1.5em

On factorization and root counting modulo prime powers

Undergraduate Project (UGP) report submitted to

Indian Institute of Technology Kanpur

Bachelor of Technology

in

Computer Science and Engineering

by

Sayak Chakrabarti

(170648)

Under the supervision of

Prof. Nitin Saxena

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur

Fall Semester, 2020-2021

December 11, 2020

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the project and giving their details in the references.

Date: December 11, 2020 (Sayak Chakrabarti)
Place: Kanpur (170648)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

KANPUR - 208016, INDIA

CERTIFICATE

This is to certify that the project report entitled "**On factorization and root counting modulo prime powers**" submitted by **Sayak Chakrabarti** (Roll No. 170648) to Indian Institute of Technology Kanpur towards no/partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a record of bona fide work carried out by him under my supervision and guidance during Fall Semester, 2020-2021.

Prof. Nitin Saxena
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur - 208016, India

Date: December 11, 2020

Place: Kanpur

Abstract

Name of the student: **Sayak Chakrabarti** Roll No: **170648**

Degree for which submitted: **Bachelor of Technology**

Department: **Computer Science and Engineering**

Project title: **On factorization and root counting modulo prime powers**

Project supervisor: **Prof. Nitin Saxena**

Month and year of Project submission: **December 11, 2020**

Factorization and root finding are fundamental problems in mathematics and computer algebra. It has some algorithms developed over fields like finite fields, rationals, p-adics etc. However this gets more difficult when we consider rings, in our case of the form of modulo prime powers, where the number of roots as well as factors might become exponential. For example the polynomial $x^2 + px \mod p^2$ has all multiples of p as its roots, which is exponentially many (exponential in $\log p$)! The natural question arises of how to find or count all the roots or find factorizations. Furthermore since rings do not have the same "nice" properties that fields have the problem of finding roots/factors becomes increasingly difficult modulo higher powers of p. In this article we explore problems of factorization and root finding, and present an overview of current state of the art in research and explore ideas to possibly extend the already established results. Existence of roots is a problem explored in Hilbert's Nullstellensatz as well. So we explain algebraic geometric techniques and Hilbert's Nullstellensatz in this article as they might lead to progress related to root finding/counting as well.

Acknowledgements

I am grateful to Prof. Nitin Saxena for arranging time in this uncertain semester and conducting regular discussions hours even though we could not. His ideas have always been interesting and had something new to discuss about every time we talked. I thank him and Ashish Dwivedi to pertaining to my often silly doubts and providing references or clarifications regarding them. My friend Rishabh Batra has also helped me a lot with helpful discussions while working together on the same topic. I also thank my seniors, Subhayan Saha and Abhibhav Garg, for suggesting resources for understanding concept, pointing out new topics and providing experienced guidance. Also, I would like to thank my parents for creating an environment and providing resources so that my studies do not get hampered.

Contents

Chapter 1

Introduction

Factorization of polynomials and root finding have been important questions to computer scientists and mathematicians. There has been extensive work on these fields since the late 20^{th} century. Factorization in finite fields has been achieved in [2, 3, 5, 14].

However rings create more problems as they are not unique factorization domains. In order to find a factorization we can Hensel's lifing. But this is not possible for every polynomial. Hensel's lifting and its obstructions has been presented in Chapter 3. In that chapter we also explain an algorithm due to [4, 9], and explained in greater detail in Chapter 5, which gives us a compact way to find and store exponentially many roots modulo powers of prime p of a polynomial in randomized polynomial time (polynomial in degree of f, k and $\log p$).

Next we describe Hilbert's Nullstellensatz in Chapter 4. In Chapter 5, we describe an algorithm to factorize polynomials in rings of the form $\mathbb{Z}/p^k\mathbb{Z}$ for $k \leq 4$ which was presented in [9]. Then we describe an algorithm to count all the roots, which can be extended to counting basic irreducible factors, that are basically those irreducible factors of f modulo p^k which remain irreducible $\mod p$ from [8]. Based on these we discuss some possible approaches and future work that can be done to obtain new results in Chapter 6.

Chapter 2

Preliminaries

We will mainly work in rings of the form $\mathbb{Z}/p^k\mathbb{Z}$, where p is a prime and $k \in \mathbb{N}$. Let $R(+, .)$ be a ring and S be a subset of R. We define the following notation for $a \in R$ as:

- $a.S = \{a.s | s \in S\}$
- $a + s = \{a + s | s \in S\}$

We will also use the notation $[n]$ to denote the set $\{1, 2, \dots n\}$.

Throughout this article, p will be considered to be a prime number unless specified.

2.1 Basic Algebraic Geometry

We are now going to describe some algebraic geometry terminologies and definitions. For a field k, we define *affine space* as follows:

$$\mathbb{A}_k^n = \{(c_1, c_2, \dots c_n) | c_i \in k\} \tag{2.1}$$

Now if $S \subset k[x_1, x_2, \dots x_n]$ be a collection of polynomials, the ideal generated by elements of S is called I_S. We define the *affine variety* as:

$$V(S) = \{(v_1, v_2, \dots v_n) \in \mathbb{A}_k^n | p(v_1, v_2, \dots v_n) = 0 \ \forall p \in S\} \tag{2.2}$$

It directly follows that $V(S) = V(I_S)$.

We also define an ideal $\mathcal{I}$ over a zero set $V \subseteq k^n$ as

$$\mathcal{I}(V) = \{f \in k[x_1, x_2, \dots x_n] | f(a) = 0 \ \forall a \in V\} \tag{2.3}$$

Now we define the projective space. Intuitively it is a space such that we include the line of intersection of two parallel lines (that do not intersect in the affine space). Again for a field k, a *projective space* is defined as

$$\mathbb{P}_k^n = \frac{k^{n+1} - \bar{0}}{\sim} \tag{2.4}$$

where $\sim$ is the equivalence relation defined by $\bar{a} = \bar{b}$ if and only if $\forall i \in [n]$, $a_i = \lambda b_i$ for some non-zero $\lambda \in k$. More related facts and definitions can be found in [13].

We also define *radical* of an ideal I, denoted as $\sqrt{I}$ given by:

$$\sqrt{I} = \{f \in k[x_1, x_2, \dots x_n] | \exists m \in \mathbb{N}; f^m \in I\} \tag{2.5}$$

It can be shown that radical of an ideal is also an ideal.

Now, for any two ideals $a, b \in k[x_1, x_2, \dots x_n]$, *Zariski Topology* states that $a \subseteq b \implies V(a) \supseteq V(b)$.

Lemma 2.1. *The following relations hold true:*

1. $V(\phi) = k^n$, $V(k[x_1, \dots x_n]) = \phi$

2. $V(ab) = V(a \cap b) = V(a) \cup V(b)$

3. $V(\sum_{i\in I} a_i) = \bigcap_{i\in I} V(a_i)$ *for a family of ideals* $(a_i)_{i\in I}$

Proofs of these can be found in [17].

2.2 Representatives

We define the notation $*$ to represent an entire ring R. For example, when we consider R of the form $\mathbb{Z}/p^k\mathbb{Z}$, for a prime p and a positive integer k, and write an element in the form $y = y_0 + y_1p + y_2p^2 + \dots y_mp^m + p^{m+1}*$ for $y_i \in \mathbb{Z}/p\mathbb{Z}\ \forall i \in \{0, 1, \dots m\}$, it refers to the set $S_y \subseteq R$ such that

$$S_y = \{y_0 + y_1p + \dots y_mp^m + z_{m+1}p^{m+1} + \dots z_{k-1}p^{k-1} | z_{m+1}, z_{m+2}, \dots z_{k-1} \in R/\langle p\rangle\} \tag{2.6}$$

This is basically a collection of (exponentially many) elements from $\mathbb{Z}/p^k\mathbb{Z}$ which has some fixed part denoted by the y_i's, and then all the elements from $\mathbb{Z}/p^{k-m-1}\mathbb{Z}$. We will sometimes write this as $y = \beta + p^{m+1}*$ where β is the fixed part. More explanation about representatives can be found in [9]. We will also denote this as the tuple $(\beta, m + 1)$.

Definition 2.2 (Representative Root)**.** For a polynomial $f(x) \in \mathbb{Z}/p^k\mathbb{Z}[x]$ and $r = \beta + p^i*$, for some natural number $i \leq k - 1$ and $\beta \in \mathbb{Z}/p^i\mathbb{Z}$, r is called a *representative root* of $f(x)$ if $\forall a \in r$, we have $f(a) \equiv 0 \mod p^k$. This also means that $\beta + p^iy$ is a root of $f(x) \mod p^k\ \forall y \in \mathbb{Z}/p^{k-i}\mathbb{Z}$

2.3 Split Ideals

First we define zero divisors in the ring of polynomials $R[x]$ where $R = \mathbb{Z}/p^k\mathbb{Z}$ to be the polynomials $f(x)$ such that $f(x) \equiv 0 \mod p$.

The concept zero set of a polynomial or a set of polynomials in the ring R is same as a variety. For $S \subseteq R[\bar{x}]$ we define the zero set as

$$\mathcal{Z}_R(S) = \{\bar{v} \in R^n | p(\bar{v}) \equiv 0 \mod p^k \ \forall p \in S\} \tag{2.7}$$

In Chapter 6, we will extensively use the concept of split ideals, to be defined in this section. The main work will be done with polynomials ideals of the form $I = \langle h_0(x_0), h_1(\bar{x}_1), h_2(\bar{x}_2), \dots h_l(\bar{x}_l)\rangle$ with each $h_i(\bar{x}_i) \in \mathbb{F}_p[\bar{x}_i]$ where $\bar{x}_i$ denotes the set of variables $\{x_0, x_1, \dots x_i\}$. We will also add another property to this ideal that $\forall i \in [l+1]$ if $\bar{a} \in \mathcal{Z}_{\mathbb{F}_p}\langle h_0(x_0), h_1(\bar{x}_1), h_2(\bar{x}_2), \dots h_{i-1}(\bar{x}_{i-1})\rangle$, then the polynomial $h_i(\bar{a}, x_i)$ splits completely into distinct linear factors.

