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*. 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.

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*. More specifically, 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

*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.

whose input 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) [6] and *feedback with carry shift registers* (FCSRs) [11] (described below) – have been used in such analysis.

LFSRs are the most widely studied pseudorandom sequence generators. 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 [15]. They are often used, however, as building blocks for generators that are secure against this attack [5, 16, 18].

The study of pseudorandom sequences most commonly deals with binary sequences, or perhaps with sequences over prime fields $\mathbf{Z}/(p)$. However, there has been a recent surge of interest in sequences over more general modular rings and Galois rings, especially sequences over $\mathbf{Z}/(4)$ [1, 3, 4, 8, 19]. This interest was triggered by the realization that the apparently linearly dual relationship between Kerdock and Preparata codes was explainable by linear codes over $\mathbf{Z}/(4)$ [8]. A variety of subjects concerning sequences over Galois rings have subsequently been studied.

It is natural, therefore, to study the register synthesis problem for sequences over Galois rings. Indeed, just this problem was considered for LFSRs over modular rings $\mathbf{Z}/(n)$ (n not prime) by Reeds and Sloane [17]. They presented a generalization of the Berlekamp-Massey algorithms (although their algorithm is considerably more complex than the Berlekamp-Massey algorithm over a field).

The register synthesis problem for FCSRs (over prime fields) was solved by Klapper and Goresky [11]. In later work, Klapper and Xu defined a generalization of both LFSRs and FCSRs called *algebraic feedback shift registers* (AFSRs) [12], described in detail in Section 2. An AFSR depends in part on a choice of an algebraic ring R and a principal ideal $I = (\pi)$ in R. It produces sequences whose elements can be thought of elements of the quotient ring R/I. LFSRs over a field F are AFSRs with R equal to the polynomial ring $F[x]$ and $\pi = x$. (More generally, F can be an arbitrary ring, usually finite). FCSRs with elements in $\mathbf{Z}/(p)$ are AFSRs with R equal to the ordinary integers and $\pi = p$. Thus the register synthesis problem for AFSRs has been solved when R is a polynomial ring over a field, when R is a polynomial ring over a modular ring $\mathbf{Z}/(n)$, and when R is the ordinary integers and π is prime. But other types of AFSRs are possible. For example, if $R = \mathbf{Z}$ and $\pi = 4$, then AFSRs produces sequences in $\{0, 1, 2, 3\}$, which can be thought of as sequences of pairs of bits. Or if $R = \mathbf{Z}$ and $\pi = 256$, then AFSRs

produces sequences of 32 bit words.

The current paper is concerned with the register synthesis problem for AFSRs over finite extensions of the ordinary integers, with π not necessarily prime. The main result of the paper is a framework that will give rise to an efficient algorithm for solving the register synthesis problem when the pair (R, π) has certain algebraic properties. This includes the case $R = \mathbf{Z}$ and $\pi = 4$ which gives sequences over $\mathbf{Z}/(4)$. The algorithm we present is based on the Berlekamp-Massey algorithm, and is very different from that used by Klapper and Goresky for FCSRs. The case when R is the ring of ordinary integers was considered previously in an extended abstract by the authors [21] and the current paper is an extension of those results.

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 4, 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 [12]. The algebraic notions used here can be found in many texts on modern algebra [9, 10]. Let R be a commutative ring which is an integral domain (no zero divisors). Let F be its field of fractions. Let $\pi \in R$. 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 $s \in S$ that reduces to a modulo π. For simplicity, we may assume that 0 and 1 are always contained in the representative sets. From time to time we may find it convenient to identify S and K. The set of power series

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

forms a ring, $\hat{R}$. Addition and multiplication are defined as for power series, but there may be carry.

