COMPUTING IGUSA’S LOCAL ZETA FUNCTIONS OF UNIVARIATE POLYNOMIALS, AND LINEAR FEEDBACK SHIFT REGISTERS

W. A. ZUNIGA-GALINDO

Abstract. We give a polynomial time algorithm for computing the Igusa local zeta function $Z(s,f)$ attached to a polynomial $f(x) \in \mathbb{Z}[x]$, in one variable, with splitting field $\mathbb{Q}$, and a prime number $p$. We also propose a new class of Linear Feedback Shift Registers based on the computation of Igusa’s local zeta function.

1. Introduction

Let $f(x) \in \mathbb{Z}[x]$, $x = (x_1, \cdots, x_n)$ be a non-constant polynomial, and $p$ a fixed prime number. We put $N_m(f,p) = N_m(f)$ for the number of solutions of the congruence $f(x) \equiv 0 \mod p^m$ in $(\mathbb{Z}/p^m\mathbb{Z})^n$, $m \geq 1$, and $H(t,f)$ for the Poincaré series

$$H(t,f) = \sum_{m=0}^{\infty} N_m(f)(p^{-m}t)^m,$$

with $t \in \mathbb{C}$, $|t| < 1$, and $N_0(f) = 1$. This paper is dedicated to the computation of the sequence $\{N_m(f)\}_{m \geq 0}$ when $f$ is an univariate polynomial with splitting field $\mathbb{Q}$.

Igusa showed that the Poincaré series $H(t,f)$ admits a meromorphic continuation to the complex plane as a rational function of $t$. In this paper we make a first step towards the solution of the following problem: given a polynomial $f(x)$ as above, how difficult is to compute the meromorphic continuation of the Poincaré series $H(t,f)$?

The computation of the Poincaré series $H(t,f)$ is equivalent to the computation of Igusa’s local zeta function $Z(s,f)$, attached to $f$ and $p$, defined as follows. We denote by $\mathbb{Q}_p$ the field of $p$–adic numbers, and by $\mathbb{Z}_p$ the ring of $p$–adic integers. For $x \in \mathbb{Q}_p$, $v_p(x)$ denotes the $p$–adic order of $x$, and $|x|_p = p^{-v_p(x)}$ its absolute value. The Igusa local zeta function associated to $f$ and $p$ is defined as follows:

$$Z(s,f) = \int_{\mathbb{Z}_p^n} |f(x)|_p^s \, dx, \quad s \in \mathbb{C},$$

where Re$(s) > 0$, and $\, dx \,$ denotes the Haar measure on $\mathbb{Q}_p^n$ so normalized that $\mathbb{Z}_p^n$ has measure 1. The following relation between $Z(s,f)$ and $H(t,f)$ holds (see

1991 Mathematics Subject Classification. Primary 11S40, 94A60; Secondary 11Y16, 14GG50.

Key words and phrases. Igusa’s local zeta function, polynomial time algorithms, one-way functions, linear feedback shift registers.

Supported by COLCIENCIAS-Grant # 089-2000.
Thus, the rationality of $Z(s, f)$ implies the rationality of the Poincaré series $H(t, f)$, and the computation of $H(t, f)$ is equivalent to the computation of $Z(s, f)$. Igusa [14, theorem 8.2.1] showed that the local zeta function $Z(s, f)$ admits a meromorphic continuation to the complex plane as a rational function of $p^{-s}$.

The first result of this paper is a polynomial time algorithm for computing the local zeta function $Z(s, f)$ attached to a polynomial $f(x) \in \mathbb{Z}[x]$, in one variable, with splitting field $\mathbb{Q}$, and a prime number $p$. We also give an explicit estimate for its complexity (see algorithm Compute $Z(s, f)$ in section 2, and theorem 7.1).

Many authors have found explicit formulas for $Z(s, f)$, or $H(f, t)$, for several classes of polynomials, among them [6], [7], [10], [11], [16] and the references therein], [19], [24], [25]. In all these works the computation of $Z(s, f)$, or $H(f, t)$, is reduced to the computation of other problems, as the computation of the number of solutions of polynomial equations with coefficients in a finite field. Currently, there is no polynomial time algorithm solving this problem [23], [22]. Moreover, none of the above mentioned works include complexity estimates for the computation of Igusa’s local zeta functions.

Of particular importance is Denef’s explicit formula for $Z(s, f)$, when $f$ satisfies some generic conditions [6]. This formula involves the numerical data associated to a resolution of singularities of the divisor $f = 0$, and the number of rational points of certain non-singular varieties over finite fields. Thus the computation of $Z(s, f)$, for a generic polynomial $f$, is reduced to the computation of the numerical data associated to a resolution of singularities of the divisor $f = 0$, and the number of solutions of non-singular polynomials over finite fields. Currently, it is unknown if these problems can be solved in polynomial time on a Turing machine. However, during the last few years important achievements have been obtained in the computation of resolution of singularities of polynomials [2], [3], [4], [21].

The computation of the Igusa local zeta function for an arbitrary polynomial seems to be an intractable problem on a Turing machine. For example, for $p = 2$, the computation of the number of solutions of a polynomial equation with coefficients in $\mathbb{Z}/2\mathbb{Z}$ is an NP-complete problem on a Turing Machine [9] page 251, problem AN9]. Then in the case of $2$–adic numbers, the computation of the Igusa local zeta function is an NP-complete problem.

Recently, Anshel and Goldfeld have shown the existence of a strong connection between the computation of zeta functions and cryptography [11]. Indeed, they proposed a new class of candidates for one-way functions based on global zeta functions. A one-way function is a function $F$ such that for each $x$ in the domain of $F$, it is easy to compute $F(x)$; but for essentially all $y$ in the range of $F$, it is an intractable problem to find an $x$ such that $y = F(x)$. These functions play a central role, from a practical and theoretical point of view, in modern cryptography. Currently, there is no guarantee that one-way functions exist even if $\textbf{P} \neq \textbf{NP}$. Most of the present candidates for one-way functions are constructed on the intractability of problems like integer factorization and discrete logarithms [12]. Recently, P. Shor has introduced a new approach to attack these problems [20]. Indeed, Shor have shown that on a quantum computer the integer factorization and discrete logarithm problems can be computed in polynomial time.
We set
\[ H = \{ H(t, f) \mid f(x) \in \mathbb{Z}[x], \text{ in one variable, with splitting field } \mathbb{Q} \}, \]
and \( N_\infty(\mathbb{Z}) \) for the set of finite sequences of integers. For each positive integer \( u \) and a prime number \( p \), we define
\[ F_{u,p} : H \rightarrow N_\infty(\mathbb{Z}) \]
\[ H(t, f) \rightarrow \{ N_0(f, p), N_1(f, p), \ldots, N_u(f, p) \}. \]

Our second result asserts that \( F_{u,p}(H(t, f)) \) can be computed in polynomial time, for every \( H(t, f) \) in \( H \) (see theorem 8.1). It seems interesting to study the complexity on a Turing machine of the following problem: given a list of positive integers \( \{ a_0, a_1, \ldots, a_u \} \), how difficult is it to determine whether or not there exists a Poincaré series \( H(t, f) = \sum_{m=0}^{\infty} N_m(f)(p^{-1}t)^m \), such that \( a_i = N_i(f), i = 1, \ldots, u \)?

Currently, the author does not have any result about the complexity of the above problem, however the mappings \( F_{u,p} \) can be considered as new class of stream ciphers (see section 8).

2. The Algorithm \texttt{Compute}_Z(s, f)

In this section we present a polynomial time algorithm, \texttt{Compute}_Z(s, f), that solves the following problem: given a polynomial \( f(x) \in \mathbb{Z}[x] \), in one variable, whose splitting field is \( \mathbb{Q} \), find an explicit expression for the meromorphic continuation of \( Z(s, f) \). The algorithm is as follows.

\begin{itemize}
  \item \textbf{Algorithm Compute}_Z(s, f)
  \item Input : A polynomial \( f(x) \in \mathbb{Z}[x] \), in one variable, whose splitting field is \( \mathbb{Q} \).
  \item Output : A rational function of \( p^{-s} \) that is the meromorphic continuation of \( Z(s, f) \).
  
