are used in the design of keystream generators. In this capacity, they must be shown to yield sequences that are unpredictable from short prefixes. Second, they are used in cryptanalysis. If it is possible to synthesize an efficient generator of a given sequence from a short prefix, then a cryptanalytic attack can be launched against the given sequence. This is what we call the *register synthesis problem.* For a sequence A and a class $\mathcal{F}$ of sequence generators, we want to find the smallest generator in $\mathcal{F}$ that outputs A. If this can be done by an efficient algorithm whose input

*Dept. of Computer Science,779A Anderson Hall, University of Kentucky, Lexington, KY, 40506-0046. E-mail: klapper@cs.uky.edu, jxu@cs.uky.edu. Project sponsored by the National Science Foundation under grant number NCR-9400762.

is a short prefix of A (where "short" is measured in terms of the size of the smallest generator in $\mathcal{F}$ that outputs A), then we have solved the register synthesis problem for $\mathcal{F}$. It follows that in order for a sequence A to be used in a secure stream cipher, the size of the smallest generator in $\mathcal{F}$ that outputs A must be large.

Various types of registers – such as *linear feedback shift registers* (LFSRs) [3] and *feedback with carry shift registers* (FCSRs) [5] – have been used in such analysis. LFSRs have been studied for several decades. They have been used as generators of statistically random sequences for a variety of applications, including radar, spread spectrum communication, Monte Carlo simulation, and cryptography. From a cryptographic point of view, LFSR sequences are weak because the register synthesis problem is solved by the Berlekamp-Massey algorithm [8]. They are often used, however, as building blocks for generators that are secure against this attack [2, 9, 10].

The analysis of LFSRs is largely based on the interpretation of an infinite sequence as the coefficients of a power series. More recently, using an interpretation of an infinite sequence as a p-adic number with p prime [7], Klapper and Goresky developed FCSRs. The register synthesis problem for FCSRs can be solved as well . These generators have been further generalized to *algebraic feedback shift registers* (AFSRs) [6], based on the I-adic structure of an arbitrary ring R and principal ideal I. These generators are described in detail in the next section. When R/I is finite, these are often finite state devices and hence interesting sequence generators. The elements of the sequences are in R/I. The register synthesis problem for AFSRs, however, has only been solved for certain special cases. For example, in one case for which a solution exists R is a Euclidean domain and I is prime.

In this paper we consider the case when I is not prime. The main result of the paper is a solution to the register synthesis problem in many cases. The case when R is the ring of ordinary integers was considered previously in an extended abstract by the authors [11] using a very different approach from that used by Klapper and Goresky for FCSRs. The current paper gives a general framework for this new approach. It can be applied whenever the pair (R, π) has certain algebraic properties, in particular in many cases when R is a finite extension of the integers.

In Section 2 the definitions and some of the basic properties of AFSRs are reviewed. In Section 3 an algorithm is described that solves the register synthesis problem when the ring R has certain properties. In Sections 5, 6, and 7 we describe several classes of rings where these properties hold.

2 Algebraic Feedback Shift Registers

In this section we recall the construction of algebraic feedback shift registers (AFSRs) and the algebraic basis for their design and analysis [6]. Let R be a commutative ring which is an integral domain (no zero divisors). Let F be its field of fractions. Let $\pi \in R$. In the original paper on AFSRs [6], π was required to be prime, but this restriction is unnecessary for most of the analysis. 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).

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

$$\sum_{i=0}^{\infty} a_i \pi^i, \quad a_i \in S, \tag{1}$$

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 $\mathrm{quo}(\alpha, \pi) \in R$.

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

Definition 2.1 *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.*

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

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

3. *Replace* $(a_0, \cdots, a_{r-1})$ *by* $(a_1, \cdots, a_r)$ *and replace* m *by* $\mathrm{quo}(\tau - q_0 a_r, \pi)$.

Such a device outputs an infinite sequence by repeatedly outputting the last element a_0 and changing states. An AFSR is only a finite state device if the extra memory takes on only finitely many values throughout an infinite execution. This is not the case in general, but conditions can be given under which it is the case.

Proposition 2.2 *[6] 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.*

Hence in the first case the AFSR can be implemented as a finite state device, and the output is eventually periodic. In some cases explicit bounds on the size of memory can be given.