For example, we can take $R = \mathbf{Z}$ and $\pi = n$, an integer greater than one, giving rise to the n-adic numbers. We can take $S = \{0, 1, \cdots, n-1\}$ in this case. We add two n-adic numbers $\sum_i a_i n^i$ and $\sum_i b_i n^i$ coefficient by coefficient, but when we add corresponding coefficients the result must be represented in terms of S: $a_i + b_i = (a_i + b_i \bmod n) + \lfloor (a_i + b_i)/n \rfloor n$. As with the ordinary integers, the term $\lfloor (a_i + b_i)/n \rfloor$ is saved as a carry

to the next term. In particular, notice that $-1 = \sum_{i=0}^{\infty}(n-1)n^i$. Indeed, if we add 1 to the n-adic number on the right we get zero due to the infinite carry.

The example $R = K[x]$, with K a field, is also instructive. We let $\pi = x$ (so the quotient field is K as above) and $S = K$. Then $\hat{R}$ is just the ring of ordinary power series (so there is no carry). When this example is used to define AFSRs (see below) we obtain ordinary LFSRs.

In general we assume that $\cap_{i=0}^{\infty} I^i = (0)$ holds[1] Then there is an embedding of R in $\hat{R}$. To see this, solve the infinite system of equations

$$a_i \equiv a - \sum_{j=0}^{i-1} a_j \pi^j, \quad a_i \in S, \quad i = 0, 1, \cdots.$$

The condition on I says that this element of $\hat{R}$ is uniquely defined, and it follows that this defines a ring homomorphism $R \to \hat{R}$.

Furthermore, if $a \in R$ is invertible modulo π (that is, its reduction modulo (π) in K is a unit), then a is invertible in $\hat{R}$. Again, this can be seen by solving an infinite system of equations. In fact if $a = \sum_{i=0}^{\infty} a_i \pi^i$ is any element of $\hat{R}$ with a_i invertible modulo π, then we want to find $b = \sum_{i=0}^{\infty} b_i \pi^i$ so that $ab = 1$. Suppose that we have found $b_0, \cdots, b_{j-1}$ so that $(\sum_{i=0}^{\infty} a_i \pi^i)(b = \sum_{i=0}^{i-1} b_i \pi^i) \equiv 1 \bmod \pi^j$. Then we have only to solve $a_0 b_j +$(terms involving already defined b_is) $\equiv 0$, with $b_j \in S$, to obtain the next coefficient. Thus we can speak of the π-adic expansion of an element $c/a \in F$ with a invertible modulo π. As we shall see, the coefficient sequences of these π-adic expansions are precisely the output sequences from AFSRs.

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

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

[1]This says that R is separable with respect to the I-adic topology, and in this case $\hat{R}$ is the completion of R.

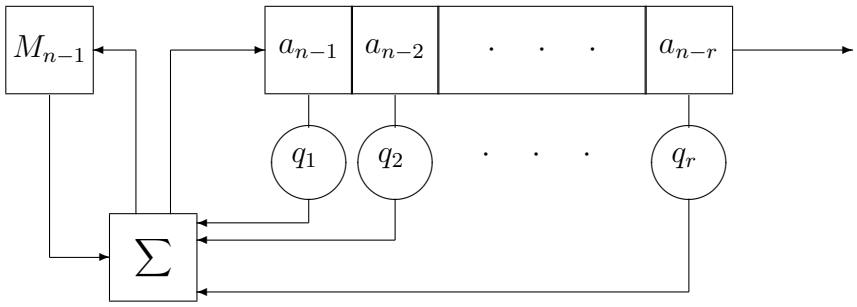

Figure 1: An AFSR Architecture

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

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

A diagram of an AFSR is given in Figure 1. Such a device outputs an infinite sequence by repeatedly outputting the last element a_0 and changing states. It is not immediate that such a device can be implemented in hardware. This is the case if and only if it enters only finitely many distinct states during an infinite execution. This is equivalent to saying that the extra memory takes on only finitely many values throughout an infinite execution. When R is a finite extension of the integers, it is equivalent to the extra memory being bounded. This is not the case in general, but conditions can be given under which it is the case.