Definition 2.3. Given a polynomial $f(x) \in R[x]$ and an ideal $I \subseteq \mathbb{F}[\bar{x}_l]$, we call I a *split ideal* wrt $f \mod p^k$ if

1. I is a triangular ideal of length $l+1$, i.e. $I = \langle h_0(x_0), h_1(\bar{x}_1), h_2(\bar{x}_2), \dots h_l(\bar{x}_l)\rangle$ for some $0 \le l \le k-1$, and $h_i(x_i) \in \mathbb{F}_p[\bar{x}_i]$ $\forall i \in \{0, 1, 2, \dots l\}$
2. $|\mathcal{Z}_{\mathbb{F}_p}(I)| = \Pi_{i=0}^{l} deg_{x_i}(h_i)$
3. for every $(a_0, a_1, \dots a_l) \in \mathcal{Z}_{\mathbb{F}_p}(I)$, we have $f(a_0 + a_1 p + \dots a_l p^l) \equiv 0 \mod p^{l+1}$

Also, the length of I is $l+1$ and its degree is $deg(I) = \Pi_{i=0}^{l} deg_{x_i}(h_i)$

Now note that the split ideal contains a notion of the roots of $f(x) \mod p^{l+1}$. Since the roots are present in the zero set of the triangular ideal, it is in some sense, a product of all the possible coordinates that can appear in the p-adic expansion of a root of $f(x)$ in $R/\langle p^{l+1}\rangle$.

Lemma 2.4 ([8])**.** *If $I_l = \langle h_0(x_0), h_1(\bar{x}_1), h_2(\bar{x}_2), \dots h_l(\bar{x}_l)\rangle$ is a split ideal in $\mathbb{F}_p[\bar{x}_l]$, then $I_j = \langle h_0(x_0), h_1(\bar{x}_1), h_2(\bar{x}_2), \dots h_j(\bar{x}_j)\rangle$ is also a split ideal in $\mathbb{F}_p[\bar{x}_j]$ for every $0 \le j \le l$.*

Lemma 2.5 ([8])**.** *A split ideal $I \subseteq \mathbb{F}_p[\bar{x}_l]$ can be decomposed as $I = \Cap_{\bar{a} \in \mathcal{Z}_{\mathbb{F}_p(I)}} I_{\bar{a}}$, where $I_{\bar{a}} = \langle (x - a_0), (x - a_1), \ldots (x - a_l) \rangle$ such that $\bar{a} = (a_0, a_1, \ldots a_l)$. Also by Chinese Remainder Theorem, we have $R/I = \bigoplus_{\bar{a} \in \mathcal{Z}_{\mathbb{F}_p}(I)} R/I_{\bar{a}}$.*

The proofs of these lemmas can be found in [8].

Chapter 3

Hensel's lifting and Representative roots

Hensel's lifting was given by Kurt Hensel to "lift" a factorization given modulo a prime ideal to modulo higher powers of that ideal. In our case we will only deal with the ideal being $\langle p \rangle_{\mathbb{Z}}$ and lifting to modulo p^k for integers $k \geq 1$. We will describe a few properties related to polynomials in rings of the form $\mathbb{Z}/p^k\mathbb{Z}$. For further reading, we refer the reader to [1].

Lemma 3.1. *A polynomial $f(x) \in \mathbb{Z}[x]$ can be uniquely written in the form $f(x) = a(x) + p.g(x)$ for a prime p, where $a(x) \in \mathbb{Z}/p\mathbb{Z}$.*

Proof. The proof directly follows from the fact that $f(x) = \sum_i c_i x^i = \sum_i ((c_i \mod p)x^i) + \sum_i (c_i - (c_i \mod p))x^i$ and $\mathbb{Z}/p\mathbb{Z}$ being an integral domain, implying uniqueness. □

Lemma 3.2. *A polynomial $f(x) \in \mathbb{Z}/p^k\mathbb{Z}[x]$ is an unit if and only if $f(x)$ can be written as $f(x) = a + p.g(x)$ where $a \in \mathbb{F}_p^\times$.*

Proof. If $f(x)$ is an unit, it can be written as $f(x) = a(x) + p.g(x)$ by Lemma 3.1. Now since this is an unit, there must be an inverse $f(x)^{-1}$ written in the form $a'(x)+pg'(x)$, product of which is one. By taking the product modulo p, we get that $a(x)a'(x) = 1$. Now since both a, a' are polynomials, there degree can not decrease after multiplication, and hence must be constants. So we get $a(x) \in \mathbb{F}_p^\times$.
For the converse, we find an inverse of $f(x) = a + p.g(x)$ for $a \in \mathbb{F}_p^\times$. Applying binomial theorem, we write $(a + p.g(x))^{-1} = a^{-1}(1 + p(a^{-1}g(x)))$. Now using the expansion $(1 + x)^{-1} = 1 - x + x^2 - \dots$, we get $f(x)^{-1} = a^{-1}(1 - p(a^{-1}g(x)) + p^2(a^{-1}g(x))^2 - \cdots + (-1)^{k-1}(a^{-1}g(x))^{k-1})$. It can be checked that $f(x)^{-1}f(x)$ is indeed 1 over $\mathbb{Z}/p^k\mathbb{Z}[x]$. □

Based on these we present Hensel's lemma, which gives us a technique to lift factorizations of certain kinds of polynomials from modulo p to modulo p^k.

Theorem 3.3. *Let $f, g, h \in \mathbb{Z}[x]$ be polynomials such that $f(x) \equiv g(x)h(x) \mod p$, and gcd$(g(x) \mod p, h(x) \mod p) = 1$ in $\mathbb{Z}/p\mathbb{Z}[x]$, then there exists polynomials (referred to as "lifts") $\tilde{g}, \tilde{h} \in \mathbb{Z}[x]$ such that $f(x) \equiv \tilde{g}(x)\tilde{h}(x) \mod p^k$ $\forall k \in \mathbb{N}$, and $\tilde{g} \equiv g \mod p$, $\tilde{h} \equiv h \mod p$.*

Proof. We give an algorithm to prove this, which has also been explained in [1].
Since gcd of g, h over $\mathbb{F}_p$ is one, $\exists \lambda, \mu \in \mathbb{F}_p[x]$ such that $\lambda g + \mu h \equiv 1 \mod p$. From this we iteratively construct a factorization of f modulo higher powers of p as follows.

The proof of correctness is by induction on i. At every step, each of the factors are

Algorithm 1 Hensel's Lifting

```
for $i \in 2, 3, \dots k$ do
    $q := \frac{f-gh}{p^{i-1}} \mod p$
    $u := q\mu$
    $v := q\lambda$
    $g := g + p^{i-1}u$
    $h := h + p^{i-1}v$
return $\tilde{g} = g, \tilde{h} = h$
```

congruent to the previous factor modulo p. Suppose after update at i^{th} step, g, h were g_{i-1}, h_{i-1}, and become updated to $\tilde{g}, \tilde{h}$ respectively. Then we have $f - \tilde{g}\tilde{g} \equiv f - (g_{i-1} + p^{i-1}u)(h_{i-1} + p^{i-1}v) \mod p^i$. From this, if we substitute the values of u, v and consider the fact that g_{i-1}, h_{i-1} are coprime modulo p (since $\lambda g + \mu h \equiv 1 \mod p$), we can show that this expression is zero modulo p^i. It can also be shown that $\tilde{g}, \tilde{h}$ are coprime over p, and the corresponding λ and μ can be found. For the complete proof, we refer the reader to [1]. □

Corollary 3.4. *Hensel's lifting is unique upto multiplication by units.*

However note that Hensel's lifting can not proceed if g, h has some non-trivial gcd modulo p. This basically means that $f(x) \mod p$ is a perfect power of some irreducible. [10] gives an analysis of the difficulties we face in this case. We now show a method from [11] which shows some of the conditions that need to be satisfied in order to lift.

Theorem 3.5 ([11])**.** *Let $f \equiv gh \mod p^k$ such that $f \equiv \phi^\ell \mod p$, where ϕ is an irreducible polynomial modulo p, and $e \leq \ell/2$ such that $g \equiv \phi^e \mod p$, $h \equiv \phi^{\ell-e} \mod p$. Then the following are equivalent:*

1. *$\frac{f-gh}{p^k} \in \mathbb{Z}[x]$ and divisible by g^e over modulo p*
2. *For every $\psi \in \mathbb{Z}[x]$ with $deg(\psi) < deg(g)$, there is a polynomial $\theta \in \mathbb{Z}[x]$ with $deg(\theta) < deg(h)$ such that $f \equiv (g + p^k\psi)(h + p^k\theta) \mod p^{k+1}$*
3. *There exist polynomials $\psi, \theta \in \mathbb{Z}[x]$, with $deg(\psi) < deg(g)$, $deg(\theta) < deg(h)$ such that $f \equiv (g + p^k\psi)(h + p^k\theta) \mod p^{k+1}$*
4. *There exist polynomials $\psi, \theta \in \mathbb{Z}[x]$ with $f \equiv (g + p^k\psi)(h + p^k\theta) \mod p^{k+1}$*

Proof. $(i) \implies (ii)$ Let $\frac{f-gh}{p^k} \equiv \phi^e\alpha \mod p$ for some $\alpha \in \mathbb{Z}[x]$, and $\psi, \theta \in \mathbb{Z}[x]$, with $deg(\psi) < deg(g)$. Let $\theta \equiv \alpha - g^{\ell-2e}\psi \mod p$. Then, using the fact that $e \leq \ell/2$, we can show that $f - (g + p^k\psi)(h + p^k\theta) \equiv 0 \mod p^{k+1}$.

$(ii) \implies (iii) \implies (iv)$ is directly follows. Now, we are required to show $(iv) \implies (i)$. Let $\psi, \theta \in \mathbb{Z}[x]$ with $f \equiv (g + p^k\psi)(h + p^k\theta) \mod p^{k+1}$. Then we have

$$\frac{f - gh}{p^k} \equiv \psi g + \theta h \equiv \psi g^{\ell-e} + \theta g^e \equiv g^2(\psi g^{\ell-2e} + \theta) \mod p$$

This proves the theorem. □

From these we see that if the polynomial to be factored is not a power of some irreducible modulo p, then we can lift it to modulo any p^k. There will be unique factors (unique upto multiplication by units) and there is an one-one correspondence of roots modulo p with roots modulo p^k. However the more difficult case is left, when we have $f \equiv \phi^\ell \mod p$ for some polynomial $\phi(x)$ irreducible modulo p. There have been some attempts to factorize polynomials of this form [19, 9], the best being [9], which achieved factorization up to modulo p^4. We will describe this in greater details in Chapter 5.

We had defined representatives in Section 2.2 which represents exponentially many elements in $\mathbb{Z}/p^k\mathbb{Z}$ in polynomial space (polynomial in $k, \log p$). Similarly representative root was defined in Definition 2.2. Motivation for defining roots of a polynomial in this way was from the fact that a polynomial might have exponentially many roots in a ring. For example $x^2 + px \mod p^2$ has roots which are all the multiples of p, i.e. $2^{\log p}$-many roots! We can also write this as the representative root $p*$. This was first introduced in [18].

