Charles University in Prague
Faculty of Mathematics and Physics

BACHELOR THESIS

Michal Ferov

Irreducible polynomials modulo p

Department of algebra

Supervisor: Mgr. Štepán Holub, Ph.D.
Study program: Mathematics
Field of study: Mathematical methods of information security

2008

I would like to thank to my supervisor Mgr. Štěpán Holub, Ph.D. for scientific counseling, many useful suggestions and advices, and for all his time. I would like also to thank to Mgr. Petr Jonáš for lending me his books on LaTeX and to RNDr. Marie Trtílková for showing me the right way.

I affirm that I wrote my bachelor thesis by myself and that I only used cited sources. I agree with distribution and publication of my work.

In Prague on the date Michal Ferov

Contents

Title: Irreducible polynomials modulo p
Autor: Michal Ferov
Department: Department of Algebra
Bachelor thesis advisor: Mgr. Štěpán Holub, Ph.D.
Advisors e-mail: holub@karlin.mff.cuni.cz

Abstract: In the presented work we study a probabilistic algorithm for finding an irreducible polynomial over $\mathbb{F}_q$ of specified degree n with expected number $O(n^3 \log(n) \log(q) + n^3 \log(n) \log(\omega(n)) + n^3\omega(n))$ of operations in $\mathbb{F}_q$. In the first chapter we show some basic properties of $\mathbb{F}_q[x]$ and derive necessary and sufficient conditions of irreducibility. In the second chapter we use Möbius inverse formula to show that probability of random monic polynomial of degree n being irreducible is at least $\frac{1}{2n}$. In the third chapter we show polynomial deterministic algorithm which decides whether given polynomial is irreducible or not.

Keywords: finite field, irreducible polynomial, Möbius inverse formula, fast exponentiation of polynomials

Název práce: Irreducibilní polynomy modulo p
Autor: Michal Ferov
Katedra: Katedra Algebry
Vedoucíbakalářské práce: Mgr. Štěpán Holub, Ph.D.
E-mail vedoucího bakalářské práce: holub@karlin.mff.cuni.cz

Abstrakt: V předložené práci studujeme probabilistický algoritmus pro konstrukci ireducibilních polynomů zadaného stupně n nad $\mathbb{F}_q$ s očekávaným počtem $O(n^3 \log(n) \log(q) + n^3 \log(n) \log(\omega(n)) + n^3\omega(n))$ provedených operací v $\mathbb{F}_q$. V první kapitole ukážeme některé základní vlastnosti $\mathbb{F}_q[x]$ a odvodíme nutné a postačující podmínky ireducibility pro polynomy. V druhé kapitole ukážeme, že pravděpodobnost $p_q(n)$, že náhodný polynom stupně n je ireducibilni nad $\mathbb{F}_q$, je alespoň $\frac{1}{2n}$. Ve třetí kapitole předvedeme deterministický algoritmus, který v polynomiálním čase rozhoduje, zda je vstupní polynom irreducibilní či nikoliv.

Klíčová slova: konečné těleso, ireducibilní polynom, Möbiova inverzní formule, rychlé umocňování polynomů

Chapter 1

Introduction

The aim of this work is to present an algorithm for obtaining irreducible polynomial over field $\mathbb{Z}_p$.

1.1 Motivation

Irreducible polynomials are used to build field extensions, which have many applications in coding theory, cryptography and many more. The fastest algorithm for constructing irreducible polynomials was presented by V. Shoup in 1994. It is a probabilistic algorithm that performs expected number of $O(n^{2+o(1)}+n^{1+o(1)}\log(q))$ operations in $\mathbb{F}_q$, however it uses different approach than "generate and test" paradigm. All known deterministic algorithms work efficiently only for fields of small characteristics and rely on generalized Riemann hypothesis. It was proven by Shoup in 1990 that problem of deterministic construction of irreducible polynomials can be polynomial-time reduced to problem of factoring polynomials over $\mathbb{F}_q$. Contrary to that, our algorithm follows "generate and test" paradigm.

1.2 Preliminaries

In following section we will introduce some basic definitions and facts that will be used throughout the whole work.