Proposition 2.2 *[12] Suppose the fraction field F of R 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.*

In some cases explicit bounds on the size of memory can be given. For example, if $R = \mathbf{Z}$, $b = \max\{|a| : a \in S\}$, and $c = \sum_{i=0}^{r} |q_i|$, then the extra memory needed by a strictly periodic sequence is bounded $|m| \leq bc/(|\pi| - 1)$.

We conclude this section by summarizing some of the properties of AFSRs. For an AFSR with taps $q_0, \cdots, q_r$, we call the element

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

in R the *connection element*. 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 element $q = q_0 + q_1\pi + \cdots + q_r\pi^r$ 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. 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.

3. 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.

In order to measure the computational complexity of algorithms it is desirable to associate a size measure with an AFSR. But this is problematic in general. Roughly speaking the size should be the number of symbols of S needed to store the state. That

is, r plus the size of the additional memory needed. This depends on the representation chosen for elements of R, but in most cases of interest the size of the additional memory for periodic sequences is logarithmic in r. More generally, for a nonperiodic output corresponding to a fraction u/q, the size is approximately the maximum of the sizes of u and q. We use such a measure below in our analysis and show for particular Rs how it relates to the size in elements of S of the associated 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

$$\alpha(A, \pi) = \frac{u}{-q_0 + \sum_{i=1}^{r} q_i \pi^i} = \frac{u}{q}. \tag{3}$$

We say this problem is loosely defined because there are many pairs u, q that satisfy this equation, and it is not stated what condition they should satisfy so that the resulting AFSR is minimal. For example, if v is a unit in R, then vu, vq could be used. Thus even if we have a way to choose u and q relatively prime and satisfying equation (3), we would need too find the unit v so that vu, vq give rise to to the smallest AFSR.

In this section we give a set of conditions on R under which a rational approximation algorithm exists. The conditions include a size measure $\Phi(u, q)$. The algorithm finds a pair u, q satisfying equation (3) given $O(t)$ symbols of A, where t is the minimal $\Phi(u, q)$ for such a pair. We also have $\Phi(u, q) \in O(t)$. First some background.

3.1 Previous Rational Approximation Algorithms

The algorithm we present here is a modification of the Berlekamp-Massey algorithm [15], which solves the register synthesis problem for LFSRs. By identifying sequences over a field F with ordinary power series over F, the register synthesis problem for LFSRs over F reduces to the problem of finding a representation for a power series $f(x) = \sum_i a_i x^i$ as a quotient of polynomials – a *rational function* – given a prefix of its coefficient sequence. The idea of the Berlekamp-Massey algorithm is to maintain at stage j a best rational approximation for f modulo x^j. When a new symbol is processed, if the current best approximation no longer works (i.e., a "discrepancy" occurs), a linear combination of

the current best approximation and a previous one results in a new best approximation. More specifically, $f(x)$ is approximated by $h_j(x)/r_j(x)$ modulo x^j, but not modulo x^{j+1}, if and only if $h_j(x) - f(x)r_j(x) \equiv cx^j$ for some $c \neq 0 \in F$. We let

$$(h_{j+1}, r_{j+1}) = (h_j, r_j) + dx^{j-m}(h_m, r_m),$$

where $d \in F$ is chosen so that $de + c = 0$ if $h_m(x) - f(x)r_m(x) \equiv ex^m$, and m is the most recent iteration when $\max\{\deg(h_m), \deg(r_m)\}$ changed. The proof that this works is quite involved. It was shown in particular that if the given sequence can be generated by a LFSR of length k (or equivalently, $f(x)$ can be written as a quotient of polynomials whose degrees are at most k), then $f = h_j/r_j$ for $j \geq 2k$. The proof of this fact depends on bounds on the degrees of the polynomials that occur. In particular, it uses the facts that (1) the degree of the sum of two polynomials is at most the maximum of the degrees of the two polynomials and (2) the degree of a polynomial multiplied by a constant equals the degree of the polynomial.