Theorem 3.6 ([4]). *A polynomial $f(x)$ of degree d has at most d-many representative roots modulo a prime power p^k.*

Based on this result, [9] developed an algorithm to find all the roots of a polynomial modulo any prime power in randomized polynomial time.

Algorithm 2 Find roots modulo p^k

```
procedure ROOT-FIND($g(x), p^i$)
    if $g(x) \equiv 0 \mod p^i$ OR $i \leq 0$ then return $\{*\}$
    $g(x) \equiv p^{\alpha}\tilde{g}(x) \mod p^i$ for $\alpha \in \mathbb{N}, \tilde{g} \in \mathbb{Z}[x]$ such that $\tilde{g}(x) \not\equiv 0 \mod p$
    $R =$ roots of $\tilde{g}(x) \mod p$ using Cantor Zassenhaus algorithm
    if $R == \phi$ then return $\{\}$ (Dead-end)
    $S := \phi$
    for each root $a \in R$ do
        $\tilde{g}_a(x) := \tilde{g}(a + px)$
        $T =$ROOT-FIND($\tilde{g}_a(x), p^{i-\alpha}$)
        $S = S \cup (a + pT)$
    return $S$
```

This algorithm can be further extended to finding roots of a polynomial $f(y) \in \mathbb{Z}[x]/\langle p, \phi^k \rangle$ for some polynomial ϕ^k, irreducible modulo p.

Chapter 4

Hilbert's Nullstellensatz

Hilbert's Nullstellensatz is a theorem that links geometry with algebra. Nullstellensatz in German means "theorem of zeroes", and this theorem establishes a connection between existence of zeroes of a system of polynomials with ideals of polynomials in algebraically closed fields. This is a computational problem to determine efficiently if Hilbert's Nullstellensatz certificates can be found (and hence decide if a system of polynomials has a common zero). For more details we refer the reader to [13]. Throughout this chapter, we will denote $\mathbb{K}$ as an algebraically closed field.

In mathematics, a fundamental question is the *Consistency Question*. Given a set of polynomials $f_1, f_2, \dots f_m \in \mathbb{K}[x_1, x_2, \dots x_n]$ and the ideal I generated by them, the consistency question asks if $V(I) = \phi$. For the decidability of this question, Weak Hilbert's Nullstellensatz (WHN) gives a certificate for this. The theorem can be stated as follows:

Theorem 4.1 (WHN). *For an ideal* $I \subseteq \mathbb{K}[x_1, x_2, \dots x_n]$, $V(I) = \phi \iff 1 \in I$.

Theorem 4.1 also means the existence of polynomials $g_1, g_2, \dots g_m \in \mathbb{K}[x_1, x_2, \dots x_n]$ such that $f_1 g_1 + f_2 g_2 + \dots f_m g_m = 1$. These polynomials g_i's are referred to as

Nullstellensatz certificates. In order to prove Theorem 4.1 we state some more theorems and lemmas. The next theorem is called Strong Hilbert's Nullstellensatz (SHN).

Theorem 4.2 (SHN)**.** *For every ideal $I \in \mathbb{K}[x_1, x_2, \dots x_n]$, $\sqrt{I} = I(V(I))$.*

Another theorem required is the Extension Theorem, stated as follows

Theorem 4.3 (Extension Theorem)**.** *Let $I = \langle f_1, f_2, \dots f_m \rangle \subset \mathbb{K}[x_1, x_2, \dots x_n]$ such that $\exists i$ with f_i having highest degree term wrt x_n as a non-zero constant in $\mathbb{K}$, and $J = I \cap \mathbb{K}[x_1, x_2, \dots x_{n-1}]$. If $(a_1, a_2, \dots a_{n-1}) \in V(J)$ then $\exists a_n \in \mathbb{K}$ such that $(a_1, a_2, \dots a_n) \in V(I)$.*

Proof. Proof of Theorem 4.3 requires the idea of resultants. Basically, when we have two polynomials $f, g \in (F[x_1, x_2, \dots x_{n-1}])[x_n]$, for a field F, we can write the linear equation $af + bg$ for $a, b \in (F[x_1, x_2, \dots x_{n-1}])[x_n]$ for $deg_{x_n}(a) < deg_{x_n}(g)$ and $deg_{x_n}(b) < deg_{x_n}(f)$. Now considering the coefficients of powers of x_n in a, b (which are in $F[x_1, x_2, \dots x_{n-1}]$) and writing them as a column vector A, we can find a matrix (from linear equations corresponding to each power of x_n), S, such that $af + bg = SA$. This matrix S is called the Sylvester matrix and resultant of f, g wrt x_n is defined as $Res_{x_n}(f, g) = det(S)$.

From the definition, it is clear that $Res_{x_n} \in (f, g)$. We also have the fact that $Res_n(f, g) = 0$ if and only if $gcd_{x_2}(f, g)$ is not trivial.

Also, if $\bar{a} = (a_1, \dots a_{n-1}) \in F^{n-1}$ then $Res_{x_n}(f, g) = l^{cd(g)} Res_{x_n}(f(\bar{a}, x_n), g(\bar{a}, x_n))$, where $cd(g)$ is the degree drop of g, given by $deg_{x_n}(g(x_1, \dots x_n)) - deg_{x_n}(g(\bar{a}, x_n))$.

More properties of resultants are available at [13]. With this notion of resultant established, we give the proof of Theorem 4.3, which is due to [7]. Denote $\bar{a} = (a_1, a_2, \dots a_{n-1}) \in V(J)$ and consider the homomorphism $\mathbb{K}[x_1, x_2, \dots x_n] \to \mathbb{K}[x_n]$ given by $f(x_1, x_2, \dots x_n) \mapsto f(\bar{a}, x_n)$. Let $I' = \{f(\bar{a}, x_n | f \in I)\} \subseteq \mathbb{K}[x_n]$. Now, since I' is a PID, $\exists f'$ such that $I' = \langle f' \rangle$. Here f can either be a constant or a non-constant polynomial.

If f is not a constant, then $\exists a_n \in \mathbb{K}$ such that $f'(a_n) = 0$ (since $\mathbb{K}$ is algebraically closed). From this we get $(\bar{a}, a_n) \in V(I)$. So it we choose any $\bar{a}$ from $V(J)$ and using the homomorphism and then considering the corresponding I', we can choose an a_n as required.

Now, let f is a constant, say $b \in \mathbb{K}$. We are given that f_i has leading coefficient wrt x_n as $c \in \mathbb{K}$. Also there must be a polynomial $f'' \in I$ such that $f''(\bar{a}, x_n) = f'(x_n) = b$. Let $r(x_1, x_2, \dots x_{n-1}) = Res_{x_n}(f_i, f'')$. Now since resultant is contained in the ideal generated by the two polynomials, we have $r \in J$. Hence $r(\bar{a}) = 0$ as $\bar{a} \in V(J)$. We prove using properties of resultants, that this can not be true. We indeed have $r(a) = c^{deg_{x_n}(f')}(f_i(\bar{a}, x_n), b)$ as degree drop of f' is $deg_{x_n}(f')$. Now $f_i(\bar{a}, x_n), b)$ is non-zero as a constant can not have a non-trivial gcd with a polynomial over a field. Hence this is a contradiction and f' can not be constant. □

Proof of WHN. Note that one direction given by $1 \in I \implies V(I) = \phi$ is trivial. Since $1 \in I$, $\exists g_1, g_2, \dots g_m \in \mathbb{K}[x_1, \dots x_n]$, $g_1 f_1 + \cdots + g_m f_m = 1$. Now, if $V(I)$ is not empty, $\exists \bar{a} \in V(I)$ such that $f_i(\bar{a}) = 0 \; \forall i \in [m]$.

For the other direction, i.e. $V(I) = \phi \implies 1 \in I$, we prove by induction on the number of variables, n.

For base case $n = 1$, it follows from the fact that an ideal formed from univariate polynomials is a PID generated by their gcd, and since they do not have any common factors, the gcd must be trivial. Hence $1 \in I$.

Now suppose this implication holds for $n-1$ variables. If any of the f_i's are constant then we are done. So we can assume that they are non-constant with degree in x_n of f_i being d_i. We now want to exploit Theorem 4.3. Choose some $z_1, z_2, \dots z_{n-1} \in \mathbb{K}$

and apply the linear transformation

$$x_n = y_n$$

$$x_{n-1} = y_{n-1} + z_{n-1}y_n$$

$$\vdots$$

$$x_1 = y_1 + z_1 y_n$$

So $f_i(x_1, \dots x_n)$ can be written as $g_i(z_1, z_2, \dots z_{n-1})y_n^{d_i}$+(lower degree terms in y_n) $=$ $f_i'(y_1, y_2, \dots y_n)$, where z_j's are seen as constants. Now, we can choose some $(z_1, \dots z_n)$ such that $g_i(z_1, z_2, \dots z_n)$ is non-zero. Notice that if $1 \in I$ then $1 \in \langle f_1', \dots f_m' \rangle$, and vice versa, as well as we are just taking a linear transformation of the coordinates. Suppose $I' = \langle f_1', \dots f_m' \rangle$ and $J = \mathbb{K}[y_1, \dots y_{n-1}] \cap I'$. We also have $V(I) = \phi \implies V(I') = \phi$, and $V(I') = \phi \implies V(J) = \phi$, as if $V(J)$ as not non-empty, then neither would $V(I)'$ be, by Extension Theorem. Hence by induction hypothesis, $1 \in J$, and since $J \subset I \implies 1 \in I' \implies 1 \in I$.

This completes the proof of WHN. □

Now in order to prove SHN, we show that SHN and WHN are same. This part of the proof is from [20].

First note that $\sqrt{I} \subseteq I(V(I))$ is trivial as, if some polynomial f has a root, then f^m has the same root as well.