Definition 1.1. *A field* is a set $\mathbb{F}$ with two operations $+$, $\cdot$ satisfying following axioms:

(ag1) $\forall\, a, b, c \in \mathbb{F} : a + (b + c) = (a + b) + c$

(ag2) $\forall\, a, b \in \mathbb{F} : a + b = b + a$

(ag3) $\exists\, 0 \in \mathbb{F}\ \forall\, a \in \mathbb{F} : a + 0 = 0 + a = a$

(ag3) $\forall\, a \in \mathbb{F}\ \exists (-a) \in \mathbb{F} : a + (-a) = (-a) + a = 0$

(mg1) $\forall\, a, b, c \in \mathbb{F} : a \cdot (b \cdot c) = (a \cdot b) \cdot c$

(mg2) $\forall\, a, b \in \mathbb{F} : a \cdot b = b \cdot a$

(mg3) $\exists\, 1 \in \mathbb{F}\ \forall\, a \in \mathbb{F} : a \cdot 1 = 1 \cdot a = a$

(mg4) $\forall\, a \in \mathbb{F} \setminus \{0\}\ \exists\, a^{-1} \in \mathbb{F} : a \cdot a^{-1} = a^{-1} \cdot a = 1$

(d) $\forall\, a, b, c \in \mathbb{F} : a \cdot (b + c) = a \cdot b + a \cdot c$

(n) $0 \neq 1$

If $\mathbb{F}$ is *finite*, then $(\mathbb{F}, +, \cdot)$ is a finite field. We refer to field $\mathbb{E}$ as to subfield of $\mathbb{F}$, if $\mathbb{E}$ is a subset of $\mathbb{F}$ and operations $+, \cdot$ are identical in $\mathbb{E}$. We denote $\mathbb{E} \leq \mathbb{F}$. We also say, that $\mathbb{F}$ is an extension of $\mathbb{E}$.

Note 1.1.

- Axioms (ag1)-(ag4) say that $(\mathbb{F}, +)$ is abelian group, we will refer to it as to additive group of the field $\mathbb{F}$.
- Axioms (mg1)-(mg4) say that $(\mathbb{F} \setminus \{0\}, \cdot)$ is abelian group. We will refer to it as to multiplicative group of the field $\mathbb{F}$ and we will use notation $\mathbb{F}^*$.

Definition 1.2. *A polynomial over a field* $\mathbb{F}$ is a formal term $f = \sum_{i=1}^{n} a_i x^i$ where $a_0, \ldots, a_n \in \mathbb{F}$ and $a_n \neq 0$. The number n is the degree of the polynomial and we use notation $deg(f) = n$. We use $\mathbb{F}[x]$ to denote the set of all polynomials over field $\mathbb{F}$. It is straightforward, that $\mathbb{F}[x]$ with operations $+$ and $\cdot$ form commutative ring.

Let $\mathbb{F}[x]$ be ring of polynomials with operations $+$ and $\cdot$. Then function

$$\nu : \mathbb{F}[x] \mapsto \mathbb{N}$$

,

$$\nu(f) = \left\{ \begin{array}{rl} 0 & \text{if } f = 0 \\ deg(f) + 1 & otherwise \end{array} \right.$$

is Euclidian norm, therefore $\mathbb{F}[x]$ is Euclidian domain. That gives us some good properties we can work with. First of all, because every Euclidian domain is also unique factorization domain, every nonzero element of $\mathbb{F}[x]$ can be uniquely factorized to its irreducible factors. Other thing is that we can use extended Euclidean algorithm to compute greatest common divisor which implies following lemma.

Lemma 1.1. *Let $\mathbb{F}$ be a field, and let $f \in \mathbb{F}[x]$ be irreducible polynomial. Then set $\mathbb{E} = \{g \in \mathbb{F}[x] | deg(g) < deg(f)\}$ with addition and multiplication modulo f is a field.*

Proof. It is straightforward that all axioms from definition 1.1 are satisfied except for (mg4). Let $g \in \mathbb{E}$ nonzero, then we can use extended Euclidian algorithm to compute $\gcd(f, g)$. Therefore we have gained $\gcd(f, g) = 1 = af + bg$.

$$af + bg \equiv bg \equiv 1 \mod f$$

Therefore $g^{-1} = b \mod g$. □

We have just shown, that every irreducible polynomial over $\mathbb{F}$ somehow defines extension of $\mathbb{F}$. In fact, any polynomial somehow defines an extension. Further on, we will work with extensions called splitting fields.