There are two difficulties with this approach to the register synthesis problem for more general AFSRs. First, suppose we are considering sequences over $\mathbf{Z}/(4)$. For example, we may want to solve the register synthesis problem for LFSRs over $\mathbf{Z}/(4)$, or for AFSRs with $R = \mathbf{Z}$ and $\pi = 4$. In either case, we may find that the integer e equals 2 while c is 1 or 3. It is then impossible to solve for d. Second, if R is a ring (such as $\mathbf{Z}$) such that there is carry in the addition operation on $\hat{R}$, then the "size" (that is, some reasonable analog of degree) of the elements h_j and r_j may grow too quickly for their quotient to converge to f.

The latter problem was avoided in the case of FCSRs (AFSRs with $R = \mathbf{Z}$) with π prime by using an somewhat different approach [11]. A register synthesis algorithm was designed based on a lattice theoretic approach to π-adic numbers due to Mahler [14] and de Weger [20]. We can think of a pair (h_j, r_j) as above as belonging to a lattice of pairs that approximate f up to the jth coefficient. In the lattice theoretic approach, an optimal basis for this lattice is maintained and updated iteratively. Unfortunately, this approach does not work when π is not prime. Furthermore, it can only be adapted to extensions of $\mathbf{Z}$ that are Euclidean domains, and such rings are quite rare.

3.2 Rational Approximation for AFSRs

Despite the objections of the preceding subsection, in this subsection we describe a modification of the Berlekamp-Massey algorithm that works for many AFSRs over (R, π) such that addition in $\hat{R}$ has carry and π is not prime. This is accomplished with two main modifications. First, the linear combination $(h_j, r_j) + d\pi^{j-m}(h_m, r_m)$ is replaced by a more general linear combination $d_1(h_j, r_j) + d_2\pi^{j-m}(h_m, r_m)$, with d_1, d_2 chosen from

a fixed small set. Second, we control the growth of the approximations by producing a new approximation that works for several new terms at once, thus compensating for the increase in size due to carry when we form these linear combinations.

To make this effective we need two structures: (1) a measure of the size of elements of R that increases in a controlled way when we perform various algebraic operations and (2) a small subset of R from which we can select the coefficients d_1 and d_2. We next describe the properties these structures must have. In later sections we describe various rings that have these structures.

For measuring the size of elements, 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. $\Phi_{R,\pi}(h_1 + h_2, r_1 + r_2) \leq \max\{\Phi_{R,\pi}(h_1, r_1), \Phi_{R,\pi}(h_2, r_2)\} + c$;
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. $\Phi_{R,\pi}(\pi^k(h_1, r_1)) = k + \Phi_{R,\pi}(h_1, r_1)$.

Here b and c are the integers appearing in Property 1.

Suppose some AFSR over R and π has connection element $q = \sum_{i=0}^{r} q_i \pi^r$ 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

$$\phi(q) \leq r + c\,\lceil \log(r+1) \rceil + e$$

and

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

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.

To control the growth of the size of a new approximation which is a combination of previous ones, we restrict the elements that are used to multiply the previous approximations and make the combination. To do so, we assume we have a subset $P_{R,\pi}$ of R such that the following properties hold.

Property 2 There are integers $B > C \geq 0$ 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 $s, t \in P_{R,\pi}$, or $s \in P_{R,\pi}$ and $t = 0$, 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 2.

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 at least B additional symbols.

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 $\pi^B | s(h_0 - \alpha r_0)$.