Lemma 4.4 (SHN $\implies$ WHN)**.** *For an ideal I, $\mathcal{I}(V(I)) \subseteq \sqrt{I}, V(I) = \phi \implies 1 \in I$.*

Proof. We have $V(I) = \phi$ and $\mathcal{I}(\phi) = K[x_1, \dots x_n]$. Now by SHN, $\mathcal{I}(V(I)) \subseteq \sqrt{I} \implies 1 \in \sqrt{I}$. Now $\exists d \in \mathbb{N}$ such that $1^d \in I$, i.e. $1 \in I$. □

Lemma 4.5 (WHN $\implies$ SHN)**.** *For an ideal I, $(V(I) = \phi \implies 1 \in I) \implies \mathcal{I}(V(I)) \subseteq \sqrt{I}$.*

Proof. First we take an arbitrary polynomial $f \in \mathcal{I}(V(I))$ and proceed to show that $f \in \sqrt{I}$, by showing the existence of $d \in \mathbb{N}$ such that $f^d = \sum_{i\in[m]} q_i f_i$.
Consider $g = 1 - yf(x_1, x_2, \dots x_n)$ and look at the ideal $I' = \langle f_1, \dots f_m, g\rangle \subseteq \mathbb{K}[x_1, \dots x_n, y]$. Notice that whenever f is zero, g is non-zero, and since $V(I) \subseteq V(\sqrt{I})$ (from Zariski Topology) and g' is non-zero whenever all of f_i's are zero, we have $V(I') = \phi$. Now WHN implies $1 \in I'$. So $\exists q'_1, \dots q'_m, q \in \mathbb{K}[x_1, x_2, \dots x_n, y]$, such that $\sum_{i\in[m]} q'_i f_i + qg = 1$. Using this identity over $\mathbb{K}(x_1, \dots x_n)[y]$, we substitute $y = \frac{1}{f(x_1,\dots x_n)}$ (for which g is zero) and hence get $1 = \sum_{i\in[m]} q'_i(x_1, \dots x_m, \frac{1}{f(x_1,\dots x_n)})f_i(x_1)$. Now, if $D = \max\{deg_y(q_i(x_1, \dots x_n, y))|i \in [m]\}$, we can multiply both sides by f^D to get $f^D = \sum_{i\in[m]}(q'_i(x_1, \dots x_n, \frac{1}{f})f^D f_i)$ where each of $q_i(x_1, \dots x_m, \frac{1}{f})f^D \in \mathbb{K}[x_1, \dots x_n]$. This implies $\forall f \in \mathcal{I}(V(I))$, $f \in \sqrt{I}$, and hence $\mathcal{I}(V(I)) \subseteq \sqrt{I}$. Both this and $\sqrt{I} \subseteq \mathcal{I}(V(I))$ prove SHN. □

Based on these results, the main question in computer science and computational mathematics is to find these Nullstellensatz certificates. Another question is the ideal membership problem, which asks if a given polynomial is present in the ideal generated by a set of polynomials. Hilbert's Nullstellensatz is a special case of this which asks if 1 is present in the ideal. A way to solve for HN certificates can be found in [6] where they gave an algorithm based on solving simultaneous linear equations. [15] proved that HN is in polynomial hierarchy, and under Generalized Reimann Hypothesis, it is actually in AM; while [12] proved that HN is in $AM \cap coAM$ under GRH.

Chapter 5

Factorization of polynomials modulo p^4

Polynomial factorization in fields has been studied extensively by computer scientists and mathematicians. Some of the randomized poly-time methods for factorization of polynomials in fields are [16] over rationals, [2], [3], [5], [14] over finite fields etc. However since rings are not unique factorization domains, polynomial factorization is relatively difficult. In this chapter, we present the techniques of polynomial factorization in rings of the form $\mathbb{Z}/p^k\mathbb{Z}$ for $k \leq 4$ given by [9]. Previously [19] had given an algorithm for factorization unto $k = 3$, but [9] has the best results until now.

5.1 Main Idea

The main theorem of this chapter is:

Theorem 5.1 ([9]). *Let p be a prime and $k \leq 4$ be an integer. Given a polynomial $f(x) \in \mathbb{Z}$, we can factorize $f(x) \mod p^k$ in randomized poly(deg$(f), \log p$)-time.*

We give an algorithm due to [9] to find a factorization and prove its correctness to show that Theorem 5.1 holds true. Note that if f is not a power of an irreducible modulo p then we can find two factors such that they do not have any non-trivial gcd. This implies that for every factorization $\mod p$, there exists at most one unique lift. So we deal with only the case of f being a power of an irreducible modulo p. This can be seen as a generalization of Hensel's lemma upto modulo p^4.

We assume that $f(x) \equiv \phi(x)^\ell \mod p$ for some $\phi(x) \in \mathbb{Z}[x]$ such that $\phi \mod p$ is irreducible. Note that a factor of $f(x) \mod p^k$ will be of the form $\phi^a - py \mod p^k$ for some $y \in (\mathbb{Z}/p^k\mathbb{Z})[x]$. With this observation, we intend to reduce factorization to that of root finding of some polynomial $E(y) \in (\mathbb{Z}[x])[y]$ to find the value of y and hence the factor $\phi^a - py$. We will later prove that this root finding need to be done in a local ring of the form $\mathbb{Z}/\langle p^k, \phi^{ak}\rangle$. The method of obtaining such an $E(y)$ has been inspired by binomial theorem and the fact that $(1-x)^{-1} = 1 + x + x^2 + \dots$, which is quite like Lemma 3.2. We consider the expansion of $f(x)/(\phi^a - py)$ and while considering $\mod p^k$, we want it to be divisible by ϕ^{ak}.

In this problem we will consider the value of y in a "sort of" p-adic expansion, decomposing the root of $E(y) \mod \langle p^k, \phi^{ak}\rangle$ into coordinates $y_0, y_1, \dots y_{k-1} \in \mathbb{F}_p[x]/\langle \phi^{ak}\rangle$ such that $y = y_0 + y_1 p + \dots y_{k-1}p^{k-1} \mod \langle p^k, \phi^{ak}\rangle$. We will later show that the root does not depend on the last two coordinates and for the case of $k = 4$, we can write $E(y)$ as $E'(y_0, y_1)$ in the ring $\mathbb{F}_p[x]/\langle \phi^{4a}\rangle$ and the roots can be found from a modification of Algorithm 2.

We also write each y_i as $y_i = y_{i,0} + y_{i,1}\phi + \dots y_{i,4a-1}\phi^{4a-1}$ and write $E'(y_0, y_1)$ as the sum of two univariates to apply root finding. This method of decomposing to coordinates is a very strong one and the tool will be used to cound the number of roots and basic irreducible factors in Chapter 6 as well.

5.2 Towards the Algorithm

We want to establish an algorithm to factorize a polynomial $f(x) \mod p^k$ of degree d. It has been shown how we can assume $f(x) \equiv \phi^\ell \mod p$ for some irreducible polynomial ϕ such that $\ell deg(\phi) \leq deg(f)$ without any loss of generality. So $f(x)$ is of the form $\phi^\ell + pg \mod p^k$ for some $g(x) \in \mathbb{Z}[x]$.

First we reduce the problem of factorization to that of root finding of some other polynmial depending on f in a local ring.

5.2.1 Reduction to root-finding

We want to find a factor h of $f \equiv \phi^\ell + pg \mod p^k$ where $h \equiv \phi^a - py$ for $a < \ell$ and $y \in (\mathbb{Z}/p^k\mathbb{Z})[x]$.

We will denote the ring $\mathbb{Z}[x]/\langle p^k, \phi^{ak}\rangle$ by R and $\mathbb{F}_p[x]/\langle \phi^{ak}\rangle$ as R_0. Consider the polynomial $E(y) \in R[y]$

$$E(y) = f(x)(\phi^{a(k-1)} + \phi^{a(k-2)}(py) + \cdots + \phi^a(py)^{k-2} + (py)^{k-1}) \tag{5.1}$$

Using $E(y)$, we reduce factoring $f \mod p^k$ to root finding as given by the following theorem.

Theorem 5.2 ([9])**.** *Let $f(x), h(x) \in \mathbb{Z}[x]$ such that $f \equiv \phi^\ell + pg \mod p^k$ and $h(x) \equiv \phi^a - py \mod p^k$, where $y, g \in (\mathbb{Z}/p^k\mathbb{Z})[x]$ and $a \leq \ell$, then $h(x)$ divides $f(x)$ modulo p^k if and only if*

$$E(y) \equiv 0 \mod \langle p^k, \phi^{ak}\rangle \tag{5.2}$$

Proof. We denote Q as the ring of fractions of $(\mathbb{Z}/p^k\mathbb{Z})[x]$. Now since ϕ is not a zero divisor (as otherwise we could just consider $f(x)$ as $f(x)/p$ until ϕ is not a zero divisor), we have $E(y)/\phi^{ak} \in Q$. We prove this theorem starting from the reverse direction.

Indeed if $E(y) \equiv 0 \mod \langle p^k, \phi^{ak} \rangle$, then $E(y)/\phi^{ak}$ is a valid polynomial over $(\mathbb{Z}/p^k\mathbb{Z})[x]$. Multiplying $E(y)/\phi^{ak} \mod p^k$ with $h \equiv \phi^a - py \mod p^k$, we get

$$(\phi^a - py)E(y)/\phi^{ak} \equiv f(x)/\phi^{ak}(\phi^a - pu)(\sum_{i=0}^{k-1} \phi^{a(k-i-1)}(py)^i) \mod p^k$$

$$\equiv (f/\phi^{ak})(\phi^{ak} - (py)^k) \equiv f \mod p^k$$

This implies that $h(x)$ divides $f(x)$ modulo p^k.

Now for the other direction. Suppose $f(x) \equiv h(x)h_0(x) \mod p^k$ for some polynomial $h_0(x) \in \mathbb{Z}[x]$. We also have, from the proof of the reverse direction $f(x) \equiv (E(y)/\phi^{ak})h(x)$. Subtracting the equations of these two factorizations, we get

$$h(x)(g(x) - E(y)/\phi^{ak}) \equiv 0 \mod p^k$$

Now since $h(x)$ is not a zero divisor, we have $E(y)/\phi^{ak} = g(x)$ in Q. Now since $g(x) \in (\mathbb{Z}/p^k\mathbb{Z})[x]$, we have $E(y) \equiv 0 \mod \langle p^k, \phi^{ak} \rangle$. □

Now notice that we can consider $a \leq \ell/2$ as otherwise we will be considering the other factor which is $f(x)/h(x) \mod p^k$, since $h(x)$ is not a zero divisor. Also, when we consider $y = y_0 + y_1 p + y_2 p^2 + \ldots y_{k-1}p^{k-1}$ for each $y_i \in R_0$, we can neglect the last two coordinates. This follows from the following lemma.