  (1) Factorize \( f(x) \) in \( \mathbb{Q}[x] \): \( f(x) = a_0 \prod_{i=1}^{r} (x - \alpha_i)^{e_i} \in \mathbb{Q}[x] \).
  
  (2) Compute
  \[ l_f = \begin{cases} 
    1 + \max\{ v_p(\alpha_i - \alpha_j) \mid i \neq j, 1 \leq i, j \leq r \}, & \text{if } r \geq 2; \\
    1, & \text{if } r = 1. 
  \end{cases} \]

  (3) Compute the \( p \)-adic expansions of the numbers \( \alpha_i, i = 1, 2, \ldots, r \) modulo \( p^{l_f+1} \).
  
  (4) Compute the tree \( T(f, l_f) \) associated to \( f(x) \) and \( p \) (for the definition of \( T(f, l_f) \) see \textbf{1.2}).
  
  (5) Compute the generating function \( G(s, T(f, l_f), p) \) attached to \( T(f, l_f) \) (for the definition of \( G(s, T(f, l_f), p) \) see \textbf{5.1}).
  
  (6) Return \( Z(s, f) = G(s, T(f, l_f), p) \).
  
  (7) End
\end{itemize}

In section 6, we shall give a proof of the correctness and a complexity estimate for the algorithm \texttt{Compute}_Z(s, f). The first step in our algorithm is accomplished by means of the factoring algorithm by A.K. Lenstra, H. Lenstra and L. Lovász \textbf{17}. If \( d_f \) denotes the degree of \( f(x) = \sum_i a_i x^i \), and
\[ \| f \| = \sqrt{\sum_i a_i^2}, \]
then the mentioned factoring algorithm needs \( O \left( d^5 + d^3 \log |f| \right) \) arithmetic operations, and the integers on which these operations are performed each have a binary length \( O \left( d^3 + d^3 \log |f| \right) \) [17, theorem 3.6].

The steps 2, 3, 4, 5 reduce in polynomial time the computation of \( Z(s, f) \) to the computation of a factorization of \( f(x) \) over \( \mathbb{Q} \). This reduction is accomplished by constructing a weighted tree from the \( p \)-adic expansion of the roots of \( f(x) \) modulo a certain power of \( p \) (see section 4), and then associating a generating function to this tree (see section 5). Finally, we shall prove that the generating function constructed in this way coincides with the local zeta function of \( f(x) \) (see section 5).

3. \( p \)-adic Stationary Phase Formula

Our main tool in the effective computing of Igusa’s local zeta function of a polynomial in one variable will be the \( p \)-adic stationary phase formula, abbreviated SPF [16]. This formula is a recursive procedure for computing local zeta functions. By using this procedure it is possible to compute the local zeta functions for many classes of polynomials [16] and the references therein, [19], [21], [25], [20].

Given a polynomial \( f(x) \in \mathbb{Z}_p[x] \setminus p\mathbb{Z}_p[x] \), we denote by \( \overline{f}(x) \) its reduction modulo \( p\mathbb{Z}_p \), i.e., the polynomial obtained by reducing the coefficients of \( f(x) \) modulo \( p\mathbb{Z}_p \). We define for each \( x_0 \in \mathbb{Z}_p \),

\[
f_{x_0}(x) = p^{-e_{x_0}} f(x_0 + px),
\]

where \( e_{x_0} \) is the minimum order of \( p \) in the coefficients of \( f(x_0 + px) \). Thus \( f_{x_0}(x) \in \mathbb{Z}_p[x] \setminus p\mathbb{Z}_p[x] \). We shall call the polynomial \( f_{x_0}(x) \) the dilatation of \( f(x) \) at \( x_0 \). We also define

\[
\nu(\overline{f}) = \text{Card}\{ \overline{x} \in \mathbb{F}_p \mid \overline{f}(\overline{x}) \neq 0 \},
\]

\[
\delta(\overline{f}) = \text{Card}\{ \overline{x} \in \mathbb{F}_p \mid \overline{x} \text{ is a simple root of } \overline{f}(\overline{x}) = 0 \}.
\]

We shall use \( \{0, 1, \ldots, p-1\} \subseteq \mathbb{Z}_p \) as a set of representatives of the elements of \( \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\} \). Let \( S = S(f) \) denote the subset of \( \{0, 1, \ldots, p-1\} \subseteq \mathbb{Z}_p \) which is mapped bijectively by the canonical homomorphism \( \mathbb{Z}_p \rightarrow \mathbb{Z}_p/p\mathbb{Z}_p \) to the set of roots of \( \overline{f}(\overline{x}) = 0 \) with multiplicity greater than or equal to two.

With all the above notation we are able to state the \( p \)-adic stationary phase formula for polynomials in one variable.

**Proposition 3.1** ([14, theorem 10.2.1]). Let \( f(x) \in \mathbb{Z}_p[x] \setminus p\mathbb{Z}_p[x] \) be a non-constant polynomial. Then

\[
Z(s, f) = p^{-1} \nu(\overline{f}) + \delta(\overline{f}) \frac{(1-p^{-1})p^{-1-s}}{(1-p^{-1-s})} + \sum_{\xi \in S} p^{-1-\epsilon_0} \int_{\mathbb{Z}_p} |f_\xi(x)|^s_p \, dx.
\]

The following example illustrates the use of the \( p \)-adic stationary phase formula, and also the basic aspects of our algorithm for computing \( Z(s, f) \).

3.1. **Example.** Let \( f(x) = (x - \alpha_1)(x - \alpha_2)^3(x - \alpha_3)(x - \alpha_4)^2(x - \alpha_5) \) be a polynomial such that \( \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5 \) are integers having the following \( p \)-adic expansions:

\[
\alpha_1 = a dp + kp^2,
\]

\[
\alpha_2 = a dp + lp^2,
\]

Then

\[
\alpha_3 = a dp + mp^2,
\]

\[
\alpha_4 = a dp + np^2,
\]

\[
\alpha_5 = a dp + qp^2.
\]
\[ \alpha_3 = b + gp + mp^2, \]
\[ \alpha_4 = c + hp + np^2, \]
\[ \alpha_5 = c + hp + rp^2, \]
where the p-adic digits \( a, b, c, d, g, h, l, m, n, r \) belong to \( \{0, 1, \ldots, p-1\} \). We assume the p-adic digits to be different by pairs. The local zeta function \( Z(s, f) \) will be computed by using SPF iteratively.