At the end of the algorithm we have a pair of elements $u, q \in R$ such that, if k is large enough, $u/q = \sum_{i=0}^{\infty} a_i \pi^i$ (this is proved later in the paper). Thus q is the connection

Rational_Approximation

begin
input $A = \{a_i \in S, 0 \leq i \leq k\}$
$\alpha \longleftarrow 1 + \pi \sum_{i=0}^{k} a_i \pi^i$
$(h_0, r_0) \longleftarrow (0, 1)$
$(h_1, r_1) \longleftarrow (1 + a_0\pi + \cdots + a_{B-2}\pi^{B-1}, 1 + \pi^B)$
$m \longleftarrow 0$
for $(i = m + 1$ to $k - 1)$
  if $((h_i - r_i\alpha) \not\equiv 0 \ mod(\pi^{i+1}))$ {
    if ($\exists s \neq 0 \in P$ 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
        $\Phi(h_i, r_i) \leq i - m + \Phi(h_m, r_m)$ and $t \neq 0)$
      $m \longleftarrow i$
  }
Let $1 + \pi(u/q) = h_k/r_k$
Find the largest power t of π that divides both u and q
output $(u/\pi^t, q/\pi^t)$
end

Figure 2: Rational Approximation Algorithm.

element for an AFSR over R that outputs $A = a_0, a_1, \cdots$. As explained above, this might not be the smallest such AFSR. If, however, R is a Euclidean domain (as is the case when $R = \mathbf{Z}$, or when R is among a small finite set of quadratic extensions of $\mathbf{Z}$ [2]), then we can find the greatest common divisor of u and q using the Euclidean algorithm and thus find the smallest such u and q with respect to the Euclidean size function. However we may still not have the smallest AFSR that outputs A – R might have an infinite group of units, so there might be infinitely many connection elements equivalent to q (in the sense that their AFSRs output the same sequences). This does not happen in $\mathbf{Z}$, and we may still be able to find the minimal q in other rings. Furthermore, even if R is not a Euclidean domain, we see below that the size of the AFSR produced is bounded by a constant (depending only on R, π, S, T, and the index function and interpolating set) times the size of the smallest such an AFSR.

4 Rational Approximation in Z

In this section we consider the case when $R = \mathbf{Z}$, the ordinary integers, treated previously by the authors [21]. We give an example of the execution of the algorithm that may help in understanding it. If $R = \mathbf{Z}$, then π is an integer (possibly composite). Let $S = \{a : 0 \le a \le \pi - 1\}$. 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 $\pi^t \le |x| < \pi^{t+1}$. 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 $C = 2$.

Let $\pi = 10$. The sequence $A = \{$ 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 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 $\cdots\}$ is the 10-adic expansion of the fraction $-52/1109$ with period 1108. For simplicity, we skip the shift step $A \to 1 + 10A$. The following steps show how the algorithm is initialized, how approximations are updated, and the simplification at convergence.

Initialization: $m = 0$, $(h_0, r_0) = (0, 1)$, $(h_1, r_1) = (972, 1001)$. The rational number $972/1001$ approximates A to at least the first 3 symbols.

First updating: Since $972/1001$ only approximates A to the first 3 symbols, at index $i = 4$ a new approximation is needed. We have $s = -44$ and $t = -39$. Then we have the new pair $(h_4, r_4) = (-42768, -434044)$, and now $(h_m, r_m) = (972, 1001)$. The rational number $42768/434044$ approximates A to at least the first 6 symbols.

Second updating: Since 42768/434044 only approximates A to the first 6 symbols, at index $i = 7$ a new approximation is needed. We have $s = -50$ and $t = 50$. Then we have the new pair $(h_7, r_7) = (50738400, 71752200)$, and now $(h_m, r_m) = (-42768, -434044)$. The rational number 50738400/71752200 approximates A to at least the first 9 symbols.

Third updating: Since 50738400/71752200 only approximates A to the first 9 symbols, at index $i = 10$ a new approximation is needed. We have $s = -49$ and $t = -42$. Then we have the new pair $(h_{10}, r_{10}) = (-689925600, 14713990200)$, and now $(h_m, r_m) = (50738400, 71752200)$. The rational number $-689925600/14713990200$ approximates A to at least the first 12 symbols.