Proposition 2.3 *Suppose* $R = \mathbf{Z}$*. Let* $b = \max\{|a| : a \in S\}$*,* $c = \max\{|a| : a \in T\}$*, and* $d = \sum_{i=0}^{r} |q_i|$*. Let* m_i *be the value of the extra memory in the ith state. Then the following hold.*

1. *If* $|m_i| \leq bd/(|\pi| - 1)$*, then* $|m_{i+1}| \leq bd/(|\pi| - 1)$*.*

2. *If* $|m_i| > bd/(|\pi| - 1)$*, then* $|m_{i+1}| \leq |m_i| - 1$*.*

Therefore, for all i *large enough,* $|m_i| \leq bd/(|\pi| - 1)$*.*

Proof: Suppose that the ith state is $(a_0, a_1, \cdots, a_{r-1})$. From the register construction, we have that
$$\tau = \sum_{i=0}^{r-1} a_i q_{r-1} + m_i = q_0 a_r + \pi m_{i+1}.$$
Let $e = \sum_{i=1}^{r} |q_i|$ and $v = \sum_{i=0}^{r-1} a_i q_{r-1}$. Then $-be \leq v \leq be$.

Suppose $|m_i| \leq bd/(|\pi|-1)$. Then

$$\begin{aligned} |m_{i+1}| &= \frac{|\tau - q_0 a_r|}{|\pi|} \\ &\leq \frac{|v| + |m_i| + |q_0 a_r|}{|\pi|} \\ &\leq \frac{be + bd/(|\pi|-1) + |q_0|b}{|\pi|} \\ &= \frac{bd}{|\pi|-1}. \end{aligned}$$

Now suppose $|m_i| > bd/(|\pi|-1)$. Then

$$\begin{aligned} |m_{i+1}| &\leq \frac{bd + |m_i|}{|\pi|} \\ &< |m_i|. \end{aligned}$$

□

We conclude this section by summarizing some of the properties of AFSRs whose proofs work easily when π is not prime. For an AFSR with taps $q_0, \cdots, q_r$, we call the algebraic integer

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

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

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

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

$$\alpha = \frac{\sum_{n=0}^{r-1}(\sum_{i=0}^{n} q_i a_{n-i})\pi^n - m\pi^r}{q}. \tag{2}$$

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.

3 Rational Approximation

It follows from the preceding section that 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 equation, and it is not stated what 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 decrease 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 [1]. Furthermore, some multiples of q may have smaller π-adic expansions than that of q. An example is given in Section 6. 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 algorithms 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 [8]. Recall that the idea of that 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. Over a ring where there is a carry in addition and multiplication, this is false for reasonable analogues of the notion 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 integer 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 $x, y \in R$. The next proposition follows immediately from the definition.

Proposition 3.1 *For any two pairs* (h_1, r_1), $(h_2, r_2) \in R \times R$ *and integer* $k \geq 0$, *1.* $\Phi_{R,\pi}(h_1+h_2, r_1+r_2) \le \max\{\Phi_{R,\pi}(h_1,r_1), \Phi_{R,\pi}(h_2,r_2)\} + c$*;*

2. $\Phi_{R,\pi}(h_1r_2 - r_1h_2, r_1r_2) \le \Phi_{R,\pi}(h_1,r_1) + \Phi_{R,\pi}(h_2,r_2) + b + c$*;*

3. $\Phi_{R,\pi}(\pi^k(h_1,r_1)) = k + \Phi_{R,\pi}(h_1,r_1)$*.*

Here b *and* c *are the integers appearing in Property 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 by equation (2). Then it follows from Property 1 that

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

and

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

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 subset $P_{R,\pi}$ of R such that the following properties hold.

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

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

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

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

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

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 1. The algorithm maintains a rational element h_i/r_i that is an approximation to α, correct for the first i symbols of the π-adic expansion of α. At each stage we check whether this approximation is correct for the next symbol. If not, we make a correction using an earlier approximation. The new approximation is guaranteed to be correct not only for the new symbol but for B additional symbols.

In the following 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.