By applying SPF with \( f(x) = (x-\overline{a})^4(x-\overline{b})(x-\overline{c})^3, \nu(\overline{f}) = p-3, \delta(\overline{f}) = 1, S = \{a, c\}, f_a(x) = p^{-1}f(a+px), \) and \( f_c(x) = p^{-3}f(c+px) \), we obtain that

\[
Z(s, f) = p^{-1}(p-3) + \frac{(1-p^{-1})p^{-1-s}}{1-p^{-1-s}} + p^{-1}(p-1)p^{-1-4s} + \int_{\mathbb{Z}_p} |f_a(x)|_p^s \, dx \]
\[
+ p^{-1-3s} \int_{\mathbb{Z}_p} |f_c(x)|_p^s \, dx \, . \tag{3.1}
\]

We apply SPF to the integrals involving \( f_a(x) \) and \( f_c(x) \) in (3.1). First, we consider the integral corresponding to \( f_a(x) \). Since \( f_a(x) = (x-\overline{a})^4(x-\overline{b})(x-\overline{c})^3, S = \{d\}, f_{a,d}(x) = p^{-4}f_a(d+px), \nu(\overline{f_a}) = p-1, \) and \( \delta(\overline{f_a}) = 0 \), it follows from (3.1) using SPF that

\[
Z(s, f) = p^{-1}(p-3) + \frac{(1-p^{-1})p^{-1-s}}{1-p^{-1-s}} + p^{-1}(p-1)p^{-1-4s} + \int_{\mathbb{Z}_p} |f_{a,d}(x)|_p^s \, dx \]
\[
+ p^{-2-8s} \int_{\mathbb{Z}_p} |f_{a,d}(x)|_p^s \, dx \, . \tag{3.2}
\]

Now, we apply SPF to the integral involving \( f_c(x) \) in (3.2). Since \( f_c(x) = (x-\overline{c})^4(x-\overline{a})(x-\overline{c})^3, S = \{h\}, f_{c,h}(x) = p^{-3}f_c(h+px), \nu(\overline{f_c}) = p-1, \) and \( \delta(\overline{f_c}) = 0 \), it follows from (3.2) using SPF that

\[
Z(s, f) = p^{-1}(p-3) + \frac{(1-p^{-1})p^{-1-s}}{1-p^{-1-s}} + p^{-1}(p-1)p^{-1-4s} + \int_{\mathbb{Z}_p} |f_{a,d}(x)|_p^s \, dx \]
\[
+ p^{-2-8s} \int_{\mathbb{Z}_p} |f_{a,d}(x)|_p^s \, dx \, . \tag{3.3}
\]

By applying SPF to the integral involving \( f_{a,d}(x) \) in (3.3), with \( f_{a,d}(x) = (x-\overline{k})(x-\overline{l})^3(x-\overline{c})(x-\overline{a})^3, S = \{k, l\}, f_{a,d,k}(x) = p^{-1}f_{a,d}(k+px), |f_{a,d,k}(x)|_p^s = |x|_p^s, f_{a,d,l}(x) = p^{-3}f_{a,d}(l+px), |f_{a,d,l}(x)|_p^s = |x|_p^s, \nu(\overline{f_{a,d}}) = p-2, \) and \( \delta(\overline{f_{a,d}}) = 1 \), we obtain that

\[
Z(s, f) = p^{-1}(p-3) + \frac{(1-p^{-1})p^{-1-s}}{1-p^{-1-s}} + p^{-1}(p-1)p^{-1-4s} + \int_{\mathbb{Z}_p} |f_{a,d}(x)|_p^s \, dx \]
\[
+ p^{-1}(p-1)p^{-1-3s} + p^{-1}(p-2)p^{-2-8s} + \frac{(1-p^{-1})p^{-3-9s}}{1-p^{-1-s}} + \frac{(1-p^{-1})p^{-3-11s}}{1-p^{-1-3s}} + p^{-2-6s} \int_{\mathbb{Z}_p} |f_{c,h}(x)|_p^s \, dx \, . \tag{3.4}
\]
Finally, by applying SPF to the integral involving $f_{c,h}(x)$ in (2.3), we obtain that

$$Z(s, f) = p^{-1}(p - 3) + \frac{(1 - p^{-1})p^{-1-s}}{1 - p^{-1-s}} + p^{-1}(p - 1)p^{-1-4s}$$

$$+ p^{-1}(p - 1)p^{-1-3s} + p^{-1}(p - 2)p^{-2-8s} + \frac{(1 - p^{-1})p^{-3-9s}}{1 - p^{-1-s}}$$

$$+ \frac{(1 - p^{-1})p^{-3-11s}}{1 - p^{-1-3s}} + p^{-1}(p - 2)p^{-2-6s} + \frac{(1 - p^{-1})p^{-3-7s}}{1 - p^{-1-s}}$$

$$+ \frac{(1 - p^{-1})p^{-3-8s}}{1 - p^{-1-2s}}.$$

(3.5)

**Remark 3.1.** If $\alpha = \frac{a}{b} \in \mathbb{Q}$, and $v_p(\alpha) < 0$, then

$$| x - \alpha |_p = | \alpha |_p,$$ for every $x \in \mathbb{Z}_p$.

On the other hand, a polynomial of the form

$$f(x) = \alpha_0 \prod_{i=1}^{r} (x - \alpha_i)^{e_i} \in \mathbb{Q}[x],$$

can be decomposed as $f(x) = \alpha_0 f_-(x)f_+(x)$, where

$$f_-(x) = \prod_{\{\alpha_i | v_p(\alpha_i) < 0\}} (x - \alpha_i)^{e_i}, \text{ and } f_+(x) = \prod_{\{\alpha_i | v_p(\alpha_i) \geq 0\}} (x - \alpha_i)^{e_i}.$$

(3.7)

From (3.6) and (3.7) follow that

$$Z(s, f) = | \alpha_0 \prod_{\{\alpha_i | v_p(\alpha_i) < 0\}} \alpha_i^{e_i} |_p Z(s, f_+).$$

Thus, from a computational point of view, we may assume without loss of generality that all roots of $f(x)$ are $p$-adic integers.

4. Trees and $p$-adic Numbers

The tree $U = U(p)$ of residue classes modulo powers of a given prime number $p$ is defined as follows. Consider the diagram

$$\{0\} = \mathbb{Z}/p^0 \mathbb{Z} \xleftarrow{\phi_1} \mathbb{Z}/p^1 \mathbb{Z} \xleftarrow{\phi_2} \mathbb{Z}/p^2 \mathbb{Z} \xleftarrow{\phi_3} \cdots$$

where $\phi_l$ the are the natural homomorphisms. The vertices of $U$ are the elements of $\mathbb{Z}/p^l \mathbb{Z}$, for $l = 0, 1, 2, \cdots$, and the directed edges are $u \rightarrow v$ where $u \in \mathbb{Z}/p^l \mathbb{Z}$ and $\phi_l(u) = v$, for some $l > 0$. Thus $U$ is a rooted tree with root $\{0\}$. Exactly one directed edge emanates from each vertex of $U$; except from the vertex $\{0\}$, from which no edge emanates. In addition, every vertex is the end point of exactly $p$ directed edges.

Given two vertices $u, v$ the notation $u > v$ will mean that there is a sequence of vertices and edges of the form

$$u \rightarrow u^{(1)} \rightarrow \cdots \rightarrow u^{(m)} = v.$$ 

The notation $u \geq v$ will mean that $u = v$ or $u > v$. The level $l(u)$ of a vertex $u$ is $m$ if $u \in \mathbb{Z}/p^m \mathbb{Z}$. The valence $\text{Val}(u)$ of a vertex $u$ is defined as the number of directed edges whose end point is $u$. 

A subtree, or simply a tree, is defined as a nonempty subset $T$ of vertices of $U$, such that when $u \in T$ and $u > v$, then $v \in T$. Thus $T$ together with the directed edges $u \rightarrow v$, where $u, v \in T$, is again a tree with root $\{0\}$.

A tree $T$ is named a \textit{weighted tree}, if there exists a weight function $W : T \rightarrow \mathbb{N}$. The value $W(u)$ is called the weight of vertex $u$.

If $x \in \mathbb{Z}_p$, and $x_l$ denotes its residue class modulo $p^l$, then every vertex of $U$ is of the type $x_l$ with $l \in \mathbb{N}$.

A \textit{stalk} is defined as a tree $K$ having at most one vertex at each level. Thus a stalk is either finite, of the type
$$\{0\} \leftarrow x_1 \leftarrow \cdots \leftarrow x_l,$$
with $x \in \mathbb{Z}$, and infinite stalks as
$$\{0\} \leftarrow x_1 \leftarrow x_2 \leftarrow \cdots,$$
with $x \in \mathbb{Z}_p$. Thus there is a $1-1$ correspondence between infinite stalks and $p$–adic integers.