Lemma 5.3 ([9]). *If $y = y_0 + y_1 p + y_2 p^2 + \ldots y_{k-1}p^{k-1}$ is a root of $E(y)$ in R, then all the elements of $y = y_0 + y_1 p + y_2 p^2 + \ldots y_{k-3}p^{k-3} + p^{k-2}*$ are also roots of $E(y)$.*

Proof. Notice that in the expansion of $E(y)$ all the y's that are present are multiplied by p, which implies y_{k-1} will have coefficient divisible by p^k, which is 0 in R. Also for y_{k-2}, all the terms of the form $(py)^i$ for $i \geq 2$ do not contribute as they are zero in R. The only remaining term is $f(x)\phi^{a(k-2)}py$. Now $f(x)\phi^{a(k-2)} \equiv \phi^{e+ak-2a} \mod \langle p, \phi^{ak}\rangle$ which also vanishes modulo $\langle p^k, \phi^{ak}\rangle$, as $e \geq 2a$ as we have assumed. □

From now on we consider $k = 4$ and analyze the problem of factorization as done by [9]. We have

$$E(y) = f(x)(\phi^{3a} + \phi^{2a}(py) + \phi^{a}(py)^2 + (py)^3) \mod \langle p^4, \phi^{4a}\rangle \tag{5.3}$$

From Lemma 5.3, we use $y = y_0 + py_1$ and use the equation

$$f \times (\phi^{3a} + \phi^{2a}p(y_0 + py_1) + \phi^{a}p^2(y_0^2 + 2py_0y_1) + (py)^3) \equiv 0 \mod \langle p^4, \phi^{4a}\rangle \tag{5.4}$$

From this, our main idea is to first solve this modulo $\langle p^3, \phi^{4a}$. Note that since $f \equiv \phi^{\ell} \mod p$, we can say that considering modulo $\langle p^3, \phi^{4a}\rangle$, the variable y_1 is redundant. The following lemma gives us a way to find the representative roots in this ring, which basically reduces the root finding to characteristic p, and from [9], we are able to find roots in rings of the form R_0.

Lemma 5.4 ([9])**.** *We can efficiently find a unique set S_0 of representative pairs (a_0, i_0), $a_0 \in R_0, i_0 \in \mathbb{N}$ such that*

$$E((a_0 + \phi^{i_0}y_0) + py_1) = p^3E'(y_0, y_1) \mod \langle p^4, \phi^{4a}\rangle$$

for $E'(y_0, y_1) \in R_0[y_0, y_1]$ depending on the representative root pair. Also we will have:

1. *$|S_0| \leq 2$. If the algorithm fails to find any such E' then $E(y) \equiv 0 \mod \langle p^4, \phi^{4a}\rangle$ has no solutions*

2. *$E'(y_0, y_1) = E_1(y_0) + E_2(y_0)y_1$ where $E_1(y_0), E_2(y_0) \in R_0[y_0]$, E_1 is a cubic in y_0 and E_2 is a linear in y_0*

3. *For every root $y \in R$ of $E(y) \equiv 0 \mod \langle p^4, \phi^{4a} \rangle$, $\exists (a_0, i_0) \in S_0$, and $(a_1, a_2) \in R \times R$ such that $y = a_0 + \phi^{i_0} a_1 + p a_2$ and $E'(a_1, a_2) \equiv 0 \mod \langle p, \phi^{4a} \rangle$*

Proof. We first look at the equation $E(y) \equiv 0 \langle p^4, \phi^{4a} \rangle$ modulo p^2. Hence

$$f\phi^{2a}(\phi^a + py_0) \equiv 0 mod \langle p^2, \phi^{4a} \rangle$$

Substituting $f = \phi^\ell + pg$ we get $pg\phi^3 a \equiv 0 \mod \langle p^2, \phi^{4a} \rangle$ implying

$$g \equiv 0 \mod \langle p, \phi^a \rangle$$

which is a necessary condition for y_0 to exist.
Also from this we get that $g_1 = pg_1 + \phi^a g_2$.

We consider modulo p^3 and get $f(\phi^3 a + \phi^{2a} p y_0 + \phi^a p^2 y_0^2) \equiv 0 \mod \langle p^3, \phi^{4a} \rangle$. Now substituting the value of $f = \phi^e + pg$ and the value of $g = pg_1 + \phi^a g_2$ we get

$$p^2(\phi^{e+a} y_0^2 + \phi^{3a} g_2 y_0 + \phi^{3a} g_1) \equiv 0 \mod \langle p^3, \phi^{4a} \rangle$$

Now we can divide this equation by $p^2\phi^{3a}$ (since $e \geq 2a$) we get an equation modulo $\langle p, \phi^a \rangle$ which is a quadratic in y_0 and its roots can be found using root-find algorithm with the modification due to [9]. So S_0 has atmost 2 representative roots according to theorem 5 of the form (a_0, i_0). So for every $y \in a_0 + \phi^{i_0} * + p*$ satisfies

$$E(y) \equiv 0 \mod \langle p^3, \phi^{4a} \rangle$$

Substituting $y = a_0 + \phi^{i_0} y_0 + p y_1$ we have

$$E(a_0 + \phi^{i_0} y_0 + p y_1) \equiv p^3 E'(y_0, y_1) \mod \langle p^4, \phi^{4a} \rangle$$

Substituting this value of y in $E(y)$ we can find E_1 and E_2 as well which are cubic and linear respectively. □

With $E'(y_0, y_1)$ established as above, we now move on to finding its roots modulo $\langle p, \phi^{4a} \rangle$.

5.2.2 Root finding of $E'(y_0, y_1)$ modulo $\langle p, \phi^{4a} \rangle$

We already have $E'(y_0, y_1)$ of the form $E_1(y_0) + E_2(y_0)y_1$ where E_2 is linear in y_0 and E_1 is a cubic. Pertaining to the normal definition of valuations, we define valuation wrt a polynomial ϕ as $v_\phi(u) = r$ is r is the largest integer such that $\phi^r | u$ over modulo p. The remaining part of the strategy is to go over all possible valuations $0 \leq r \leq 4a$ and find y_0 such that $E_2(y_0)$ has valuation wrt ϕ exactly r, but $E_1(y_0)$ has valuation greater than or equal to r. From this y_1 can be obtained by dividing $E_1(y_0)$ by $E_2(y_0)$. From this we will have $y_1 \equiv -(E_1(y_0)/\phi^r)/(E_2(y_0)/\phi^r) \mod \langle p, \phi^{4a-r} \rangle$. This gives rise to the following lemma. Note that if $r = 4a$, we take y_1 to be $*$.

Lemma 5.5 ([9]). *A tuple $(y_0, y_1) \in R_0 \times R_0$ satisfies $E_1(y_0) + y_1 E_2(y_0) \equiv 0 \mod \langle p, \phi^{4a} \rangle$ if and only if $v_\phi(E_1(y_0)) \geq v_\phi(E_2(y_0))$.*

Finally we prove another lemma which will be used in filtering out the "bad" y_0's for which valuation of $E_2(y_0)$ is more than r.

Lemma 5.6. *][[9]] Given $E_2(y_0) \in R_0[y_0]$, which is a linear polynomial, and for some $0 < r \leq 4a - 1$, let (b, i) be a representative root modulo $\langle p, \phi^r \rangle$, consider the quotient $E_2'(y_0) = E_2(b + \phi^i y_0)/\phi^r$.*

If $E_2'(y_0)$ is not identically zero modulo $\langle p, \phi \rangle$, then there exists at most one $\theta \in R_0/\langle \phi \rangle$ such that $E_2'(\theta) \equiv 0 \mod \langle p, \phi \rangle$, and this θ can be efficiently found.

Proof. We can write $E_2(y_0)$ as $u+vy_0 \mod \langle p, \phi^r\rangle$. Since y_0 can take any value, we have $v_\phi(u), v_\phi(v) \geq r$. Now, in the three cases as follows, we can find θ as:

1. $\mathrm{val}_\phi(u) \geq r$ and $\mathrm{val}_\phi(v) = r$, then $E_2(\theta) \equiv 0 \mod \langle p, \phi\rangle$ only when $\theta = \frac{-(u/\phi^r)}{(v/\phi^r)}$ mod $\langle p, \phi\rangle$
2. $\mathrm{val}_\phi(u) = r$ and $\mathrm{val}_\phi(v) > r$, $E_2(\theta)$ will never be zero modulo $\langle p, \phi\rangle$ for any θ in $R_0/\langle\phi\rangle$
3. $\mathrm{val}_\phi(u) > r$ and $\mathrm{val}_\phi(v) > r$, then consider some $r' > r$ and do the same.

□

Using these results we give the algorithm to factorize a polynomial modulo p^4 in the next section.

5.3 The Algorithm

The following algorithm correctly finds a factor of $f(x) \mod p^k$ where $f \equiv \phi^\ell$ mod p for some irreducible polynomial ϕ.

Theorem 5.7 ([9]). *The output of the algorithm 3 (the set $Z - Z'$) contains only those $y_0 \in R_0$ such that there is some $y_1 \in R_0$ with $y = y_0 + y_1 p$ as a root of $E(y)$ in R. We can do this computation in randomized poly(deg$(f), \log p$) time. Thus we can find all the roots $y = y_0 + y_1 p + y_2 p^2$ of $E(y)$ in R where $y_2 = *$ in R.*

Proof. This algorithm basically outputs the roots of $E(y) = f(x)(\phi^{3a} + \phi^{2a}(py) + \phi^a(py)^2 + (py)^3) \mod \langle p^4, \phi^{4a}\rangle$ where $y = y_0 + y_1 p + y_2 p^2$ with $y_i \in R_0$.
Using Lemma 5.4, the algorithm fixes some y_0 from the set S_0 and attempts to find roots of $E'(y_0, y_1) \mod \langle p, \phi^{4a}\rangle$. This lets us count all the roots y_0's as well for which