Rational_Approximation

> **begin**
> **input** $A = \{a_i \in S, 0 \le i \le k\}$
> $\alpha \longleftarrow 1 + \pi \sum_{i=0}^{k} a_i \pi^i$
> $(h_0, r_0) \longleftarrow (0, 1)$
> $(h_1, r_1) \longleftarrow (1 + a_0\pi + \cdots + a_{B-2}\pi^{B-1}, 1 + \pi^B)$
> $m \longleftarrow 0$
> **for** $(i = m+1$ to $k-1)$
>   **if** $((h_i - r_i\alpha) \not\equiv 0 \; mod(\pi^{i+1}))$ {
>   **if** ($\exists s \neq 0 \in$ with $(\pi^{i+B} \mid s(h_i - r_i\alpha)))$
> $(h_{i+1}, r_{i+1}) \longleftarrow s(h_i, r_i)$
>   **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
> $\Phi(h_i, r_i) \le i - m + \Phi(h_m, r_m)$ and $t \neq 0)$
> $m \longleftarrow i$
>   }
> **output** u, q with $1 + \pi(u/q) = h_k/r_k$
> **end**

Figure 1: Rational Approximation Algorithm.

At the start of the algorithm, we set $\alpha \leftarrow 1 + \pi\alpha$. 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 $\pi^B | s(h_0 - \alpha r_0)$.

4 Proof of Correctness

In this section we show that the algorithm outputs a correct rational representation of α when enough bits are given. We start with some definitions that make the explanation simpler.

Definition 4.1 *We define an index to be a* turning point *as follows:*

1. *The initial index $m = 0$ is a turning point.*
2. *If m_1 is a turning point, then m_2 is the turning point following m_1 if it is the smallest integer greater than m_1 satisfying*
   (a) $(h_{m_2} - \alpha r_{m_2}) \not\equiv 0 \ (mod\ \pi^{m_2+1})$;
   (b) *there is no $s \neq 0$ such that $s \in P$ and $\pi^{m_2+B} | s(h_{m_2} - \alpha r_{m_2})$;*
   (c) $\Phi(h_{m_2+1}, r_{m_2+1}) > \Phi(h_{m_2}, r_{m_2})$;
   (d) $\Phi(h_{m_2}, r_{m_2}) \leq (m_2 - m_1) + \Phi(h_{m_1}, r_{m_1})$.

Conditions 4.1.2.a and 4.1.2.b hold with $m_2 = 0$. An index m is a turning point if it is either zero or it is one where the updating $m \longleftarrow i$ occurs.

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 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 *turn-updating*. That is, i is the turning point next to m.

4.1 The Approximations Are Valid

Recall that each pair (h_i, r_i) in the algorithm corresponds to a fraction $h_i/r_i \in F$. Therefore we must know that r_i is not zero. Otherwise the output from the algorithm would not correspond to an AFSR. We use the following three lemmas to prove this fact. moreover, we use the fact that (h_i, r_i) corresponds to a rational element in many of the following arguments. 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 *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 4.3 *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. At the initial stage $m = 0$, $(h_m, r_m) = (0, 1)$ and $(h_{m+1}, r_{m+1}) = (1 + a_0\pi + \cdots + a_{B-2}\pi^{B-1}, 1 + \pi^B)$. 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 4.4 *At any updating index i we have $\phi(h_i) < i$.*

Proof: The proof is by induction on i. At the initial stage $h_m = 0$. Hence $\phi(h_m) < m$. Now suppose $\phi(k) < k$ 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 $\phi(h_{i+B}) < i + B$. 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 4.5 *For every j, $r_j \neq 0$.*

Proof: The proof is by induction. It is true initially by construction. Since it is true for a type 1 updating, we only need to consider the type 2 updating case. Let $j = i + 1$, 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 4.4, 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 4.3. □

4.2 Convergence

We say the algorithm is *convergent at index i* if $h_i/r_i = \alpha$. In this subsection we determine how many iterations suffice to guarantee convergence. We start with a lower bound on this number in terms of the sizes of the approximations. We then show that the sizes of the approximations grow slowly enough that convergence is guaranteed.