Convergence: The rational number $u/q = h_{10}/r_{10} = -689925600/14713990200$ gives A exactly.

Reduction: $\gcd(-689925600, 14713990200) = 13267800$. After factoring out the gcd, we have the reduced rational number $u/q = -52/1109$, as desired.

5 Proof of Correctness

In this section we show that the algorithm outputs a correct rational representation of α when enough bits are given. We first show that the output is meaningful. The algorithm computes pairs (h_i, r_i) satisfying $h_i - \alpha r_i \equiv 0 \bmod \pi^i$. We want to interpret (h_i, r_i) as a fraction $h_i/r_i \in F$, and hence as defining an AFSR whose output is $a_0, a_1, \cdots$. But this only makes sense if r_i is not zero. The proof of this fact is essentially the same as in the case $R = \mathbf{Z}$ given previously [21], so it is omitted here.

Theorem 5.1 *For every j, $r_j \neq 0$.*

It remains to prove that if $A = a_0, a_1, \cdots$ can be generated by an AFSR, then after some finite number of steps the algorithm outputs a description of such an AFSR. We say the algorithm is *convergent at index i* if $h_i/r_i = \alpha$.

Definition 5.2 *The minimum value of $\Phi(u, q)$ such that $u/q = \alpha(A)$ is denoted by $\lambda(A) = \lambda$.*

Theorem 5.3 *Let i be any index and $\alpha(A) = u/q$ with $\Phi(u, q)$ minimal. Then when*

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

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

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

The proof of Theorem 5.3 requires a series of lemmas that bound the ϕ values of the various quantities involved. We start with definitions that make the explanation simpler. For any $i \geq 0$, let $\mu(i) = i - \Phi(h_i, r_i)$.

Definition 5.4 *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) $(h_{m_2} - \alpha r_{m_2}) \equiv 0 \ (mod\ \pi^i)(i \leq m_2), \quad \not\equiv 0 \ (mod\ \pi^{m_2+1})$;
   (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) $\mu(m_1) \leq \mu(m_2)$.

Conditions 5.4.2.a and 5.4.2.b hold with $m_2 = 0$. An index m is a turning point if it is either zero or it is one where the assignment $m \longleftarrow i$ 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 (h_i, r_i) an element $s \in R$ or as a linear combination $s(h_i, r_i) + t(h_m, r_m)$. 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 least turning point greater than m.

Next we determine a number of iterations that guarantees 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.

Lemma 5.5 *Suppose* $\alpha = u/q$ *with* $\Phi(u, q)$ *minimal in the set of* $\Phi(h, r)$ *with* $\alpha = h/r$*. If* $\mu(i) > \Phi(u, q) + b + c$*, then* $h_i/r_i = u/q$.

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

Thus if we show that $\Phi(h_i, r_i)$ grows more slowly than i, then we can show the algorithm converges. Let m and m_1 be consecutive turning points. Let

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

and $\beta_0 = 0$. 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 5.6 *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

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

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

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

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

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

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

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

Equivalently, we have the desired result

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

which completes the proof. □

Let λ_m be the smallest $\Phi(h, r)$ with $h - \alpha r = 0 \bmod \pi^m$.

Lemma 5.7 *If m is a turning point, then $\lambda_{m+1} + b + c \ge \mu(m)$.*

Proof: Let $h - \alpha r \equiv 0 \bmod \pi^{m+1}$ and $\lambda_{m+1} = \Phi(h, r)$. Then $(h, r) \ne (h_m, r_m)$. 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 $x \ne 0 \in R$. Therefore $\Phi(hr_m - rh_m, rr_m) \ge m$. On the other hand, we have

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

Consequently, $\lambda_{m+1} + b + c = \Phi(h, r) + b + c \ge \mu(m)$. □

We now can complete the proof of Theorem 5.3.