Definition 1.3. Let $\mathbb{F}$ be a field and $f \in \mathbb{F}[x]$. We call extension $\mathbb{E}$ of field $\mathbb{F}$ a *splitting field of f over* $\mathbb{F}$ if

- $\mathbb{F} \leq \mathbb{E}$,
- $f \in \mathbb{E}[x]$ can be factorized into linear factors, in other words: $f = (x - \theta_1) \dots (x - \theta_m)$, where $m = deg(f)$ and $, \theta_1, \dots, \theta_m \in \mathbb{E}$
- $\mathbb{E}$ is the smallest possible.

This extension defined by polynomial f is given uniquely, as is given in following theorem.

Theorem 1.2. *For every field $\mathbb{F}$ and every polynomial $f \in \mathbb{F}[x]$ having degree at least 1 exist splitting field of f over $\mathbb{F}$. Every two splitting fields of f over $\mathbb{F}$ are $\mathbb{F}$-isomorphic.*

Proof. Proof can be found in [1] on page 9. □

Lemma 1.3. *Let $\mathbb{F}$ be a field of cardinality q, then for every $a \in \mathbb{F}$ we have $a^q = a$.*

Proof. If $a = 0$ then the equation holds trivially, therefore we can suppose $a \in \mathbb{F}^*$. The order of $\mathbb{F}^*$ is q so $a^{q-1} = 1$, therefore $a^q = a$. □

Lemma 1.4. *Let $\mathbb{F}$ be a field of cardinality q, then the two following polynomials are equal:*

$$x^q - x = \prod_{a \in \mathbb{F}} (x - a).$$

Proof. According to the lemma mentioned before, $a^q = a$ for every $a \in \mathbb{F}$, therefore $x - a$ divides $x^q - x$, furthermore polynomials $x - a$ and $x - b$ are relatively prime whenever a and b are distinct, so $\prod_{a \in \mathbb{F}} (x - a)$ divides $x^q - x$. Both polynomial have the same degree q and are monic therefore they are equal. □

1.3 Conditions for irreducibility

In this section we will derive necessary conditions that polynomial has to satisfy to be irreducible.

Theorem 1.5. *Let $f \in \mathbb{F}_q[x]$ be irreducible polynomial of degree n. Then $f|(x^{q^m} - x)$ if and only if $n|m$.*

Proof. (⇒) Let $f|(x^{q^n} - x)$. Polynomial $x^{q^n} - x$ can be factorized into linear factors in field $\mathbb{F}_{q^n}$ and therefore f also factorizes to linear factors. In other words there is a root of F in $\mathbb{F}_{q^n}$, let us denote it α. Now we got sequence of fields $\mathbb{F}_q \leq \mathbb{F}_q(\alpha) \leq \mathbb{F}_{q^n}$. Field $\mathbb{F}_q(\alpha)$ is an algebraic extension of $\mathbb{F}_q$ given by polynomial f and therefore it is isomorphic to $\mathbb{F}_{q^m}$. Therefore $n|m$.

(⇐) Let $n|m$. Then $\mathbb{F}_{q^m} \leq \mathbb{F}_{q^n}$. Field $\mathbb{F}_{q^m}$ is extension of $\mathbb{F}_q$ given by f, therefore it contains root α of polynomial f. Because α is also root of $x^{q^n} - x$ and f is minimal polynomial (after being divided by its leading coefficient) of α over $\mathbb{F}_q$, then $f|(x^{q^n} - x)$.

□

Corollary 1.6. *Polynomial $x^{q^n} - x$ is a product of all the monic irreducible polynomials over $\mathbb{F}_q[x]$ whose degrees divide n.*

Proof. Theorem 1.5 tells us that all the monic irreducible polynomials over $\mathbb{F}_q$ of degree dividing n divide $x^{q^n} - x$. In other words there are no other irreducible factors than those. Because $x^{q^n} - x$ has no multiple root in its splitting field $\mathbb{F}_{q^n}$, none of its irreducible factors appears in factorization of $x^{q^n} - x$ twice. □

This allows us to construct a simple deterministic algorithm which decides, whether given polynomial is irreducible or not. If f is irreducible, then it divides $x^{q^n} - x$ but does not divide $x^{q^d} - x$, where d is a non-trivial divisor of n. In fact, it does not divide $x^{q^k} - x$ for any $1 \leq k < n$.

```
IrredTest1($f$)
$h \leftarrow x$
for $i \leftarrow 1$ to $\left\lfloor \frac{\deg(f)}{2} \right\rfloor$
    do $h \leftarrow h^q$
        if $\gcd(h - x, f) \neq 1$
            then return $FALSE$
return $TRUE$
```

It is obvious that this algorithm will need to compute greatest common divisor $\Theta(\deg(f))$ times, but it is not necessary because we need to compute it only if $i | \deg(f)$. In fact, we only need to compute it when $i = \frac{\deg(f)}{p}$ where p is a prime divisor of $\deg(f)$. We present our solution to exercise 20.2 proposed in [2] to prove it.