4.1. \textbf{Tree Attached to a Polynomial.} Let

\begin{equation}
(4.1) \quad f(x) = a_0 \prod_{i=1}^r (x - \alpha_i)^{e_i} \in \mathbb{Q}[x]
\end{equation}

be a non-constant polynomial, in one variable, of degree $d_f$, such that $v_p(\alpha_i) \geq 0$, $i = 1, 2, \cdots, r$. We associate to $f(x)$ and a prime number $p$ the integer

$$l_f = \begin{cases} 1 + \max\{v_p(\alpha_i - \alpha_j) \mid i \neq j, 1 \leq i, j \leq r\}, & \text{if } r \geq 2; \\ 1, & \text{if } r = 1. \end{cases}$$

We set
$$
\alpha_i = a_{0,i} + a_{1,i} p + \cdots + a_{j,i} p^j + \cdots + a_{l_f,i} p^{l_f} \mod p^{l_f+1},
$$
$a_{j,i} \in \{0, 1, \cdots, (p-1)\}$, $j = 0, 1, \cdots, l_f$, $i = 1, 2, \cdots, r$, for the $p$–adic expansion modulo $p^{l_f+1}$ of $\alpha_i$. We attach a weighted tree $T(f, l_f)$ to $f$ as follows:

\begin{equation}
(4.2) \quad T(f, l_f, p) = T(f, l_f) = \bigcup_{i=1}^r K(\alpha_i, l_f),
\end{equation}

where $K(\alpha_i, l_f)$ denotes the stalk corresponding to the $p$-adic expansion of $\alpha_i$ modulo $p^{l_f+1}$. Thus $T(f, l_f)$ is a rooted tree. We introduce a weight function on $T(f, l_f)$, by defining the weight of a vertex $u$ of level $m$ as

\begin{equation}
(4.3) \quad W(u) = \begin{cases} \sum_{i \mid \alpha_i \equiv u \mod p^m} e_i, & \text{if } m \geq 1; \\ 0, & \text{if } m = 0. \end{cases}
\end{equation}

Given a vertex $u \in T(f, l_f)$, we define the stalk generated by $u$ to be
$$B_u = \{v \in T(f, l_f) \mid u \geq v\}.$$
We associate a weight $W^*(B_u)$ to $B_u$ as follows:

\begin{equation}
W^*(B_u) = \sum_{v \in B_u} W(v).
\end{equation}

4.2. Computation of Trees Attached to Polynomials. Our next step is to show that a tree $T(f,l_f)$ attached to a polynomial $f(x)$, of type (4.1), can be computed in polynomial time. There are well known programming techniques to construct and manipulate trees and forests (see e.g. [8, Volume 1]), for this reason, we shall focus on showing that such computations can be carry out in polynomial time, and set aside the implementation details of a particular algorithm for this task. We shall include in the computation of $T(f,l_f)$, the computation of the weights of the stalks generated by its vertices; because all these data will be used in the computation of the local zeta function of $f$.

**Proposition 4.1.** The computation of a tree $T(f,l_f)$ attached to a polynomial $f(x)$, of type (4.1), from the $p$-adic expansions modulo $p^{l_f+1}$ of its roots

$\alpha_i = a_{0,i} + a_{1,i} p + \cdots + a_{l_f,i} p^{l_f} \mod p^{l_f+1}$

and multiplicities $e_i$, $i = 1, 2, \cdots, r$, involves $O(l_f^2 d_f^3)$ arithmetic operations on integers with binary length $O(\max\{\log p, \log(l_f d_f)\})$.

**Proof.** We assume that $T(f,l_f)$ is finite set of the form

\begin{equation}
T = \{\text{Level}_0, \cdots, \text{Level}_j, \cdots, \text{Level}_{l_f+1}\},
\end{equation}

where Level$_j$ represents the set of all vertices with level $j$. Each Level$_j$ is a set of the form

Level$_j = \{u_{j,1}, \cdots, u_{j,i}, \cdots, u_{j,m_j}\}$,

and each $u_{j,i}$ is a weighted vertex for every $i = 1, \cdots, m_j$. A weighted vertex $u_{j,i}$ is a set of the form

$u_{j,i} = \{W(u_{j,i}), Val(u_{j,i}), W^*(B_{u_{j,i}})\}$,

where $W(u_{j,i})$ is the weight of $u_{j,i}$, $Val(u_{j,i})$ is its valence, and $W^*(B_{u_{j,i}})$ is the weight of stalk $B_{u_{j,i}}$. The weight of the stalk generated by $u_{j,i}$ can be written as

$W^*(B_{u_{j,i}}) = \sum_{v \in B_{u_{j,i}}} W(v)$.

For the computation of a vertex $u_{j,i}$ of level $j$, we proceed as follows. We put $I = \{1, 2, \cdots, r\}$, and

$M_j = \{\alpha_i \mod p^j \mid i \in I\}$.

For each $0 \leq j \leq l_f + 1$, we compute a partition of $I$ of type

\begin{equation}
I = \bigcup_{i=1}^{l_f} I_{j,i},
\end{equation}

such that

$\alpha_t \mod p^j = \alpha_s \mod p^j$,

for every $t, s \in I_{j,i}$. Each subset $I_{j,i}$ corresponds to a vertex $u_{j,i}$ of level $j$. This computation requires $O(l_f r^2)$ arithmetic operations on integers with binary length $O(\log p)$. Indeed, the cost of computing a “yes or no” answer for the question: $\alpha_t$
Thus the computation of the weight of a vertex requires $O(jr^2)$ arithmetic operations on integers with binary length $O(\log p)$. In the worst case, there are $r$ vectors $M_j$, and the computation of partition (4.6), for a fixed $j$, involves the comparison of $\alpha_l$ with $\alpha_t$ for $l = t + 1, t + 2, \ldots, r$. This computation requires $O(jr^2)$ arithmetic operations on integers with binary length $O(\log p)$. Since $j \leq l_f + 1$, the computation of partition (4.6) requires $O(l_f r^2)$ arithmetic operations on integers with binary length $O(\log p)$.

The weight of the vertex $u_{j,i}$ is given by the expression

$$W(u_{j,i}) = \sum_{k \in I_{j,i}} e_k.$$

Thus the computation of the weight of a vertex requires $O(r)$ additions of integers with binary length $O(\log d_f r)$.

For the computation of the valence of $u_{j,i}$, we proceed as follows. The valence of $u_{j,i}$ can be expressed as

$$\text{Val}(u_{j,i}) = \text{Card}\{I_{j+1,l} \mid I_{j+1,l} \subseteq I_{j,i}\},$$

where $I_{j+1,l}$ runs through all possible sets that correspond to the vertices $u_{j+1,l}$, with level $j + 1$. Thus the computation of $\text{Val}(u_{j,m})$ involves the computation of a “yes or no” answer for the question $I_{j+1,l} \subseteq I_{j,i}$? The computation of a “yes or no” answer involves $O(r)$ comparisons of integers with binary length $O(\log r)$. Therefore the computation of $\text{Val}(u_{j,i})$ involves $O(r)$ comparisons and $O(r)$ additions of integers with binary length $O(\log r)$.

For the computation of the weight of $B_{u_{j,i}}$, we observe that $W^*(B_{u_{j,i}})$ is given by the formula

$$W^*(B_{u_{j,i}}) = \sum_{l=0}^{j-1} \sum_{I_{l,k} \subseteq I_{l,k}} W(I_{l,k}),$$

where $W(I_{l,k}) = W(v_{l,k})$, and $v_{l,k}$ is the vertex corresponding to $I_{l,k}$. Thus the computation of $W^*(B_{u_{j,i}})$ involves $O(l_f)$ additions of integers with binary length $O(\log l_f d_f)$, and $O(l_f r)$ comparisons of integers with binary length $O(\log r)$.