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 4.6 *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 1we have $\Phi(x\pi^i, qr_i) \geq i$. On the other hand, by Property 2 we have $\Phi(x\pi^i, qr_i) = \Phi(h_i q - r_i u, r_i q) \leq \Phi(r, q) + \Phi(h_i, r_i) + b + c$. When $i > \Phi(r, q) + \Phi(h_i, r_i) + b + c$, which is a contradiction. Hence $x = 0$ and $h_i/r_i = u/q = \alpha$. □

Suppose the hypotheses of Lemma 4.6 hold at a non-turning point. Besides updating (h_i, r_i) to (h_{i+1}, r_{i+1}) by using s and t, there are two cases that may cause this. The first is when (h_{i+1}, r_{i+1}) equals (h_i, r_i), so $\Phi(h_i, r_i)$ is unchanged. Lemma 4.6 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 just multiplying (h_i, r_i) by an element $s \in R$. 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 occurs enough times, then the hypotheses of Lemma 4.6 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 4.7 *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 until the next turning point.

Lemma 4.8 *Let m be a turning point, $\delta = \Phi(h_{m+1}, r_{m+1}) - \Phi(h_m, r_m)$. 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 4.7 all the updating indices $i \geq k$ are normal. It follows from Lemma 4.6 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 4.6, 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

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

We set $\beta_0 = 0$. Let k_m be the number of turning points less than m. Let $d = \max\{\phi(a) : a \in S\} \cup \{\phi(1)\}$ here and in what follows.

Lemma 4.9 *At any turning point m*

$$\Phi(h_{m+1}, r_{m+1}) \leq (m + B + c\lceil \log(B) \rceil + d) + Ck_m + \sum_{j \leq m} \beta_j - \Phi(h_m, r_m),$$

and

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

Proof: The proof is by induction. For the base case, $m = 0$, we have $k_0 = 0$, $\beta_0 = 0$, $\Phi(h_1, r_1) = B + c\lceil \log(B) \rceil + d$, and $\Phi(h_0, r_0) = \phi(1) \le d$. 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 $u_i \ge B$ 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

$$\begin{aligned}
\Phi(h_{m_1+1}, r_{m_1+1}) &\le (m_1 - m) + C + \Phi(h_m, r_m) \\
&\le (m_1 - m) + C + (m + B + c\lceil \log(B) \rceil + d) + Ck_m + \sum_{j \le m} \beta_j \\
&\quad -\Phi(h_{m+1}, r_{m+1}) \\
&= (m_1 + B + c\lceil \log(B) \rceil + d) + C(k_m + 1) + \sum_{j \le m} \beta_j + \beta_{m_1} \\
&\quad -\Phi(h_{m_1}, r_{m_1}) \\
&= (m_1 + B + c\lceil \log(B) \rceil + d) + Ck_{m_1} + \sum_{j \le m_1} \beta_j - \Phi(h_{m_1}, r_{m_1}).
\end{aligned}$$

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

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

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

$$\begin{aligned}
Cm_1 &= Cm + C(u_0 + u_1 + \cdots + u_w) \\
&\ge BCk_m + B(\sum_{j \le m} \beta_j) + C(u_0 + u_1 + \cdots + u_w)
\end{aligned}$$

$$\begin{aligned}
&\geq BCk_m + B(\sum_{j\leq m}\beta_j) + BC(w+1)\\
&\geq BCk_m + B(\sum_{j\leq m}\beta_j) + BC + B\beta_{m_1}\\
&= BCk_{m_1} + B(\sum_{j\leq m_1}\beta_j).
\end{aligned}$$

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 $A = a_0, a_1, \cdots$, $a_i \in S$, 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 4.10 *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 the main theorem on the convergence of the algorithm.

Theorem 4.11 *Let i be any index and $\alpha(A) = u/q$ with $\Phi(u,q)$ minimal. Then when*

$$i > \frac{B(2(b+c) + B + c\lceil\log(B)\rceil + d)}{B-C} + 1 + \frac{2B}{B-C}\lambda,$$

where $d = \max\{\phi(a) : a \in S\} \cup \{\phi(1)\}$, *the algorithm is convergent at* i. *That is,*

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

Proof: By Lemma 4.6 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 $w \le t/B$. Then

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

where the second line follows from Lemma 4.9 and the third line follows from Lemma 4.10. It follows that $b+c+\lambda+\Phi(h_i, r_i) < i$ if

$$2(b+c)+B+c\lceil\log(B)\rceil + d + 2\lambda \le \frac{B-C}{B}(i-1).$$

This is equivalent to the hypotheses on i in the statement of the theorem. $\square$