Lemma 1.7. *Let $f \in \mathbb{F}_q[x]$ monic and $\deg f = n > 0$. Then f is irreducible if and only if following the conditions are satisfied for all p prime divisors of n.*

(i) $f | (x^{q^n} - x)$

(ii) $\gcd\left(x^{q^{\frac{n}{p}}} - x, f\right) = 1$

Proof. ($\Rightarrow$) Let f be a monic irreducible polynomial of degree n. Then (i) holds, because theorem 1.5 implies, that $f | (x^{q^n} - x)$. Second condition holds trivially.

($\Leftarrow$) The condition (i) tells us that f is product of one or more monic irreducible polynomials whose degrees divide n. Condition (ii) holds too, therefore f is not divisible only by any polynomial of degree d where $d|n$. Therefore f is irreducible.

□

Note 1.2. One might ask whether it is really necessary to test the condition (i). It is. We show that by simple example. Let $n = \deg(f)$ be a prime and let $f = ab$ where $1 < \deg(a), \deg(b)$. Then it is obvious that $f \not|(x^{q^n} - x)$ and $\gcd(x^q - x, f) = 1$, therefore if we did not test the condition (i) we would get the result that f is irreducible, but it is not.

This result gives a way to decrease the number of gcd computations to $\Theta(\omega(\deg(f))$ where $\omega(n)$ is number of distinct prime divisors of n. We will use this notation further in our work.

```
IrredTest2($f$)
$n \leftarrow \deg(f), n = p_1^{e_1} \dots p_k^{e_k}$
if $f|(x^{q^n} - x)$
   then for $i \leftarrow 1$ to $k$
            do $n_i \leftarrow \frac{n}{p_i}$
               if $\gcd(x^{n_i} - x, f) \neq 1$
                  then return FALSE
        return TRUE
   else return FALSE
```

Chapter 2

Probability bounds

In this chapter we will set bounds for probability for random polynomial over $\mathbb{F}_q$ to be irreducible by using Möbius inverse formula.

2.1 Möbius function and inverse formula

In this section we will derive formula for number of irreducible polynomials over $\mathbb{F}_q$

Definition 2.1. *Möbius function* $\mu : \mathbb{N} \mapsto \{-1, 0, 1\}$ is defined as follows:

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1 \\ (-1)^k & \text{n is a product of k distinct primes} \\ 0 & \text{if } p^2 | n \text{ for some } p \in \mathbb{P} \end{cases}$$

Lemma 2.1. *For any $n \in \mathbb{N}$ the Möbius function satisfies:*

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & \textit{if } n = 1 \\ 0 & \textit{if } n > 1 \end{cases}$$

Proof. It is straightforward that equality holds for $n = 1$. Now let's suppose that $n > 1$ and $n = p_1^{k_1} \ldots p_l^{k_l}$ is it's prime factorization. We need to take into account only those divisors d of n that are product of i distinct primes where

$1 \leq i \leq l$ because the others are zero from definition of Möbius function.

$$\sum_{d|n}\mu(d) = \mu(1) + \sum_{i=1}^{l}\mu(p_i) + \sum_{1\leq i_1<i_2\leq l}\mu(p_{i_1}p_{1_2}) + \cdots + \mu(p_1 \dots p_l)$$
$$= \sum_{i=0}^{l}(-1)^i\binom{l}{i} = (1-1)^l = 0$$

□

Theorem 2.2. Möbius inverse formula. *Let* $\mathbb{G} = (G, +)$ *be an abelian group and let* $H, h : \mathbb{N} \mapsto G$ *are functions. Then*

$$H(n) = \sum_{d|n} h(d)$$

if and only if

$$h(n) = \sum_{d|n}\mu(d)H\left(\frac{n}{d}\right) = \sum_{d|n}\mu\left(\frac{n}{d}\right)H(d)$$