Algorithm 3 Factorization modulo p^4

```
1: Given $E(y_0 + py_1)$, let $S_0$ be set of all representative pairs $(a_0, i_0)$ such that $p^3 | E((a_0 + \phi^{i_0} y_0) + py_1) \mod \langle p^4, \phi^{4a} \rangle$
2: Initialize sets $Z = \phi$, $Z' = \phi$, seen as subsets of $R_0$
3: for each $(a_0, i_0) \in S_0$ do
4:     Substitute $y_0 \mapsto a_0 + \phi^{i_0} y_0$, let $E'(y_0, y_1) = E_1(y_0) + y_1 E_2(y_0) \mod \langle p, \phi^{4a} \rangle$ be as defined before
5:     if $E_2(y_0) \not\equiv 0 \mod \langle p, \phi \rangle$ then find $\theta$ as in Lemma 5.6 such that $E_2(\theta) \equiv 0 \mod \langle p, \phi \rangle$. Update $Z \leftarrow Z \cup (a_0 + \phi^{i_0} *)$ and $Z' \leftarrow Z' \cup (a_0 + \phi^{i_0}(\theta + \phi *))$
6:     for $r \in [4a]$ do
7:         Initialize sets $Z_r = \phi$ and $Z'_r = \phi$
8:         Call modified root-find algorithm on $E_1, \phi^r$ to get set $S_1$ of rep. pairs $(a_1, i_1)$ such that $E_1(a_1 + \phi^{i_1} y_0) \equiv 0 \mod \langle p, \phi^r \rangle$
9:         for each $(a_1, i_1) \in S_1$ do
10:            Consider $E'_2(y_0) = E_2(a_1 + \phi^{i_1} y_0) \mod \langle p, \phi^{4a} \rangle$
11:            Call modified root-find algorithm on $E'_2, \phi^r$ to get rep. pair $(a_2, i_2)$ such that $E'_2(a_2 + \phi^{i_2} y_0) \equiv 0 \mod \langle p, \phi^r \rangle$
12:            if $r = 4a$ then
13:                Update $Z_r \leftarrow Z_r \cup (a_1 + \phi^{i_1}(a_2 + \phi^{i_2} *))$ and $Z'_r \leftarrow Z'_r \cup \{\}$
14:            else if $E'_2(a_2 + \phi^{i_2} y_0) \not\equiv 0 \mod \langle p, \phi^{r+1} \rangle$ then
15:                Get $\theta$, if exists, such that $E'_2(a_2 + \phi^{i_2}(\theta + \phi y_0)) \equiv 0 \mod \langle p, \phi^{r+1} \rangle$. Update $Z'_r \leftarrow Z'_r \cup (a_1 + \phi^{i_1}(a_2 + \phi^{i_2}(\theta + \phi *)))$
16:                Update $Z_r \leftarrow Z_r \cup (a_1 + \phi^{i_1}(a_2 + \phi^{i_2} *))$
17:        $Z \leftarrow Z \cup (a_0 + \phi^{i_0} Z_r)$ and $Z' \leftarrow Z' \cup (a_0 + \phi^{i_0} Z'_r)$
18: return $Z, Z'$
```

y_1 exists. In this way we find all the solutions of $E'(y_0, y_1)$ by looping over all the possible valuations wrt ϕ of $E_2(y_0)$. Lemma 5.5 shows why doing this is sufficient.

Next we consider all the representative roots $b + \phi^i *$ such that, for a fixed valuation r, $E_1(b + \phi^i y_0) \equiv E_2(b + \phi^i y_0) \equiv 0 \mod \langle p, \phi^r \rangle$, where $b + \phi^i *$ is basically $a_1 + \phi^{i_1}(a_2 + \phi^{i_2} *)$ as obtained in Steps 13 and 16 of Algorithm **??**.

We are now left to filter out those y_0's for which $E_2(b + \phi^i y_0)$ has valuation wrt ϕ as more than r. This can be done from Lemma 5.6 to obtain an unique $\theta \in R_0 / \langle \phi \rangle$ such that $E_2(b + \phi^i(\theta + \phi y_0)) \equiv 0 \mod \langle p, \phi^{r+1} \rangle$.

The partial roots are stored in the sets Z_r and Z'_r, where Z'_r contains these "bad" values filtered out, while Z_r contains all the possible roots in $b + \phi^i *$.

So if we choose a "good" $z_0 \in Z_r - Z'_r$, we can find z_1 given by $(E_1(z_0)/\phi^r)/(E_2(z_0)/\phi^r)$ mod $\langle p, \phi^{4a-r}\rangle$. From this we get the final sets $Z = a_0 + \phi^{i_0} Z_r$ and $Z' = a_0 + \phi^{i_0} Z'_r$ for $(a_0, i_0) \in S_0$ for the corresponding r, as given in Lemma 5.4. From this we get our desired results as outputs. □

This proof of correctness of the algorithm also gives us the proof of Theorem 5.1 explained through the algorithm. This can be used to find the number of factors as well. But in the next chapter, we give another new algorithm due to [8] which gives the count of all the basic irreducible factors modulo p^k for any integer $k \geq 1$ in deterministic polynomial time, using more techniques from mathematics.

Chapter 6

Counting roots modulo p^k

In this chapter we describe a deterministic approach to root counting of a polynomial modulo p^k given by [8]. This is an important result as it is the first deterministic algorithm to count the total number of roots modulo p^k for any integer $k \geq 1$. We have seen in Chapter 5 how factorization can be done modulo p^k for $k \leq 4$, but this is a randomized algorithm and does not work beyond $\mod p^k$. However although we are yet not able to achieve deterministic polynomial factorization in rings of the form $\mathbf{Z}/p^k\mathbb{Z}$, deterministic root counting was a progress towards the problem of root finding, and establishes some results which can be used in root finding as well. The paper [8] discusses methods to count basic irreducible factors as well, but in this article we only state the main idea which is used to count only the total number of roots to give a summary of the ideas that we might use in future. Also, we consider $R = \mathbb{Z}/p^k\mathbb{Z}$.

6.1 Main Idea

We give an algorithm which is due to [8] to the following theorem to deterministically count all the roots of a polynomial to prove the following theorem:

Theorem 6.1 ([8])**.** *Given a polynomial $f(x) \in \mathbb{Z}[x]$, we can count all the roots of $f(x) \mod p^k$ in deterministic poly$(deg(f), k \log p)$-time.*

The main idea is to partition the root set of $f \mod p^k$ into d disjoint sets, quite like representative roots store them. However now we store this roots in split ideals defined in Chapter 2. In this way we show how we can count them. However a specific root can still not be obtained from this in deterministic time. The algorithm basically counts all the lifts of each root of $f \mod p$ to $f \mod p^k$.

6.1.1 Data Structures Involved

In order to make this algorithm feasible, we need to construct efficient data structures to store the split ideals and perform computations on them. We define the list data structure $\mathcal{L}$ which partitions the root set of $f \mod p^k$ into $deg(f)$ many disjoint subsets. The construction and other operations on this $\mathcal{L}$ is done using some special arithmetic tools that we will discuss in the next section. One can draw a similarity between each representative root and one disjoint subset stored in $\mathcal{L}$, as we will soon see that the construction of $\mathcal{L}$ is quite similar to Algorithm 2, by considering a new polynomial $f(a + px)$ if a is a root.

A split ideal, as defined in Section 2.3, is denoted as an ideal I_l formed by $l+1$ polynomials and is of degree b. We saw how it is a triangular ideal $I_l = \langle h_0(x_0), h_1(\bar{x}_1), h_2(\bar{x}_2)$ $, \ldots h_l(\bar{x}_l)\rangle$ and $b = \Pi_{i=0}^{l} deg_{x_i}(h_i(\bar{x}_i))$ with properties states in Definition 2.3. It basically stores a subset of the root set of $f \mod p^k$ of size b, where the roots are considered upto precision $l+1$, i.e. the first $l+1$ coordinates of the p-adic expansion taken.

Now, we keep splitting these ideals until we reach a maximal ideal, which we will call maximal split ideal. Note that these do not give us the implicit roots, but gives us the root count. However, if we were able to solve a system of polynomial equations and find the variety generated, we would be able to find roots of a given polynomial modulo p^k in deterministic time as well!

We also describe the list data structure $\mathcal{L}$ which we will use. It basically stores a subset of the root set of $f \mod p^k$, might partition it as well. This is basically a technical way to represent the ideal I_l and store all of them, after splitting, in the form of a tree. We can view the maximal split ideal as a representative root, and in the end of the algorithm, $\mathcal{L}$ stores at most d maximal ideals. Suppose $\mathcal{L} = \{I_1(l_1, d_1), I_2(l_2, d_2), \dots I_n(l_n, d_n)\}$, where each ideal $I_j \subseteq \mathbb{F}_p[\bar{x}_{k-1}]$ has two parameters, the length l_j and degree d_j. After repeated partitioning, a maximal split ideal $I(l, D)$ stores a subset of the root set of $f \mod p^k$ which is of size D. Now l means the number of coordinates considered, and the rest will be included in the $*$ portion of the representative roots. This implies that $I(l, D)$ has size Dp^{k-l} for the corresponding root. For efficiency of the algorithm, we consider a stack S, where we store tuples of the form (I_j, f_{I_j}), and will keep updating the values. Now, $f_{I_j} = f(x_0 + px_1 + p^2x_2 + \dots p^{l_j-1}x_{l_j-1} + p^{l_j}x) \mod I_j$.

Based on these we also use the *root tree data structure.* It will be used to show that $|\mathcal{L}|$ and degree of split ideals in $\mathcal{L}$ always remains at most $deg(f)$. We will call this tree as RT.

In RT, each generator h_1 in any $I \in \mathcal{L}$ corresponds to an edge, and each node denotes the splitting of that ideal. There is an attribute to each node, called the degree, which measures the possible extensions to the next level. This degree is distributed to its children degrees.

Definition 6.2. Degree of leaves is defined to be 1.

Let $N_I = (I, f_I)$ be a node corresponding to a split ideal $I \subseteq \mathbb{F}_p[\bar{x}_l]$, where $f_I(\bar{x}_l, x) \in R[\bar{x}_l, x]$. Let α be the largest power of p dividing $f_I \mod \hat{I}$ and $g_I(\bar{x}_l, x) = f_I/p^\alpha \mod \hat{I}$ ($g_I = 0$ if $\alpha \geq k$), then the degree of N denoted by $[N]$ is $[N] = \max(1, deg_x(g_I \mod I) \times deg(I))$.

We also define, for each node $N_{\langle 0 \rangle} = (\langle 0 \rangle, f_{\langle 0 \rangle} = f(x))$. We will set $deg(\langle 0 \rangle) = 1$. So this follows from the definition that $[N_{\langle 0 \rangle}] = d = deg(f)$. [8] gives some more properties of the degree defined in this way.

The construction of the RT is as follows.

In order to show an upper bound, we will use the concept of the roots tree (RT), which basically keeps track of the updates on S. A node is given by $N = (I, f_I)$, where (I, f_I) is an element of the stack S. Each push into the stack will create a new node.

The root of RT is given by $N_{\langle 0 \rangle} = (\langle 0 \rangle, f_{\langle 0 \rangle} = f(x))$. Add a child node N_{I_0} to the root which corresponds to initialization of the stack with (I_0, f_{I_0}), where $I_0 = \langle h_0(x_0) \rangle$ as defined before.