From the above reasoning follows that the computation of a vertex of a tree $T(f, l_f)$ involves at most $O(l_f r^2)$ arithmetic operations (additions and comparisons) on integers with binary length $O(\max\{\log p, \log l_f d_f\})$. Finally, since the number of vertices of $T(f, l_f)$ is at most $O(l_f d_f)$, it follows that the computation of a tree of type $T(f, l_f)$ involves $O(l_f^2 d_f^2)$ arithmetic operations on integers with binary length $O(\max\{\log p, \log l_f d_f\})$.

5. Generating Functions and Trees

In this section we attach to a weighted tree $T(f, l_f)$ and a prime $p$ a generating function $G(s, T(f, l_f), p) \in \mathbb{Q}(p^{-s})$ defined as follows.

We set

$$\mathcal{M}_{T(f, l_f)} = \left\{ u \in T(f, l_f) \mid W(u) = 1, \text{and there no exists } v \in T(f, l_f) \text{ with } W(v) = 1, \text{such that } u > v \right\},$$

and
\[ L_u(p^{-s}) = \begin{cases} 
\frac{(1-p^{-1})p^{-s}f(u)-W^s(B_u)s}{(1-p^{-1}W^s(B_u)s)}, & \text{if } l(u) = 1 + f, \text{ and } W(u) \geq 2; \\
1 - (p - Val(u)p^{-l(u)}-W^s(B_u)s), & \text{if } 0 \leq l(u) \leq f, \text{ and } W(u) \neq 1; \\
\frac{(1-p^{-1})p^{-s}f(u)-W^s(B_u)s}{1-p^{-s}}, & \text{if } u \in M_{T(f,l_f)}; \\
0, & \text{if } W(u) = 1, \text{ and } u \notin M_{T(f,l_f)}. 
\end{cases} \]

With all the above notation, we define the generating function attached to \( T(f,l_f) \) and \( p \) as

\[ G(s,T(f,l_f),p) = \sum_{u \in T(f,l_f)} L_u(p^{-s}). \]

Our next goal is to show that \( G(s,T(f,l_f),p) = Z(s,f) \). The proof of this fact requires the following preliminary result.

**Proposition 5.1.** The generating function attached to a tree \( T(f,l_f) \) and a prime \( p \) satisfies

\[ G(s,T(f,l_f),p) = p^{-1}v(T) + \delta(T) \frac{(1-p^{-1})p^{-1-s}}{1-p^{-1-s}} + \sum_{\xi \in S} p^{-1-\varepsilon\xi}G(s,T(f_s,l_f-1),p). \]

**Proof.** Let \( A_f = \{ u \in T(f,l_f) \mid l(u) = 1, W(u) = 1 \} \), and \( B_f = \{ u \in T(f,l_f) \mid l(u) = 1, W(u) \geq 2 \} \). We have the following partition for \( T(f,l_f) \):

\[ T(f,l_f) = \{0\} \bigcup A_f \bigcup \left( \bigcup_{u \in B_f} T_u \right), \]

with

\[ T_u = \{ v \in T(f,l_f) \mid v \geq u \}. \]

Each \( T_u \) is a rooted tree with root \( \{ u \} \). From the partition and the definition of \( G(s,T(f,l_f),p) \), it follows that

\[ G(s,T(f,l_f),p) = p^{-1}(p - Val(\{0\})) + \operatorname{Card}(A_f) \frac{(1-p^{-1})p^{-1-s}}{1-p^{-1-s}} + \sum_{u \in B_f} G(s,T_u), \]

with \( G(s,T_u) = \sum_{v \in T_u} L_v(p^{-s}). \)

Since there exists a bijective correspondence between the roots of \( \mathcal{T}(x) \equiv 0 \mod p \) and the vertices of \( T(f,l_f) \) with level 1,

\[ p - Val(\{0\}) = \nu(T), \text{ and } \operatorname{Card}(A_f) = \delta(T). \]

Now, if the vertex \( u \) corresponds to the root \( \mathcal{T}(\xi) \equiv 0 \mod p \), then

\[ T_u = \left( \bigcup_{\{ \alpha_i \mid \alpha_i \equiv \xi \mod p \}} K(\alpha_i,l_f) \right) \setminus \{0\}. \]
On the other hand, we have that
\[(5.7)\quad T(f_\xi, l_f - 1) = \bigcup_{\{\alpha_i | \alpha_i \equiv \xi \mod p\}} K(\frac{\alpha_i - \xi}{p}, l_f - 1).\]

Now we remark that the map \(\alpha_i \rightarrow \frac{\alpha_i - \xi}{p}\) induces an isomorphism between the trees \(T_u\) and \(T(f_\xi, l_f - 1)\), that preserves the weights of the vertices; and thus we may suppose that \(T_u = T(f_\xi, l_f - 1)\). The level function \(l_T\) of \(T(f_\xi, l_f - 1)\) is related to the level function \(l_{T_u}\) of \(T_u\) by means of the equality \(l_T - l_{T_u} = -1\). In addition, \(B_f = S\), where \(S\) is the subset of \{0, 1, \ldots, p - 1\} \(\subseteq \mathbb{Z}_p\) whose reduction modulo \(p\mathbb{Z}_p\) is equal to the set of roots of \(f(\xi) = 0\) with multiplicity greater or equal than two. Therefore, it holds that
\[(5.8)\quad G(s, T_u) = p^{-1-\varepsilon_\xi} G(s, T(f_\xi, l_f - 1), p).\]

The result follows from \((5.4)\) by the identities \((5.5)\) and \((5.8)\).

**Lemma 5.1.** Let \(p\) be a fixed prime number and \(v_p\) the corresponding \(p\)-adic valuation, and
\[f(x) = \alpha_0 \prod_{i=1}^{r} (x - \alpha_i)^{e_\xi} \in \mathbb{Q}[x] \setminus \mathbb{Q},\]
a polynomial such that \(v_p(\alpha_i) \geq 0\), for \(i = 1, \ldots, r\). Then
\[Z(s, f) = G(s, T(f, l_f), p).\]

**Proof.** We proceed by induction on \(l_f\).

Case \(l_f = 1\)
If \(r = 1\) the proof follows immediately, thus we may assume that \(r \geq 2\). Since \(l_f = 1\), it holds that \(v_p(\alpha_i - \alpha_j) = 0\), for every \(i, j\), satisfying \(i \neq j\), and thus \(\overline{\alpha_i} \neq \overline{\alpha_j}\), if \(i \neq j\). By applying SPF, we have that
\[(5.9)\quad Z(s, f) = \frac{1}{(1 - p^{-1})(1 - p^{-1-s})} + \sum_{\xi \in S} p^{-1-\varepsilon_\xi} \frac{(1 - p^{-1})}{(1 - p^{-1-\varepsilon_\xi})},\]
where each \(e_\xi = e_j \geq 2\), for some \(j\), and \(\alpha_j = \xi + p\beta_j\).

On the other hand, \(T(f, l_f)\) is a rooted tree with \(r\) vertices \(v_j\), satisfying \(l(v_j) = 1\), and \(W(v_j) = e_j\), for \(j = 1, \ldots, r\). These observations allow one to deduce that \(Z(s, f) = G(s, T(f, l_f), p)\).

By induction hypothesis, we may assume that \(Z(s, f) = G(s, T(f, l_f), p)\), for every polynomial \(f\) satisfying both the hypothesis of the lemma, and the condition \(1 \leq l_f \leq k\), \(k \in \mathbb{N}\).