Proof of Theorem 5.3: By Lemma 5.5 it suffices to show that $\mu(i) > b + c + \lambda$. 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

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

where the second line follows from Lemma 5.6 and the third line follows from Lemma 5.7. It follows that $b + c + \lambda < \mu(i)$ if

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

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

5.1 Complexity

In this subsection we analyze the computational complexity of the algorithm. At each updating index i we have $\Phi(h_{i+1}, r_{i+1}) \leq \max\{\Phi(h_i, r_i), \Phi(h_m, r_m)\} + C$ by Property 2, where m is the most recent turning point. Furthermore, at most one out of every B consecutive indices can be an updating index. If i is a non-updating index, then $\Phi(h_{i+1}, r_{i+1}) = \Phi(h_i, r_i)$. Therefore $\Phi(h_i, r_i) \leq C\lceil i/B \rceil$, so $\Phi(h_i, r_i) \leq i$ if $i \geq B - 1$. Also, note that we do not have to save all the intermediate values (h_i, r_i), just the current value and the value for the most recent turning point.

Suppose we have a bound $\sigma(m)$ on the time required to add two elements $a, b \in R$ with $\phi(a), \phi(b) \leq m$. Then we have the following.

Corollary 5.8 *The Rational Approximation Algorithm has worst case time complexity in* $O(\sum_{m=1}^{\lambda} \sigma(m))$. *The space required is* $O(\lambda \log(|S|))$.

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

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

This is well defined because, by the irreducibility of $X^d - \epsilon\tau$, this representation of x is unique. Let $c' = cd$ and $b' = cd\lceil \log(d) \rceil + bd$. Then for any $x, y \in R$ and non-zero integer k,

1. $\phi_{R,\pi}(\pi^k x) = k + \phi_{R,\pi}(x)$;
2. $\phi_{R,\pi}(x \pm y) \leq \max\{\phi_{R,\pi}(x), \phi_{R,\pi}(y)\} + cd$;
3. $\phi_{R,\pi}(xy) \leq \phi_{R,\pi}(x) + \phi_{R,\pi}(y) + b'$.

Let $e = \max\{\phi_{Q,\tau}(x) : x \in S\}$ and let k satisfy

$$k - c\lceil \log(k) \rceil \geq e + \frac{b' + c'}{d} + 1. \tag{4}$$

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) 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 $P_{R,\pi} = \{u - v : u, v \in P_0\}$. Then Properties 1 and 2 hold with constants b', c', B', C'.

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 $x_0, x_1, y_0, y_1 \in Q$

$$\phi_{Q,\tau}((x_0+y_0)^2+\Delta(x_1+y_1)^2) \leq \max\{\phi_{Q,\tau}(x_0^2+\Delta x_1^2), \phi_{Q,\tau}(y_0^2+\ \Delta y_1^2)\}+c'. \qquad (5)$$

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

$$\frac{z+2r+2b\lceil\log(r)\rceil-a}{a} \leq k \leq \frac{4r-2a+5}{2(a-1)} \qquad (6)$$

(since the gap between the upper and lower bounds is at least one). It follows from equation (6) that

$$z+2r+2b\lceil\log(r)\rceil < (k+1)a+1 \qquad (7)$$

and

$$2a+2k(a-1)-1 \leq 4(r+1). \qquad (8)$$

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 (7).

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

$$(2gm-\pi)^{a+k(a-1)} = (2gm-\pi)^{(k+1)(a-1)}(2gm-\pi) = \pi^{k+1}(mf+\pi h),$$

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 (8) that for any $u, v \in R$ there are $s, t, s', t' \in P_0$ such that $su + tv \equiv s'u + t'v \bmod \pi^{B'}$. 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 (5) 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 4. It suffices to show

equation (5) 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 $\phi_{\mathbf{Z},N}(x_0) + \phi_{\mathbf{Z},N}(x_1) \leq \phi_{\mathbf{Z},N}(x_0^2+\Delta x_1^2) = \phi_{R,\pi}(m)$ elements of $\{0, 1, \cdots, N-1\}$. Thus, by the discussion following Proposition 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 $2\phi_{Q,\tau}(u) - 2 \leq \phi_{Q,\tau}(u^2)$.*