5 Rational Approximation in Z

In this section we consider the case when $R = \mathbf{Z}$, the ordinary integers, treated previously by the authors [11]. In this case π is an integer (possibly composite). Let $S = \{a : 0 \le a \le \pi - 1\}$. If $x \ne 0$ and $|x| = a_0 + a_1\pi + \cdots + a_t\pi^t$ with $a_i \in S$ and $a_t \ne 0$, then we define $\phi_{\mathbf{Z},\pi}(x) = t$. Equivalently, $\phi_{\mathbf{Z},\pi}(x) = t$ if $\pi^t \le |x| < \pi^{t+1}$. Then Property 1 holds with $b = 1$ and $c = 0$. We also define

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

Then Property 2 holds with $B = 3$ and $C = 2$. It follows that the Rational Approximation Algorithm converges in $16 + 6\lambda$ steps.

Suppose m is the initial memory of an AFSR over $\mathbf{Z}$, π. If $\phi_{\mathbf{Z},\pi}(m) \le k$, then m can be represented by $k+1$ elements of S plus one sign bit. Thus, by the discussion following Proposition 3.1, 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.

6 Rational Approximation in 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

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

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

Proposition 6.1 *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 $w = \max\{\phi_{Q,\tau}(a_i) : 0 \le i \le d-1\}$ 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 $1 \le i \le d-1$. Let $w' = \max\{\phi_{Q,\tau}(a'_i) : 0 \le i \le d-1\}$ 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 $\phi_{R,\pi}(x') = dw' + j' = dw + d = (dw + j - 1) + 1 = \phi_{R,\pi}(x) + 1$. $\square$

Proposition 6.2 *For any $x, y \in R$, $\phi_{R,\pi}(x \pm y) \le \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y)\} + cd$.*

Proof: Let
$$x = \sum_{i=0}^{d-1} a_i \pi^i \text{ and } y = \sum_{i=0}^{d-1} b_i \pi^i,$$
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
$$\phi_{Q,\tau}(c_i) \leq \max\{\phi_{Q,\tau}(a_i), \phi_{Q,\tau}(b_i)\} + c,$$
it follows that
$$\begin{aligned} \phi_{R,\pi}(z) &= \max\{d\phi_{Q,\tau}(c_i) + i\} \\ &\leq \max\{d\phi_{Q,\tau}(a_i) + i, d\phi_{Q,\tau}(b_i) + i\} + cd \\ &= \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y)\} + cd. \end{aligned}$$
This proves the proposition. □

Proposition 6.3 *Let $b' = cd\lceil\log(d)\rceil + bd$. Then for any $x, y \in R$, $\phi_{R,\pi}(xy) \leq \phi_{R,\pi}(x) + \phi_{R,\pi}(y) + b'$.*

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

which proves the proposition. $\square$

Thus 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

$$k - c\lceil\log(k)\rceil \geq e + \frac{b' + c'}{d} + 1. \tag{3}$$

Let $B' = 2d(k+1)$ and $C' = d(k + c\lceil\log(k)\rceil + e) + d - 1 + b' + c'$. Then it follows from equation (3) that $B' > C'$. For any $x = \sum_{i=0}^{d-1} a_i\pi^i \in R$, let $x \in P_0$ if $\phi_{Q,\tau}(a_i) \leq k + c\lceil\log(k)\rceil + e$ 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 6.4 *For any $x, y \in R$, there are $s, t \in R$ such that $s, t \in P_{R,\pi}$ and $\pi^{B'}|sx + ty$.*

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. $\square$

Proposition 6.5 *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 6.2 and 6.3,

$$\phi_{R,\pi}(sh_1 + th_2) \leq \max\{\phi_{R,\pi}(sh_1), \phi_{R,\pi}(th_2)\} + b'.$$

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 3.1, 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 $f = \lfloor\log_\tau(d)\rfloor$. That is, 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.