Case \(l_f = k + 1\), \(k \in \mathbb{N}\)
Let \(f(x)\) be a polynomial satisfying the lemma’s hypothesis, and \(l_f = k + 1\), \(k \geq 1\). By applying SPF, we obtain that
\[(5.10)\quad Z(s, f) = \frac{1}{(1 - p^{-1})(1 - p^{-1-s})} + \sum_{\xi \in S} p^{-1-\varepsilon_\xi} \int | f_\xi(x) |^s_p \, dx.\]

Now, since \(l_{f_\xi} = l_f - 1\), for every \(\xi \in S\), it follows from the induction hypothesis applied to each \(f_\xi(x)\) in \((5.10)\), that
\[(5.11)\quad Z(s, f) = \frac{1}{(1 - p^{-1})(1 - p^{-1-s})} + \sum_{\xi \in S} p^{-1-\varepsilon_\xi} G(s, T(f_\xi, l_f - 1), p).\]
Finally, from identity (5.2), and (5.11), we conclude that
\[(5.12)\quad Z(s, f) = G(s, T(f, l_f), p).\]

The following proposition gives a complexity estimate for the computation of 
\[G(s, T(f, l_f), p).\]

**Proposition 5.2.** The computation of the generating function 
\[G(s, T(f, l_f), p)\]
from \(T(f, l_f)\), involves \(O(l_f d_f)\) arithmetic operations on integers with binary length 
\[O(\max\{\log p, \log(l_f d_f)\}).\]

**Proof.** This is a consequence of proposition (5.11) and the definition of generating function. 

6. Computation of \(p\)-adic Expansions

In this section we estimate the complexity of the steps 2 and 3 in the algorithm Compute \(Z(s, f)\).

**Proposition 6.1.** Let 
\[B = \max_{1 \leq i, j \leq r, i \neq j} \{|c_{j,i}|, |d_{j,i}|\} \quad |\alpha_j - \alpha_i = \frac{c_{j,i}}{d_{j,i}}, c_{j,i}, d_{j,i} \in \mathbb{Z} \setminus \{0\}|.\]

The computation of the integer \(l_f\) involves \(O(d_f^2 \log B / \log p)\) arithmetic operations on integers with binary length \(O(\max\{\log B, \log p\}).\)

**Proof.** First, we observe that for \(c \in \mathbb{Z} \setminus \{0\}, \) the computation of \(v_p(c)\) involves \(O(\log c / \log p)\) divisions of integers of binary length \(O(\max\{\log |c|, \log p\}).\) Thus the computation of \(v_p(\frac{c}{d}) = v_p(c) - v_p(d),\) involves \(O(\max\{\log |c|, \log |d|\})\) divisions and subtraction of integers with binary length 
\[O(\max\{|c|, \log d|, \log p\}).\]

From these observations follow that the computation of \(v_p(\alpha_j - \alpha_i), i \neq j, 1 \leq i, j \leq r,\) involves \(O(r^2 \log B / \log p)\) arithmetic operations on integers with binary length \(O(\max\{\log B, \log p\}).\) Finally, the computation of the maximum of the \(v_p(\alpha_j - \alpha_i), i \neq j, 1 \leq i, j \leq r,\) involves \(O(\log r)\) comparisons of integers with binary length \(O(\max\{\log B, \log p\}).\) Therefore the computation of the integer \(l_f\) involves at most \(O(d_f^2 \log B / \log p)\) arithmetic operations on integers with binary length \(O(\max\{\log B, \log p\}).\)

**Proposition 6.2.** Let \(p\) be a fixed prime and \(\gamma = \frac{c}{d} \in \mathbb{Q},\) with \(c, b \in \mathbb{Z} \setminus \{0\},\) and \(v_p(\gamma) \geq 0.\) The \(p\)-adic expansion 
\[\gamma = a_0 + a_1 p + \cdots + a_j p^j + \cdots + a_m p^m,\]
modulo \(p^{m+1}\) involves \(O(m + \log(\max\{|b|, p\}))\) arithmetic operations on integers with binary length \(O(\max\{\log |c|, \log |b|, \log p\}).\)
5.1. The complexity estimates are obtained as follows: the number of arithmetic operations needed in the steps 2 (cf. proposition 6.1), 3 (cf. corollary 6.1), 4 (cf. proposition 6.2), and 6 is at most \( O(\log(\max\{ | b |, p \})) \) (cf. [3] Volume 2, section 4.5.2).

We set \( \gamma = \gamma_0 = \frac{s}{b}, c_0 = c \), and define \( a_0 \equiv yc \mod p \). With this notation, the \( p \)-adic digits \( a_i, i = 1, \cdots, m \), can be computed recursively as follows:

\[
\gamma_i = \frac{(c_{i-1} - a_{i-1})}{p}, \quad \frac{a_i}{b} = \gamma_i \mod p.
\]

Thus the computation of the \( p \)-adic expansion of \( \gamma \) needs \( O(m + \log(\max\{ | b |, p \})) \) arithmetic operations on integers with binary length \( O(\max\{\log | \gamma |, \log | b |, \log p \}) \).

\[\blacksquare\]

**Corollary 6.1.** Let \( p \) be a fixed prime number and \( v_p \) the corresponding \( p \)-adic valuation, and

\[ f(x) = a_0 \prod_{i=1}^{r} (x - \alpha_i)^{c_i} \in \mathbb{Q}[x], \]

a non-constant polynomial such that \( v_p(\alpha_i) \geq 0, i = 1, \cdots, r \). The computation of the \( p \)-adic expansions modulo \( p^{l_f+1} \) of the roots \( \alpha_i, i = 1, 2, \cdots, r, \) of \( f(x) \) involves \( O(d_f l_f + d_f \log(\max\{ | B |, p \})) \) arithmetic operations on integers with binary length \( O(\max\{\log B, \log p \}) \).

**Proof.** The corollary follows directly from the two previous propositions. \[\blacksquare\]

7. **Computing local zeta functions of polynomials with splitting \( \mathbb{Q} \)**

In this section we prove the correctness of the algorithm \( \text{Compute}_Z(s, f) \) and estimate its complexity.

**Theorem 7.1.** The algorithm \( \text{Compute}_Z(s, f) \) outputs the meromorphic continuation of the Igusa local zeta function \( Z(s, f) \) of a polynomial \( f(x) \in \mathbb{Z}[x] \), in one variable, with splitting field \( \mathbb{Q} \). The number of arithmetic operations needed by the algorithm is

\[ O(d_f^3 + d_f^2 \log(\max f \parallel s)) + l_f d_f^3 + d_f^2 \log(\max\{ B, p \})) \]

and the integers on which these operations are performed have a binary length \( O(\max\{\log p, \log l_f d_f, \log B, d_f^3 + d_f^2 \log(\max f \parallel s)) \}) \).

**Proof.** By remark 6.1, we may assume without loss of generality that

\[ f(x) = a_0 \prod_{i=1}^{r} (x - \alpha_i)^{c_i} \in \mathbb{Q}[x] \setminus \mathbb{Q}, \]

with \( v_p(\alpha_i) \geq 0, i = 1, \cdots, r \). The correctness of the algorithm follows from lemma 6.1. The complexity estimates are obtained as follows: the number of arithmetic operations needed in the steps 2 (cf. proposition 6.1), 3 (cf. corollary 6.1), 4 (cf. proposition 6.2), and 6 is at most

\[ O(l_f^2 d_f^2 + d_f^2 \log(\max\{ B, p \})) \].
and these operations are performed on integers whose binary length is at most

\[ O(\max\{\log p, \log f, \log B\}) \]

The estimates for the whole algorithm follow from the above estimates and those of the factoring algorithm by A. K. Lenstra, H. Lenstra and L. Lovász (see theorem 3.6 of [17]).

8. Stream Ciphers and Poincaré series

There is a natural connection between Poincaré series and stream ciphers. In order to explain this relation, we recall some basic facts about stream ciphers [18]. Let \( \mathbb{F}_p^n \) be a finite field with \( p^n \) elements, with \( p \) a prime number. For any integer \( r > 0 \) and \( r \) fixed elements \( q_i \in \mathbb{F}_p^n, i = 1, \ldots, r \) (called taps), a Linear Feedback Shift Register, abbreviated LFSR, of length \( r \) consists of \( r \) cells with initial contents \( \{a_i \in \mathbb{F}_p^n \mid i = 1, \ldots, r\} \). For any \( n \geq r \), if the current state is \((a_{n-1}, \ldots, a_{n-r})\), then \( a_n \) is determined by the linear recurrence relation

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

The device outputs the rightmost element \( a_{n-r} \), shifts all the cells one unit right, and feeds \( a_n \) back to the leftmost cell.