Proof. Note that $\mu(d)H\left(\frac{n}{d}\right) = \sum_{d|n}\mu\left(\frac{n}{d}\right)H(d)$ holds trivially. First, let's prove ($\Rightarrow$):

$$\sum_{d|n}\mu(d)H\left(\frac{n}{d}\right) = \sum_{d|n}\left(\mu(d)\sum_{c|\frac{n}{d}}h(c)\right) = \sum_{d|n}\sum_{c|\frac{n}{d}}\mu(d)h(c)$$
$$= \sum_{c|n}\sum_{d|\frac{n}{c}}\mu(d)h(c) = \sum_{c|n}\left(h(c)\sum_{d|\frac{n}{c}}\mu(d)\right)$$
$$= h(n)\mu(1) + \sum_{c|n, c\neq n}\left(h(c)\sum_{d|\frac{n}{c}}\mu(d)\right) = h(n)$$

Where the last equality was obtained by using lemma 2.1.
Now, lets prove ($\Leftarrow$):

$$\begin{aligned}\sum_{d|n} h(d) &= \sum_{d|n} h\left(\frac{n}{d}\right) = \sum_{d|n}\sum_{c|\frac{n}{d}} \mu\left(\frac{n}{cd}\right) H(c) \\ &= \sum_{c|n}\sum_{d|\frac{n}{c}} \mu\left(\frac{n}{cd}\right) H(c) = \sum_{c|n}\left(H(c)\sum_{d|\frac{n}{c}} \mu\left(\frac{n}{cd}\right)\right) \\ &= H(n)\mu(1) + \sum_{c|n, c\neq n}\left(H(c)\sum_{d|\frac{n}{c}} \mu\left(\frac{n}{cd}\right)\right) = H(n)\end{aligned}$$

Again, the last equality was obtained by using lemma 2.1 □

Now, our goal is to specify the number of monic irreducible polynomials of degree n over $\mathbb{F}_q$, let's denote it $N_q(d)$. We already know that $x^{q^n} - x$ is a product of all monic polynomials whose degree divides n. By comparing q^n with degrees of canonical factorization of $x^{q^n} - x$ we get the following formula

$$N_q(n) = \sum_{d|n} dN_q(d)$$

, which we will use in following theorem.

Theorem 2.3. *The number $N_q(n)$ of monic irreducible polynomials of degree n over $\mathbb{F}_q$ is given by:*

$$N_q(n) = \frac{1}{n}\sum_{d|n} \mu\left(\frac{n}{d}\right) q^d = \frac{1}{n}\sum_{d|n} \mu(d) q^{\frac{n}{d}}.$$

Proof. We apply Möbius inverse formula to the group $\mathbb{G} = (\mathbb{Z}, +)$ and functions $h(n) = nN_q(n)$ and $H(n) = q^n$. $H(n) = \sum_{d|n} h(d)$ is satisfied because of the formula derived above, therefore we get

$$nN_q(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) q^d = \sum_{d|n} \mu(d) q^{\frac{n}{d}}.$$

□

2.2 Upper and lower bounds for $N_q(n)$

We already know the number of irreducible polynomials of specified degree, but the formula is not easy to compute and especially for great n it would take some time to get the exact value. However, we don't really need to be precise. In fact, we need lower and upper bounds, which behave asymptotically like $N_q(n)$. We present our solution to exercises 5.4.(7) and 5.4.(8) from [1], page 9.

Lemma 2.4.

$$\frac{1}{n}\left(q^n - \frac{q^{\frac{n}{2}}-1}{q-1}\right) \leq N_q(n)$$

Proof. It is important to notice, that every non-trivial divisor of n is no greater than $\frac{n}{2}$, therefore

$$\begin{aligned}\frac{1}{n}\left(q^n - \frac{q^{\frac{n}{2}}-1}{q-1}\right) \leq \frac{1}{n}\left(q^n - \frac{q^{\lfloor\frac{n}{2}\rfloor}-1}{q-1}\right) = \frac{1}{n}\left(q^n - \sum_{i=0}^{\lfloor\frac{n}{2}\rfloor} q^i\right) \\ \leq \frac{1}{n}\sum_{d|n}\mu\left(\frac{n}{d}\right)q^d = N_q(n)\end{aligned}$$

□

This lemma can be also used to prove existence of a irreducible polynomial of arbitrary degree. $N_q(n) \geq \frac{1}{n}\left(q^n - \frac{q^{\frac{n}{2}}-1}{q-1}\right) \geq 1$, therefore there has to exist at least one irreducible monic polynomial of degree n, therefore there is at least $q-1$ irreducible polynomials of degree n.