7 Rational Approximation in 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 $x_0, x_1, y_0, y_1 \in Q$

$$\phi_{Q,\tau}((x_0 + y_0)^2 + \Delta(x_1 + y_1)^2) \leq \max\{\phi_{Q,\tau}(x_0^2 + \Delta x_1^2), \phi_{Q,\tau}(y_0^2 + \Delta y_1^2)\} + c'. \tag{4}$$

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

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

(since the gap between the upper and lower bounds is at least one). It follows from equation (5) that

$$z + 2r + 2b\lceil\log(r)\rceil < (k+1)a + 1 \tag{6}$$

and

$$2a + 2k(a-1) - 1 \leq 4(r+1). \tag{7}$$

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 (6).

As in the Section 6,

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

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

Lemma 7.1 *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 7.1 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 the pair x_0, y_0 is $N^{2a+2k(a-1)-1}$. It follows from equation (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 7.2 *If equation (4) holds, then there is a rational approximation algorithm for R with respect to π.*

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.

7.1 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 5. It suffices to show equation (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

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

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 7.3 *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 $\phi_{\mathbf{Z},N}(x_0) + \phi_{\mathbf{Z},N}(x_1) \le \phi_{\mathbf{Z},N}(x_0^2+\Delta x_1^2) = \phi_{R,\pi}(m)$ elements of $\{0, 1, \cdots, N-1\}$. Thus, by the discussion following Proposition 3.1, 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.

7.2 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, we assume 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 6, with constants b, c, B, and C for Properties 1 and 2.

Lemma 7.4 *If $u \in Q$, then $2\phi_{Q,\tau}(u) - 2 \le \phi_{Q,\tau}(u^2)$.*

Proof: Let $u = u_0 + u_1\tau$, so $u^2 = u_0^2 + Nu_1^2 + 2u_0u_1\tau$. We have $u_0^2 \le u_0^2 + Nu_1^2$. Suppose $N^{k-1} < u_0 \le N^k$ and $N^{l-1} < u_1 \le N^l$. 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

$$\begin{aligned} \phi_{Q,\tau}(u^2) &\ge 2\phi_{\mathbf{Z},N}(u_0^2 + Nu_1^2) \\ &\ge 2\max\{2k - 1, 2l\} \\ &= 2\phi_{Q,\tau}(u) - 2, \end{aligned}$$

which proves the lemma. □

Lemma 7.5 *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 $2\phi_{Q,\tau}(u) \le \phi_{Q,\tau}(u^2 + \Delta v^2) + 2$ and $2\phi_{Q,\tau}(v) \le \phi_{Q,\tau}(u^2 + \Delta v^2) + 2$.*

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

$$\begin{aligned} u^2 + \Delta v^2 &= u_0^2 + Nu_1^2 + \Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1 \\ &\quad + (2u_0u_1 + 2\Delta_0 v_0 v_1 + \Delta_1 v_0^2 + \Delta_1 N v_1^2)\tau. \end{aligned} \tag{8}$$

We have

$$\begin{aligned} \Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1 &= \Delta_0(v_0 + \sqrt{N} v_1)^2 + 2v_0 v_1 \sqrt{N}(\Delta_1\sqrt{N} - \Delta_0) &(9) \\ &= \Delta_0(v_0 - \sqrt{N} v_1)^2 + 2v_0 v_1 \sqrt{N}(\Delta_1\sqrt{N} + \Delta_0). &(10) \end{aligned}$$

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 of is positive and one is negative. Whatever the sign of $v_0 v_1$ is, either expression (9) or expression (10) is nonnegative. It follows from equation (8) that

$$\begin{aligned} \phi_{Q,\tau}(u^2 + \Delta v^2) &\geq 2\phi_{\mathbf{Z},N}(u_0^2 + Nu_1^2) \\ &\geq \max\{4\phi_{\mathbf{Z},N}(u_0) - 2, 4\phi_{\mathbf{Z},N}(u_1)\} \\ &= 2\phi_{Q,\tau}(u) - 2. \end{aligned}$$