Any configuration of the \( r \) cells forms a state of the LSFR. If \( q_r \neq 0 \), the following polynomial \( q(x) \in \mathbb{F}_p^n[x] \) of degree \( r \) appears in the analysis of LFSRs:

\[ q(x) = q_0 + q_1x + \cdots + q_rx^r \quad \text{with} \quad q_0 = -1. \]

This polynomial is called the connection polynomial. An infinite sequence \( A = \{a_i \in \mathbb{F}_p^n \mid i \in \mathbb{N}\} \) has period \( T \) if for any \( i \geq 0 \), \( a_{i+T} = a_i \). Such a sequence is called periodic. If this is only true for \( i \) greater than some index \( i_0 \), then the sequence is called eventually periodic. The following facts about an LFSR of length \( r \) are well-known [13].

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

2. The Poincaré series \( g(x) = \sum_{i=0}^{\infty} a_i x^i \) associated with the output sequence is called the generating function of the sequence. It is a rational function over \( \mathbb{F}_p^n \) of the form \( g(x) = \frac{L(x)}{R(x)} \), with \( L(X) \in \mathbb{F}_p^n[x], \deg(R(X)) < r \). The output sequence is strictly periodic if and only if \( \deg(L(X)) < \deg(R(X)) \).

3. There is a one-to-one correspondence between LFSRs of length \( r \) with \( q_r \neq 0 \) and rational functions \( \frac{L(x)}{R(x)} \) with \( \deg(R(X)) = r \) and \( \deg(L(X)) < r \).

We set \( \mathbb{F}_p^n(x) \) for the field of rational functions over \( \mathbb{F}_p^n \), and \( N^\infty(\mathbb{F}_p^n) \) for the set of sequences of the form \( \{b_0, \ldots, b_u\}, b_i \in \mathbb{F}_p^n, \ 0 \leq i \leq u, u \in \mathbb{N} \). From the above considerations, it is possible to identify an LFSR with a function \( F_u, u \in \mathbb{N} \), defined as follows:

\[
\begin{align*}
F_u : \mathbb{F}_p^n(x) & \rightarrow N^\infty(\mathbb{F}_p^n) \\
n \sum_{i=0}^{\infty} a_i x^i & \mapsto \{a_0, \cdots, a_u\}. 
\end{align*}
\]
We set
\[ \mathcal{H} = \{ H(t, f) \mid f(x) \in \mathbb{Z}[x], \text{ in one variable, with splitting field } \mathbb{Q} \}, \]
and \( N^\infty(\mathbb{Z}) \) for the set of finite sequences of integers. Also, for each \( u \in \mathbb{N} \), and a prime number \( p \), we define
\[
F_{u,p} : \mathcal{H} \rightarrow \mathbb{N}^\infty(\mathbb{Z}) \quad H(t, f) \mapsto \{ N_0(f, p), N_1(f, p), \ldots, N_u(f, p) \}.
\]
Thus the mappings \( F_{u,p} \) can be seen as LFSRs, or stream ciphers, over \( \mathbb{Z} \). If we replace each \( N_u(f, p) \) by its binary representation, then the \( F_{u,p} \) are LFSRs. For practical purposes it is necessary that \( F_{u,p} \) can be computed efficiently, i.e., in polynomial time. With the above notation our second result is the following.

**Theorem 8.1.** For every \( H(t, f) \in \mathcal{H} \), the computation of \( F_{u,p}(H(t, f)) \) involves \( O(u^2d_f l_f) \) arithmetic operations, and the integers on which these operations are performed have binary length

\[ O(\max\{ (l_f + u) \log p, \log(d_f l_f) \}). \]

The proof of this theorem will be given at the end of this section. This proof requires some preliminary results. We set \( t = q^{-s} \), and

\[ Z(s, f) = Z(t, f) = \sum_{m=0}^{\infty} c_m(f, p)t^m, \]
with \( c_m(f, p) = \text{vol}(\{ x \in \mathbb{Z}_p \mid v_p(f(x)) = m \}) \).

**Proposition 8.1.** Let \( f(x) \in \mathbb{Z}[x] \setminus \mathbb{Z} \) be a polynomial in one variable and \( p \) a prime number. The following formula holds for \( N_n(f, p) \):

\[
N_n(f, p) = \begin{cases} 
1, & \text{if } n = 0; \\
p^n \left(1 - \sum_{j=1}^{n} c_{j-1}(f, p)\right), & \text{if } n \geq 1.
\end{cases}
\]

**Proof.** The result follows by comparing the coefficient of \( t^n \) of the series

\[ \sum_{n=0}^{\infty} \frac{N_n(f, p)}{p^n} t^n \quad \text{and} \quad \sum_{n=0}^{\infty} d_n t^n, \]
in the following equality:

\[ H(t, f) = \sum_{n=0}^{\infty} \frac{N_n(f, p)}{p^n} t^n = \frac{1-t \left(\sum_{m=0}^{\infty} c_m(f, p)t^m\right)}{1-t} = \sum_{n=0}^{\infty} d_n t^n. \]
We associate to each \( u \in T(f, l_f) \), and \( j \in \mathbb{N} \), a rational integer \( a_j(u) \) defined as follows:

\[
(8.4) \quad a_j(u) = \begin{cases} 
\frac{(p-1)}{p^{l(u)+1-\text{Val}(u)}}, & \text{if } l(u) = 1 + l_f, \ W(u) \geq 2, j = W^*(B_u) + y(u)W(u), \\
\frac{(p-1)}{p^{l(u)+1}}, & \text{if } 0 \leq l(u) \leq l_f, \ W(u) \neq 1, j = W^*(B_u); \\
\frac{(p-V_{\text{a}}(u))}{p^{l(u)+1-\text{Val}(u)}}, & \text{if } u \in \mathcal{M}_{T(f,l_f)}, j = W^*(B_u) + y(u), \\
0, & \text{if } W(u) = 1, \text{ and } u \notin \mathcal{M}_{T(f,l_f)}; \\
0, & \text{in other cases.}
\end{cases}
\]

**Proposition 8.2.** Let \( f(x) \in \mathbb{Z}[x] \backslash \mathbb{Z} \) be a polynomial in one variable, with splitting field \( \mathbb{Q} \), and \( p \) a prime number. The following formula holds:

\[
(8.5) \quad c_j(f, p) = \sum_{u \in T(f, l_f)} a_j(u), \ j \geq 0.
\]

**Proof.** As a consequence of lemma 5.1, we have the following identity:

\[
(8.6) \quad Z(t, f) = \sum_{u \in T(f, l_f)} L_u(t),
\]

with

\[
(8.7) \quad L_u(t) = \begin{cases} 
\frac{(p-1)t^{W^*(B_u)}}{p^{l(u)+1-\text{Val}(u)}}, & \text{if } l(u) = 1 + l_f, \ W(u) \geq 2; \\
\frac{(p-V_{\text{a}}(u))}{p^{l(u)+1-\text{Val}(u)}}t^{W^*(B_u)}, & \text{if } 0 \leq l(u) \leq l_f, \ W(u) \neq 1; \\
\frac{(p-1)}{p^{l(u)+1}(1-p^{-1})}, & \text{if } u \in \mathcal{M}_{T(f,l_f)}; \\
0, & \text{if } W(u) = 1, \text{ and } u \notin \mathcal{M}_{T(f,l_f)}.
\end{cases}
\]

The result follows by comparing the coefficient of \( t^j \) in the series \( Z(t, f) = \sum_{m=0}^{\infty} c_m(f, p)t^m \), and \( Z(t, f) = \sum_{u \in T(f, l_f)} L_u(t) \).