Lemma 2.5.

$$N_q(n) \leq \frac{1}{n}(q^n - q)$$

Where both sides are equal if n is a prime number.

Proof. Let $n = p_1^{e_1}\dots p_{\omega(n)}^{e_{\omega(n)}}$ be the prime factorization of n.

$$\begin{aligned}N_q(n) &= \frac{1}{n}\sum_{d|n}\mu(d)q^{\frac{n}{d}} \\ &= \frac{1}{n}\left(q^n - \sum_{i=1}^{\omega(n)} q^{\frac{n}{p_i}} + \sum_{1\leq i_1<i_2\leq\omega(n)} q^{\frac{n}{p_{i_1}p_{i_2}}} - \dots + (-1)^{\omega(n)} q^{\frac{n}{p_1\cdots p_{\omega(n)}}}\right)\end{aligned}$$

We can suppose that $2 \leq n$. It is obvious that $\omega(n) \leq \log(n)$. Since for all primes p we have $2 \leq p$. We can use that to obtain following inequality:

$$\sum_{i=1}^{\omega(n)} q^{\frac{n}{p_i}} \leq \omega(n)\sqrt{q^n} \leq \log(n)\sqrt{q^n} < q^n.$$

If we substitute n by any of its divisors the inequality still holds. By using this we get:

$$\sum_{i=1}^{\omega(n)} q^{\frac{n}{p_i}} > \sum_{1 \leq i_1 < i_2 \leq \omega(n)} q^{\frac{n}{p_{i_1} p_{i_2}}} > \cdots > q^{\frac{n}{p_1 \cdots p_{\omega(n)}}} \geq q.$$

Therefore we get:

$$\frac{1}{n}\left(q^n - \sum_{i=1}^{\omega(n)} q^{\frac{n}{p_i}} + \sum_{1 \leq i_1 < i_2 \leq \omega(n)} q^{\frac{n}{p_{i_1} p_{i_2}}} - \cdots + (-1)^{\omega(n)} q^{\frac{n}{p_1 \cdots p_{\omega(n)}}}\right) \leq \frac{1}{n}(q^n - q)$$

□

So we have lower and upper bound:

$$\frac{1}{n}\left(q^n - \frac{q^{\frac{n}{2}} - 1}{q - 1}\right) \leq N_q(n) \leq \frac{1}{n}(q^n - q)$$

In fact, we have proved that $N_q(n) \in \Theta\left(\frac{1}{n}q^n\right)$.

2.3 Probability

Now we want to determine the probability $p_q(n)$ for random monic polynomial of degree n being irreducible. Since there are q^n monic polynomials, the probability is $p_q(n) = \frac{N_q(n)}{q^n}$. By using lemmas 2.4 and 2.5 we get:

$$\frac{1}{n}\left(1 - \frac{q^{\frac{n}{2}} - 1}{q^n(q - 1)}\right) \leq p_q(n) \leq \frac{1}{n}\left(1 - \frac{1}{q^{n-1}}\right)$$

We use following modifications to obtain different lower bound:

$$\frac{1}{n}\left(1 - \frac{1}{\sqrt{q^n}}\right) \leq \frac{1}{n}\left(1 - \frac{1}{q^{\frac{n}{2}+}} + \frac{1}{q^n}\right) \leq \frac{1}{n}\left(1 - \frac{q^{\frac{n}{2}} - 1}{q^n(q - 1)}\right).$$

This bound is a little less precise, but it gives us nice insight about the probability $p_q(n)$.

$$\frac{1}{n}\left(1-\frac{1}{\sqrt{q^n}}\right) < p_q(n) < \frac{1}{n}\left(1-\frac{1}{q^{n-1}}\right)$$

Because every polynomial of dergee $n = 1$ is irreducible, we can assume that $n \geq 2$. We also know, that $q \geq 2$, therefore $\frac{1}{\sqrt{q^n}} < \frac{1}{2}$, which we use to get upper and lower bounds independent to q:

$$\frac{1}{2n} < p_q(n) < \frac{1}{n}$$

In fact, we have showed that $p_q(n) \in \Theta(\frac{1}{n})$. We will use the lower bound for $p_q(n)$ further in this work. Consider this trial-error algorithm:

```
IrredGen($n$)
1  $a_0 \leftarrow R, \ldots, a_{n-1} \leftarrow R$
2  $f \leftarrow \sum_{i=1}^{n-1} a_i x^i + x^n$
3  if $f$ is irreducible
4     then return $f$
5     else go to 1
```