If at some time, the algorithm pops (I_{l-1}, f_{l-1}) from S, then we make our current node the leaf node corresponding to (I_{l-1}, f_{l-1}) and do the following changes:

1. If I_{l-1} is grown as $I_l = I_{l-1} + \langle h_{l-1}(\bar{x}_{l-1}) \rangle$, then we create a child of $N_{I_{l-1}}$ using edge label h_l and label this new node N_{I_l}

2. If the algorithm reaches a dead-end (no updates in S and $\mathcal{L}$ occur at this point), then add a child labelled $\mathcal{D}$ to $N_{I_{l-1}}$. This indicates a dead-end in this branch. Similarly, if we obtain a maximal split ideal, then we add $\mathcal{M}$ as its child.

3. Suppose processing a split ideal I_{l-1} needs us to factorize h_i and hence factor each ideal in S. Then we consider each U and move to the ancestor node of it which corresponds to an ideal of length i, say $N_{U_{i-1}} = (U_{i-1}, f_{U_{i-1}})$. Then we make m copies of the subtree at $N_{U_{i-1}}$ and these subtrees are attached with an edge $h_{i,j}$ for every $j \in [m]$.

6.1.2 Mathematical Tools

The main tool involved in counting the roots is considering the p-adic representation. We know that for a polynomial $f \mod p^k$, a root has p-adic decomposition as $r_0 + r_1 p + \dots r_{k-1}p^{k-1}$. Thus we represent each root as $x_0 + x_1 p + x_2 p^2 + \dots x_{k-1}p^{k-1}$, where x_i's are variables storing each coordinate of a root, and for some x_j corresponding to a root, it depends on the previous x_i's, $0 \leq i < j$.

Another tool used is the Frobenius polynomial $x^p - x \mod p$. This contains all the roots of $f(x) \mod p$ and we can consider gcd of this frobenius polynomial with $f(x)$ in modulo p to find a product of all the roots. Thus the degree of the gcd represents the number of roots of f in $\mathbb{F}_p$. The next challenge is to find a method to store the next coordinate in the p-adic representation of each root. We have the polynomial $h_0(x_0)$ as the gcd of f with $x^p - x$ modulo p and store the ideal $I_0 = \langle h_0(x_0) \rangle$.

In order to obtain the next coordinate, we consider the bivariate polynomial $g(x_0, x_1)$ as $f(x_0 + px) \equiv p^\alpha g(x_0, x_1) \mod \hat{I}_0$, where $\hat{I}_0$ is a lift of I_0, which we will explain later, and α is such that $g \not\equiv 0 \mod p$. Note that the main idea for using modulo the ideal I_l is basically because, for a multivariate polynomial $h(x_1, x_2, \dots x_l, x)$, and constants a_i's, we have $h(a_1, a_2, \dots a_l, x) = h(x_1, x_2, \dots x_l, x) \mod \langle x_1 - a_1, x_2 - a_2, \dots x_l - a_l \rangle$. Now. notice that if we fix x_0 to any root of $f \mod p$, then the set of possible x_1 are the roots of $g(x_0, x) \mod p$ for some fixed x_0. So we again calculate the gcd as $gcd(g(x_0, x) \mod p, x^p - x) \mod I_0$ and denote this as h_1. Now we increment the split ideal as $I_1 = I_0 + \langle h_1(x_0, x_1) \rangle$. In this way we get a count of roots upto modulo p^2. In order to obtain till precision modulo p^{l+1} from p^l, we continue doing this method. We solve for a new g each step given by $f(x_0 + x_1p + \dots x_lp^l + xp^{l+1}) \equiv p^\alpha g(\bar{x}_l, x) \mod \hat{I}_l$. Then we consider $h_{l+1} = gcd(g(\bar{x}_l, x) \mod p, x^p - x) \mod I_l$ and update $I_{l+1} = I_l + \langle h_{l+1}(\bar{x}_{l+1})$. We update these ideals by adding to $\mathcal{L}$, which we will describe later.

We also need to show that this construction of $\mathcal{L}$ is efficient and $|\mathcal{L}| \leq deg(f)$. We saw in the description of RT how the degree represented in every node gets distributed to its children. Note that the operations on $\mathcal{L}$ like reduction modulo current split ideal, inversion, zero divisor testing, gcd, exponentiation, counting valuations wrt p etc. are all bounded by poly$(deg(f), k \log p, deg(I))$. However its more difficult to bound the number of iterations and $deg(I)$. We analyze the number of iterations with the help of roots-tree RT. Every node of the RT corresponds to an intermediate split Ideal I, where an edge height i from the root corresponds to $h_i(\bar{x}_i)$, which is a generator of I. Whenever we update the split as $I_l = I_{l-1} + h_l(\bar{x}_l)$, a child is added to the node that corresponds to I_{l-1}, and the edge connecting this child to the parent is labelled as h_l. Similarly, when we split an ideal at some $h_i(\bar{x}_i)$ into say m ideals, m new subtrees are created which is connected to the parent by edges labelled as h_i. In this way the depth of the RT upper bounds the number of iterations.

Now, in order to look at the degree distributions (Definition 6.2) in the RT, we look at each node N with an associated parameter $[N]$, which will denote the degree of node N. We can see how degree of a parent distributes to the degrees of its children. This gives an intuition to measure the possible extensions of x_l modulo I_{l-1}, which is a multiple of $deg(I_{l-1})$. Now, from the distributive property of $[N]$ for some ideal I_{l-1} comes from the fact that, when we update $I_l = I_{l-1} + \langle h_l \rangle$, the degree of the child is bounded by the multiplicity of roots in $h_l(\bar{a}, x)$ times $deg(I_{l-1})$, for some root $a \in \mathcal{Z}_{\mathbb{F}_p}(I_{l-1})$. These details will be explained later.

We now give an algorithm to prove that Theorem 6.1 holds true.

6.2 The Algorithm

In this section we develop an algorithm due to [8], used to partition the root sets of $f(x) \mod p^k$ and count the total number of roots using the idea developed in Section . This algorithm takes a monic $f \mod p^k$ of degree d and returns a set of at most d maximal split ideals, whose zero sets partition the root set of $f \mod p^k$. For the maximal ideal $I_j = \langle h_0(x_0), h_1(\bar{x}_1), \ldots h_l(\bar{x}_l)\rangle$, its root set $\mathcal{Z}_{\mathbb{F}_p}(I_j)$ has size $\Pi_{i=0}^{l} deg_{x_i} h_i(\bar{x}_i)$, and each such zero represents p^{k-l-1} zeroes of $f \mod p^k$ (denotes a representative root of the form $\beta + p^{l+1}*$). Adding over all j's we can thus get a count of the total number of roots of $f \mod p^k$.

The algorithm starts with initializing the stack S containing the roots of $f \mod p$, i.e. S contains only the ideal $I_0 = \langle h_0(x_0)\rangle$, where $h_0(x_0) = gcd(f(x_0) \mod p, x_0^p - x_0)$. The zero set of this ideal contains all the roots of $f \mod p$. By $\hat{I}_0 \subseteq R[x_0]$, the lift of I_0, we refer to the ideal generated by $\{h_0(x_0)\}$ over $R[x_0]$.

During the iterations of the main while loop of the algorithm, we pop a split ideal from S, and try to increase its precision, i.e. find out the next coordinate of the roots in their p-adic decomposition. This process leads to two cases. First case is that we get another split ideal such that the precision of the root set has increased by a new coordinate x_{l+1}, or the current split ideal factors into more split ideals, seen as children in the RT. Now after the splitting, in order to include x_{l+1} into the coordinates of the roots, we update f_I as $f_I(\bar{x}_l, x_{l+1} + px) \mod \hat{J}$, where J is the new split ideal. Next we compute $g(\bar{x}_l, x)$ as $f(x_0+x_1p+\ldots x_lp^l+p^{l+1}x) = p^\alpha g(\bar{x}_l, x) \mod I$ to prevent the degree blowup.

Continuing in this way, whenever we get a maximal split ideal, we move it to $\mathcal{L}$. However if the ideal can not be extended then we get a dead end and discard that ideal. Intuitively this is because a number which is a root of $f \mod p^l$ for some $l < k$ might not always be a root of $f \mod p^k$. In both these cases, the size of the stack decreases. We keep on doing this and terminate the algorithm when the size of S becomes 0.

In Algorithm 4, we use the following procedures:

1. Modify f: We do this whenever we push elements into the stack. We have a given split ideal $I \subseteq \mathbb{F}_p[\bar{x}_l]$ and let $f_I(\bar{x}_l, x) \in R[\bar{x}_l, x]$ be reduced $\mod \hat{I}$. Let $J \subseteq \mathbb{F}_p[\bar{x}_{l+1}]$ such that $J = I_l + \langle h_{l+1}(x_{l+1}) \rangle$, and $\hat{J}$ be its lift to $R[\bar{x}_{l+1}]$. Then we can perform operations like creating $f_J(\bar{x}_{l+1}, x) = f_I(\bar{x}_l, x_{l+1} + px) \mod \hat{J}$ in deterministic poly-time. A formal proof is available in [CITE]. Also, when we have the tuple $(U = \langle h_0(x_0), h_1(\bar{x}_1), \dots h_l(\bar{x}_l) \rangle, f_{\langle U \rangle}) \in S$, we can consider the factorization $h_i = h_{i,1} h_{i,2} \dots h_{i,m}$ and create tuples $(U_j, f_{\langle U_j \rangle})$ in deterministic poly-time. This has also been showin in [CITE].

2. Reduce$(a(\bar{x}_l), J_l)$ returns reduced form of a modulo J_l. [CITE] gives an algorithm to do this as well, by recursively reducing while fixing every variable and doing operations only on the last variable x_l as x.

3. Test-Zero-Div$(a(\bar{x}_l, I_l))$ either reports that $a \mod \hat{I}_l$ is not a zero divisor, or outputs a factorization of one of the generators of I_l when true. In this as well, operations are done by fixing all the variables but the last one and recursively calling and checking the leading term of x_l wrt I_{l-1}, which is a polynomial in $\bar{x}_{l-1}$.

4. GCD$(a(\bar{x}_l, x), b(\bar{x}_l, x), I_l)$ computes the gcd if $a(\bar{x}_l, x) \mod I_l$, $b(\bar{x}_l, x) \mod I_l$, considering x as a variable, or returns False if a zero divisor occurs in intermediate calculations.

Based on these, the algorithm to calculate the number of roots of $f \mod p^k$ is as follows.