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

$$\begin{aligned} \phi_{R,\pi}(x + y) &= \phi_{Q,\tau}((x_0 + y_0)^2 + \Delta(x_1 + y_1)^2) \\ &\leq \max\{\phi_{Q,\tau}((x_0 + y_0)^2), \phi_{Q,\tau}((x_1 + y_1)^2) + \phi_{Q,\tau}(\Delta) + b\} + c \\ &\leq \max\{2\phi_{Q,\tau}(x_0 + y_0), 2\phi_{Q,\tau}(x_1 + y_1) + \phi_{Q,\tau}(\Delta) + b\} + c + 4 \\ &\leq \max\{2\phi_{Q,\tau}(x_0), 2\phi_{Q,\tau}(y_0), 2\phi_{Q,\tau}(x_1) + \phi_{Q,\tau}(\Delta) + b, \\ &\qquad 2\phi_{Q,\tau}(y_1) + \phi_{Q,\tau}(\Delta) + b\} + 3c + 4. \end{aligned}$$

By Lemma 7.5, 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 7.6 *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 $tm^a - g^2m^2 = \Delta_0 + \Delta_1\tau$ 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 $\sum_{i=0}^{3} \phi_{\mathbf{Z},N}(x_i)$ elements, plus four sign bits. We have

$$\begin{aligned}
\sum_{i=0}^{3} \phi_{\mathbf{Z},N}(x_i) &\leq 4\max\{\phi_{\mathbf{Z},N}(x_i) : i = 0, \cdots, 3\} \\
&\leq 2\max\{\phi_{Q,\tau}(x_0 + x_1\tau), \phi_{Q,\tau}(x_2 + x_3\tau)\} \\
&\leq \phi_{Q,\tau}((x_0 + x_1\tau)^2 + \Delta(x_2 + x_3\tau)^2) + 2 \\
&= \phi_{R,\pi}(m) + 2.
\end{aligned}$$

Thus, by the discussion following Proposition 3.1, 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.

8 Conclusion

For rational π-adic numbers over a domain R, a general rational approximation algorithm has been developed. This algorithm can be used to cryptanalyze eventually periodic sequences 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.

References

[1] Z. Borevich and I. Shafarevich, *Number Theory.* Academic Press, New York (1966).

[2] D. Gollman and W. Chambers, Clock-controlled shift registers: a review, IEEE Journal on Selected Areas in Communication, vol. 7 (1989) pp. 525-533.

[3] S. Golomb, *Shift Register Sequences*, Aegean Park Press, Laguna Hills, CA, 1982.

[4] A. Klapper and M. Goresky, 2-adic shift registers, Fast Software Encryption, (R. Anderson, ed.) Lecture Notes in Computer Science, vol. 809, Springer-Verlag, Berlin, 809 (1994) pp. 174-178.

[5] A. Klapper and M. Goresky, Feedback shift registers, 2-adic span, and combiners with memory, Journal of Cryptology, vol. 10 (1997) pp. 111-147.

[6] A. Klapper and J. Xu, Algebraic feedback shift registers, Theoretical Computer Science, vol. 226 (1999) pp. 61-93.

[7] N. Koblitz, *p-Adic Numbers, p-Adic Analysis, and Zeta Functions*, Graduate Texts in Mathematics Vol. 58, Springer-Verlag, New York, 1984.

[8] J. Massey, Shift-register synthesis and BCH decoding, IEEE Transactions on Information Theory, vol. IT-15 (1969) pp. 122-127.

[9] J. Massey and R. Rueppel, *Methods of, and Apparatus for, Transforming a Digital Data Sequence into an Encoded Form*, vol. 4797922 of *U.S. Patent*, 1989.

[10] R. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, New York, 1986.

[11] J. Xu and A. Klapper, Feedback with carry shift registers over $\mathbf{Z}/(n)$, *Proceedings of SETA '98*, Springer-Verlag, New York, 1998.

Footnotes

Affiliation of author 1:
Dept. of Computer Science, University of Kentucky, Lexington, KY, 40506-0046.

* Project sponsored by the National Science Foundation under grant number NCR-9400762.

Affiliation of author 2:
Dept. of Computer Science, University of Kentucky, Lexington, KY, 40506-0046.

Key Words

Feedback shift register, pseudorandom generator, stream cipher, register synthesis, N-adic numbers.

Contact Author

Andrew Klapper
Department of Computer Science
779A Anderson Hall
University of Kentucky
Lexington, KY 40506-0046
e-mail: klapper@cs.uky.edu
phone: (617) 257-6743
fax: (617) 323-1971