Proof: Straightforward. □

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 $2\phi_{Q,\tau}(u) \leq \phi_{Q,\tau}(u^2 + \Delta v^2) + 2$ 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

$$\begin{aligned} u^2 + \Delta v^2 &= u_0^2 + Nu_1^2 + \Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1 \\ &\quad +(2u_0u_1 + 2\Delta_0 v_0 v_1 + \Delta_1 v_0^2 + \Delta_1 N v_1^2)\tau. \end{aligned} \tag{9}$$

We have

$$\begin{aligned} \Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1 &= \Delta_0(v_0 + \sqrt{N}v_1)^2 + 2v_0v_1\sqrt{N}(\Delta_1\sqrt{N} - \Delta_0) \quad (10) \\ &= \Delta_0(v_0 - \sqrt{N}v_1)^2 + 2v_0v_1\sqrt{N}(\Delta_1\sqrt{N} + \Delta_0). \quad (11) \end{aligned}$$

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_0v_1 is, either expression (10) or expression (11) is nonnegative. It follows from equation (9) 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

$$\begin{aligned} \phi_{Q,\tau}(u^2 + \Delta v^2) &\geq 2\phi_{\mathbf{Z},N}(\Delta_0 v_0^2 + \Delta_0 N v_1^2 + 2\Delta_1 N v_0 v_1) \\ &\geq 2\max\{\phi_{\mathbf{Z},N}((v_0 \pm \sqrt{N}v_1)^2), \phi_{\mathbf{Z},N}(2\sqrt{N}v_0v_1)\}. \end{aligned}$$

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 $\phi_{\mathbf{Z},N}(2\sqrt{N}v_0v_1) \geq \max\{2m, 2l + 1\} - 1$. 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 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] A. Bonnecaze, P. Sole, C. Bachoc and B. Mourrain, Type II codes over $\mathbf{Z}_4$, IEEE Trans. Info. Theory, vol. IT-43 (1997) pp. 969-976.

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

[3] J. Conway and N. J. Sloane, Self-dual codes over the integers modulo 4, J. Combin. Theory, Ser. A, vol. 62 (1993) pp. 30-45.

[4] J. Fields and P. Gaborit, On the non $\mathbf{Z}_4$-linearity of certain good binary codes, IEEE Trans. Info. Theory, vol. IT-45 (1999) pp. 1674-1677.

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

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

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

[8] N. A. Hammons, P. Kumar, A. Calderbank, N. Sloane, and P. Sole, $\mathbf{Z}_4$ Linearity of Kerdock, Preparata, Goethals, and related codes, IEEE Trans. Infor. Theory, vol. 40 (1994) pp. 301-319.

[9] N. Jacobson, *Basic Algebra I*, W.H. Freeman, San Francisco (1974).

[10] N. Jacobson, *Basic Algebra II*, W.H. Freeman, San Francisco (1980).

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

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

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

[14] K. Mahler, On a geometrical representation of p-adic numbers, Ann. of Math., vol. 41 (1940) pp. 8-56.

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

[16] 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.

[17] J. Reeds and N. Sloane, Shift-register synthesis (modulo m), SIAM J. Comp., vol. 14 (1985) pp. 505-513.

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

[19] A. Shanbhag, P. Kumar and T. Helleseth, Improved binary codes and sequence families from $\mathbf{Z}_4$ linear codes, IEEE Trans. Info. Theory, vol. IT-42 (1996) pp. 1582-1587.

[20] B. M. M. de Weger, Approximation lattices of p-adic numbers, J. Num. Thy., vol. 24 (1986) pp. 70-88.

[21] 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.

Page 4: 1. This says that R is separable with respect to the I-adic topology, and in this case $\hat{R}$ is the completion of R.

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: (859) 257-6743
fax: (859) 323-1971