Algorithm 4 Count roots modulo p^k

```
1: $\mathcal{L} = \phi$
2: $S = \phi$
3: Let $\tilde{f}(x_0) = f(x_0) \mod p$, degree $d$
4: $h_0(x_0) = gcd(\tilde{f}, x_0^p - x_0)$, $I = \langle h_0(x_0) \rangle \subseteq R_0[x_0]$, $\hat{I}$ is lift of $I$ to $R[x_0]$
5: $f_I(x_0, x) = f(x_0 + px) \mod \hat{I}$
6: $S \leftarrow push(I, f_I)$
7: while $S \neq \phi$ do
8:     $S_{top} \leftarrow pop(S)$, Let $S_{top} = (\{h_0(x_0), \ldots h_l(\bar{x}_l), f_I(\bar{x}_l, x))$
9:     Compute $\alpha, \tilde{g}$ such that $f_I \equiv p^\alpha \tilde{g}(\bar{x}_l, x) \mod \hat{I}$ such that $p^\alpha || f_I \mod \hat{I}$
10:    if $\alpha \geq k$ then Update $\mathcal{L} = \mathcal{L} \cup I$, Go to Step 7
11:    Let $\tilde{g} = g(\bar{x}_l, x) \mod I$ with $g_1(\bar{x}_l)$ being leading coefficient wrt $x$
12:    if TEST-ZERO-DIV($g_1(\bar{x}_l, I)) = True$ then
13:        TEST-ZERO-DIV($g_1(\bar{x}_l, I)$) returns factorization $h_i(\bar{x}_i) = h_{i,1}(\bar{x}_i) h_{i,2}(\bar{x}_i) \ldots h_{i,m}(\bar{x}_i) \mod I_{i-1}$ for some $i$, Go To Step 23
14:    Filter out distinct $\mathbb{F}_p$ roots by taking gcd with $x^p - x$
15:    Recompute $\tilde{g} = g(\bar{x}_l, x).g_1(\bar{x}_l)^{-1} \mod I$
16:    Using repeated squaring and reducing modulo $I + \langle \bar{g} \rangle$, compute $\tilde{h}_{l+1}(\bar{x}_l, x) = x^p - x \mod I$
17:    if $gcd(\tilde{g}, \tilde{h}_{l+1}, I) = False$ then
18:        This returns factorization $h_i(\bar{x}_i) = h_{i,1}(\bar{x}_i) h_{i,2}(\bar{x}_i) \ldots h_{i,m}(\bar{x}_i) \mod I_{i-1}$ for some $i$, Go To Step 23
19:    else if $\tilde{g}$ and $\tilde{h}_{l+1}$ are coprime then
20:        Ideal can not grow more, Go To Step 7
21:    else
22:        $gcd_x(\tilde{g}, \tilde{h}_{i+1}) \mod I$ is non-trivial, say $h_{l+1}(\bar{x}_l, x)$, which is monic wrt $x$. Substitute $x$ by $\bar{x}_{l+1}$ and update $J = I + \langle h_{l+1}(\bar{x}_{l+1}) \rangle$. Let $\hat{J}$ be the lift of $J$ to $R[\bar{x}_{l+1}]$. Compute $f_J(\bar{x}_{l+1}) = f(\bar{x}_l, x_{l+1} + px) \mod \hat{J}$, puch $(J, f_J)$ to $S$, Go To Step 7
23:    We are given factorization of $h_i \mid I_{i-1}$. Push $S_{top}$ back in stack. For every $(U, f_{\langle U \rangle}) \in S$, find $m$ split ideals $U_j$'s, compute $(U_j, f_{\langle U_j \rangle})$ and push all of them to stack for $j \in [m]$
24: return $\mathcal{L}$, the list of maximal splitting ideals partitioning root set.
```

6.3 Analysis of Algorithm 4

In order to prove the correctness of the algorithm, the main goal is to prove the following:

Theorem 6.3 ([8])**.** *Algorithm 4 describes the partition of the root set of $f \mod p^k$ through a list data structure $\mathcal{L}$, which is a collection of maximal ideals $I_1, I_2, \dots I_n$. It partitions the root set such that $\mathcal{Z}_R(f) = \cup_{j\in[n]} S_j$, where $S_j = \mathcal{Z}_{\mathbb{F}_p}(I_j)$*

Proof Outline. It was proven in [8] that S always contains split ideals, $l < k$ and $\alpha > l$ at any iteration. The also proved the interesting result that, for any two polynomials $z(\bar{x}_l), w(\bar{x}_l) \in \mathbb{F}_p[\bar{x}_l]$, and a split ideal $I_{l-1} \in \mathbb{F}_p[\bar{x}_{l-1}]$, the algorithm given to compute $\text{GCD}(z(\bar{x}_l), w(\bar{x}_l), I_{l-1})$ in [8] returns a polynomial $h(\bar{x}_l)$ such that for any $a \in I_{l-1}$, $h(\bar{a}, x)$ is same as $gcd(z(\bar{a}, x), w(\bar{a}, x))$ upto multiplication by units in $\mathbb{F}_p^\times$.

It is also assured that the algorithm finally terminates. This is true as, we know the degree of the node is distributed among its children in the RT. Whenever we have a dead end, it is same as removing that polynomial along with the degree. Now, it can be shown that the number of disjoint polynomials always increases and the total number is upper bounded by the degree. So the algorithm must terminate.

The complete proof of this can be found in [8] (Theorem 9). □

Based on the construction of the RT, [8] also showed some properties of RT which were used to later prove that the algorithm terminates after polynomial many steps. Note that at each step, the size of RT increases, but we never delete any node from it. So the number of iterations is upper bounded by the size of this tree after it is completely constructed.

From the construction, also note that for a node $N_I = (I, f_I(\bar{x}_l, x))$, and its child $N_J = (J, f_J(\bar{x}_{l+1}, x))$[8] proved that this algorithm runs for polynomial many steps, i.e. time complexity is polynomial in degree of f and $k \log p$.

Using this approach and the same calculations, [8] also give a method to count the basic irreducible factors mod p^k. They use the fact that a basic irreducible factor $g(x) \in (\mathbb{Z}/p^k\mathbb{Z})[x]$ of $f \mod p^k$ of degree b completely splits in the Galois ring $G(p^k, b) = \mathbb{Z}[y]/\langle p^k, \phi(y)\rangle$, wherre $\phi(y) \mod p$ is an irreducible polynomial of degree b. Conversely finding roots of a polynomial f in $G(p^k, b)$ is same as finding its basic irreducible factors, and the same approach follows.

Chapter 7

Conclusion and future work

In Chapters 5 and 6 we saw methods which are used to find factors and count roots of polynomials modulo prime powers. The approach in 5 gets increasingly difficult going modulo p^5 with more number of variables. For this we intend to employ techniques used to find roots of polynomials having more than one variables, efficiently. Indeed we saw in both Chapters 5 and 6 how writing the p-adic representation helps in approaches. So if we consider the each coordinate as a variable as intend to apply certain conditions to reduce them to root finding of polynomials. Chapter 5 did use root finding of a given polynomial $E(y)$. If we can solve this modulo higher powers of p, this approach might be possible to factorize polynomials modulo p^5 and higher. Indeed split ideals as used in 6 also is a system of polynomials. We can intend to do something similar and apply Hilbert's Nullstellensatz to decide if they are have a common root, and hence decide irreducibility.

Bibliography

[1] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory.* Vol. 1. Cambridge, MA: MIT Press, 1996.

[2] E. R. Berlekamp. “Factoring polynomials over finite fields”. In: *Bell System Technical Journal* 46(8) (1967), pp. 1853–1859.

[3] E.R. Berlekamp. “Factoring polynomials over large finite fields”. In: *Mathematics of Computation* 24 (July 1970), pp. 713–735. DOI: 10.1090/S0025-5718-1970-0276200-X.

[4] Jérémy Berthomieu, Grégoire Lecerf, and Guillaume Quintin. “Polynomial root finding over local rings and application to error correcting codes”. In: *Applicable Algebra in Engineering, Communication and Computing* 24.6 (2013), pp. 413–443.

[5] David Cantor and Hans Zassenhaus. “A New Algorithm for Factoring Polynomials Over Finite Fields”. In: *Mathematics of Computation* 36 (Apr. 1981). DOI: 10.2307/2007663.

[6] Jesus A. De Leora et al. “Computing infeasibility certificates for combinatorial problems through Hilbert’s Nullstellensatz”. In: *Journal of Symbolic Computation* (2011), pp. 1260–1283.

[7] Ashish Dwivedi. “On Complexity of Hilbert’s Nullstellensatz over Positive Characteristic”. MA thesis. Indian Institute of Technology, Kanpur, 2017.

[8] Ashish Dwivedi, Rajat Mittal, and Nitin Saxena. "Counting basic-irreducible factors mod p^k in deterministic poly-time and p-adic applications". In: *Computational Complexity Conference (CCC19)* (Feb. 2019).

[9] Ashish Dwivedi, Rajat Mittal, and Nitin Saxena. "Efficiently Factoring Polynomials Modulo p^4". In: *International Symposium on Symbolic and Algebraic Computation (ISSAC)* (July 2019), pp. 139–146. DOI: `10.1145/3326229.3326233`.

[10] *Factorization of polynomials modulo small prime powers.* Tech. rep. University of Paderborn, Germany, 1996.

[11] Joachim van zur Gathen and Silke Hartlieb. "Factoring modular polynomials". In: *International Symposium on Symbolic and Algebraic Computation (ISSAC)* (1996).

[12] Zeyu Guo, Nitin Saxena, and Amit Sinhababu. "Algebraic dependencies and PSPACE algorithms in approximative complexity". In: *Computational Complexity Conference* (2018).

[13] *Ideals, Varieties and Algorithms: An introduction to Computational Algebraic Geometry and Commutative Algebra.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[14] Kiran Kedlaya and Christopher Umans. "Fast Polynomial Factorization and Modular Composition". In: *SIAM Journal on Computing* 40 (Jan. 2008). DOI: `10.1137/08073408X`.

[15] Pascal Koiran. "Hilbert's Nullstellensatz is in Polynomial Hierarchy". In: *Journal of Complexity* 12.0019 (May 1996), pp. 273–286.

[16] Arjen Lenstra, H. Lenstra, and László Lovász. "Factoring Polynomials with Rational Coefficients". In: *Mathematische Annalen* 261 (Dec. 1982). DOI: `10.1007/BF01457454`.

[17] J. Milne. *Algebraic Geometry, version v6.02*. Lecture Notes in Mathematics, 2017.

[18] Peter N Panayi. "Computation of Leopoldt's P-adic regulator." PhD thesis. University of East Anglia, 1995.

[19] Carlo Sircana. "Factorization of polynomials over $\mathbf{Z}/(p^n)$". In: *International Symposium on Symbolic and Algebraic Computation (ISSAC)* (2017), pp. 405–412.

[20] Madhu Sudan. "Algebra and Computation". In: *Lecture Notes in Computer Science* (2005).