Lemma 2.6. *The algorithm will perform expected number $O(n)$ of tests.*

Proof. Expected value is given by:

$$E(X) = \sum_{x_i \in X} x_i p_{x_i}.$$

Event x_i is that the algorithm has to generate $i - th$ polynomial, in other words that all the polynomial generated before were reducible. Value of x_i is 1 because the algorithm generates just one i-th polynomial. Probability p_{x_i} is equal to $(1 - p_q(n))^{i-1}$ therefore the expected number of trials is $\sum_{i=0}^{\infty} (1 - p_q(n))^i$. If we use $\frac{1}{2n}$ as the probability then we get:

$$\sum_{i=0}^{\infty}\left(1-\frac{1}{2n}\right)^k = \frac{1}{1-\left(1-\frac{1}{2n}\right)} = 2n.$$

So the expected number is in $O(n)$. □

What is the probability that we will need more than n trials? That means that all of those n polynomials the algorithm has generated were reducible so:

$$\left(1 - \frac{1}{2n}\right)^n \to e^{-\frac{1}{2}} = 0.6065306...$$

which is still greater than $\frac{1}{2}$, but if we double the number of trials, we get $e^{-1} = 0.36787...$ which is much better. In fact, if we generate cn polynomials, then the probability that none of them is irreducible is approximately $e^{-\frac{c}{2}}$.

Chapter 3

Test

In this chapter we will present a polynomial deterministic algorithm, which decides whether input polynomial is irreducible or not. First idea could be just to simply factorize the input using Berlekamp algorithm or Cantor-Zassenhaus algorithm, but this idea has few drawbacks. Cantor-Zassenhaus uses expected number of $O(n^3 \log(q))$ operations in $\mathbb{F}_q$, so it can get pretty slow for fields of big characteristics, whereas Berlekamp uses expected number $O(n^3 + n^2 \log n \log q)$ operations in $\mathbb{F}_q$ so it is faster for big fields, but it uses $O(n^2)$ memory, which can be quite demanding.

3.1 Fast exponentiation

In section 1.3 we have shown that it is not necessary to compute $\gcd(x^{q^{\frac{n}{d}}} - x, f)$ for every d divisor of n, but only when d is a prime number. Computing $\gcd(f, g)$ takes $O(n^2 \log(n))$ operations in $\mathbb{F}_q$ where $n = \max(\deg(f), \deg(g))$, therefore it is highly desirable to keep the degrees as low as possible, but degrees of polynomials $x^{q^{n_i}} - x$ grow exponentially with n. We can prevent this from happening by computing $\gcd(h_i - x, f)$ where $h_i = x^{q^{n_i}} \mod f$, because $\gcd(h_i - x, f)$ and $x^{q^{n_i}} - x$ are equal. This is straightforward from definition of Euclid's algorithm.

But now we face different problem: how to efficiently compute h_i? By using classic repeated-squaring method we get that if g is a polynomial of degree smaller than n we can obtain $g^q \mod f$ with using $O(\log(q))$ multiplications $\mod f$ which gives us $O(n^2 \log(q))$ operations $\mathbb{F}_q$ if we use the standard polynomial multiplication. Following algorithm takes as input two polynomials α and f where $\deg(\alpha) \leq \deg(f) = n$ and positive integer m.

```
Pow($\alpha, f, m$)
1  $m = a_0 + a_1 q + \cdots + a_{\log(m)} q^{\log(m)}$
2  $h_0 \leftarrow \alpha$
3  for $i \leftarrow 1$ to $\log(m)$
4      do $h_i \leftarrow h_{i-1}^q \mod f$
5  for $i \leftarrow 0$ to $\log(m)$
6      do $h_i \leftarrow h_i^{a_i} \mod f$
7  $h \leftarrow h_0 \dots h_{\log(m)} \mod f$
8  return $h$
```

Lemma 3.1. *Algorithm* Pow*(α,f,m) returns* $\alpha^{q^m} \mod f$ *after computing*

$$O(n^2 \log(m) \log(q))$$

operations in $\mathbb{F}_q$.