**Proposition 8.3.** Let \( f(x) \in \mathbb{Z}[x] \backslash \mathbb{Z} \) be a polynomial in one variable, with splitting field \( \mathbb{Q} \), and \( p \) a prime number.

1. The computation of \( N_n(f, p), n \geq 1 \), from the \( c_{j-1}(f, p) \), \( j = 1, \cdots, n \), involves \( O(n) \) arithmetic operations on integers with binary length \( O(n \log p) \).
2. The computation of \( c_j(f, p), j \geq 0 \), from \( Z(t, f) \), involves \( O(d_j l_f) \) arithmetic operations on integers with binary length \( O(\max\{(j + l_f) \log p, \log p, \log(d_j l_f)\}) \).
(3) The computation of any $N_n(f, p)$, $n \geq 1$, from $Z(t, f)$, involves $O(nd_f l_f)$ arithmetic operations on integers with binary length $O(\max\{(n + l_f) \log p, \log(d_f l_f)\})$.

Proof. (1) By (8.4) and (8.5), $c_j(f, p) = \frac{w}{p^j}$, $v_j, m_j \in \mathbb{N}$. In addition,

$$c_{j-1}(f, p) = p^{-j+1}N_{j-1}(f, p) - p^{-j}N_j(f, p).$$

Thus $p^nc_{j-1}(f, p) \in \mathbb{N}$, for $j = 1, \ldots, n$, and $m_j \leq n$, for $j = 1, \ldots, n$. From (8.8), it follows that

$$N_n(f, p) = p^n - \sum_{j=1}^{n} p^n c_{j-1}(f, p), \quad n \geq 1.$$  

The above formula implies that the computation of $N_n(f, p)$, $n \geq 1$, from the $c_{j-1}(f, p)$, $j = 1, \ldots, n$, involves $O(n)$ arithmetic operations on integers with binary length $O(n \log p)$.

(2) The computation of $a_j(u)$ from $L_u(t)$ (i.e. from $Z(t, f)$, cf. (8.4)) involves $O(1)$ arithmetic operations (cf. (8.4), (8.5)) on integers of binary length $O(\max\{(\log p, \log(d_f l_f))\})$. Indeed, since the numbers $l(u), W^u(B_u), W(u), u \in T(f, l_f)$ are involved in this computation, we know by proposition 4.1 that their binary length is bounded by $O(\max\{\log p, \log(d_f l_f)\})$.

The cost of computing $c_j(f, p)$ from $L_u(t), u \in T(f, l_f)$ (i.e. from $Z(t, f)$) is bounded by the number of vertices of $T(f, l_f)$ multiplied by an upper bound for the cost of computing $a_j(u)$ from $L_u(t)$, for any $j$, and $u$ (cf. (8.5)). Therefore, from the previous discussion the cost of computing $c_j(f, p)$ from $Z(t, f)$ is bounded by $O(d_f l_f)$ arithmetic operations. These arithmetic operations are performed on integers of binary length bounded by $O(\max\{(j + l_f) \log p, \log p, \log(d_f l_f)\})$. Indeed, the binary lengths of the numerator and the denominator of $a_j(u) + a_j(u')$, $u, u' \in T(f, l_f)$ are bounded by $(l_f + 1 + j) \log p$ (cf. (8.5)). Thus, the mentioned arithmetic operations for calculating $c_j(f, p)$ from $L_u(t)$ are performed on integers whose binary length is bounded by $O(\max\{(j + l_f) \log p, \log p, \log(d_f l_f)\})$.

(3) The third part follows the first and second parts by (8.8). 

8.1. Proof of Theorem (8.4). The theorem follows from proposition (8.8) (3).

References

[1] Anshel, M., and Goldfeld, D., Zeta functions, one-way functions and pseudorandom number generators, Duke Math. J., 88, 2 (1997), 371-390.
[2] Bierstone, E., Milman, P., Canonical desingularization in characteristic zero by blowing up the maximum strata of a local invariant, Invent. Math. 128 (1997), 207-302.
[3] Bodnár, Gábor; Schicho, Josef. A computer program for the resolution of singularities. Resolution of singularities (Obergurgl, 1997), 231–238, Progr. Math., 181, Birkhäuser, Basel, 2000.
[4] Bodnár, G., Schicho, J., Automated resolution of singularities for hypersurfaces, Journal of Symbolic Computation 30 (2000), 401-428.
[5] Denef J., Report on Igusa’s local zeta function, Séminaire Bourbaki 1990/1991 (730-744) in Asterisque 201-203 (1991), 359-386.
[6] Denef J., On the degree of Igusa’s local zeta functions, Amer. Math. J. 109 (1987), 991-1008.
[7] Denef J., Hoornaert Kathleen, Newton polyhedra and Igusa local zeta function, Journal of Number Theory 89 (2001), 31-64.
[8] Knuth, D., The art of computer programming, 3 volumes, Addison-Wesley, 1999.
[9] Garey, M. R., Johnson, D. S., Computers and intractability: A guide to the theory of NP-Completeness, 1979, W. H. Freedman and Company, New York.
[10] Goldman, Jay R., Numbers of solution of congruences: Poincaré series for strongly nondegenerate forms, Proc. Amer. Math. Soc., 87 (1983), 586-590.

[11] Goldman, Jay R., Numbers of solution of congruences: Poincaré series for algebraic curves, Adv. in Math. 62 (1986), 68-83.

[12] Goldreich, O., Levin, L. A., and Nisan, N., On constructing 1-1 one-way functions, preprint available at [http://www.wisdom.weizmann.ac.il/~oded/cryptography.html](http://www.wisdom.weizmann.ac.il/~oded/cryptography.html).

[13] Goldreich, O., Krawczyk, H., Luby, M., On the existence of pseudorandom number generators, SIAM J. on Computing, Vol. 22 (1993), 1163-1175.

[14] Igusa, Jun-Ichi, An introduction to the theory of local zeta functions, AMS/IP studies in advanced mathematics, v. 14, 2000.

[15] Igusa, J., Complex powers and asymptotic expansions, I Crelles J. Math., 268/269 (1974), 110-130; II, ibid., 278/279 (1975), 357-368.

[16] Igusa, J., A stationary phase formula for $p$-adic integrals and its applications, in Algebraic Geometry and its Applications, Springer-Verlag (1994), 175-194.

[17] Lenstra, A.K., Lenstra, H.W., Lovász, L., Factoring polynomials with rational coefficients, Math. Ann. 261 (1982), 515-534.

[18] Ruepel R., Analysis and design of stream ciphers, Springer-Verlag, New York, 1986.

[19] Saia, M.J., Zuniga-Galindo, W.A., Local zeta functions, Newton polygons and non degeneracy conditions, to appear in Trans. Amer. Math. Soc.

[20] P. Shor, Algorithms for quantum computation, discrete logarithms, and factoring, in Proceedings of 35th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los alamitos, California, 1994, 124-134.

[21] Villamayor, O., Constructiveness of Hironaka’s resolution, Ann. Scient. Ecole Norm. Sup. 4 (1989), 1-32.

[22] von zur Gathen, J., Karpinski, M., Shparlinski, I., Counting curves and their projections, in Proceedings ACM STOC 93, 805-812.

[23] Daqing Wan, Algorithmic theory of zeta functions over finite fields, to appear in MSRI Computational Number Theory Proceedings.

[24] Zuniga-Galindo W. A., Igusa’s local zeta functions of semiquasihomogeneous polynomials, Trans. Amer. Math. Soc. 353, (2001), 3193-3207.

[25] Zuniga-Galindo W. A., Local zeta functions and Newton polyhedra, to appear in Nagoya Math. J.

[26] Zuniga-Galindo W.A., Local zeta function for non-degenerate homogeneous mappings, preprint 2003.

Department of Mathematics and Computer Science, Barry University, 11300 N.E. Second Avenue, Miami Shores, Florida 33161, USA

E-mail address: wzuniga@mail.barry.edu