Proof. The first for cycle repeats $O(\log(m))$ times and in every loop there are $O(n^2 \log(q))$ operations in $\mathbb{F}_q$ performed. the second for cycle also repeats $O(\log(m))$ and in every single loop there are $O(n^2 \log(q))$ operations performed, because $a_i < q$ for every $i \in \{0, 1, \dots, \log m\}$. And on the line 7 we multiply $O(\log(m))$ polynomial of degree less than n mod f. Altogether there are $O(n^2 \log(m) \log(q))$ operations in $\mathbb{F}_q$ performed. □

However, we need to compute $\omega(n)$ polynomials. We could exponentiate every one of them separately, which would lead us to $O(n^2 \log(q) \log(n) \omega(n))$ operations in $\mathbb{F}_q$, but it can be done better. We present our solution to exercise 3.39 from [2]. Following algorithm gets as input two polynomials α, f where $\deg(a) \leq \deg(f) = n$ and a list of positive integers $(m_1, \dots, m_k)$, where $m_i > 1$ for $i \in \{1, ..., r\}$.Let $m = m_1 \dots m_k$ Output of the algorithm is $(\alpha^{m_1^*} \mod f, \dots, \alpha^{m_k^*} \mod f)$ where $m_i^* = \frac{m}{m_i}$.

```
MExp($\alpha, f, (m_1, \dots, m_k)$)
if $k = 2$
   then return $\alpha^{m_2}, \alpha^{m_1} \mod f$
   else  $e_1 \leftarrow m_{\frac{k}{2}+1} \dots m_k$
         $e_2 \leftarrow m_1 \dots m_{\frac{k}{2}}$
         $\alpha_1 \leftarrow \alpha^{e_1} \mod f$
         $\alpha_2 \leftarrow \alpha^{e_2} \mod f$
```

```
            return $\begin{cases} \text{MExp } (\alpha_1, f, (m_1, \ldots, m_{\frac{k}{2}})) \\ \text{MExp } (\alpha_2, f, (m_{\frac{k}{2}+1}, \ldots, m_k)) \end{cases}$
```

Lemma 3.2. *The algorithm* $\textsc{MExp}(\alpha, f, (m_1, \ldots, m_k))$ *will perform*

$$O(n^2 \log(k) \log(m))$$

operations in $\mathbb{F}_q$ *to compute* $(\alpha^{m_1^*} \mod f, \ldots, \alpha^{m_k^*} \mod f)$

Proof. One could object, that k does not necessarily have to be even, which is of course true. However, we can without loss of generality assume that k is a power of 2 because we can simply define $m_{k+1} = 1, m_{k+2} = 1, ..., m_{2^l}$ where $l = \lceil \log_2 k \rceil$ so $2^l < 2k$.
Let $k = 2^l$. It is obvious that run of the algorithm will look like complete binary tree with l levels. On the i-th level the algorithm computes 2^i exponentiations $\mod f$, but if we sum their complexities, we get

$$O(n^2(\log(m_1 \ldots m_{2^{l-1}}) + \cdots + \log(m_{2^l - 2^{l-i}+1} \ldots m_{2^l}))) = O(n^2 \log(m)).$$

There are l levels and $l \in O(len(m))$, therefore complete number of operations in $\mathbb{F}_q$ is $O(n^2 \log(m) \log(k))$. □

We also need to factorize n in order to find its prime divisors. We present our own algorithm which takes n as input and returns its prime factorization.

```
Factor($n$)
  $k \leftarrow 1, p \leftarrow 2$
  while $1 < n$
      do if $p|n$
            then $j \leftarrow 0$
                 while $p|n$
                     do $n \leftarrow \frac{n}{p}$
                        $j \leftarrow j + 1$
                        $p_k \leftarrow p$
                 $e_k \leftarrow j$
                 $k \leftarrow k + 1$
         $p \leftarrow p + 1$
  return $(p_1, \ldots, p_k), (e_1, \ldots, e_k)$
```

In algorithm $\textsc{Factor}(n)$ there are no operations in $\mathbb{F}_q$ performed, therefore we don't have to take it's complexity in account.

3.2 Test

In this section we present our deterministic algorithm that decides whether the input polynomial is irreducible.

```
IrredTest($f$)
 1  $n \leftarrow \deg(f)$
 2  $h \leftarrow$ Pow $(x, f, n)$
 3  if $h = x$
 4     then Factor $(n)$
 5          $e \leftarrow p_1^{e_1 - 1} \dots p_{\omega(n)}^{e_{\omega(n)} - 1}$
 6          $\alpha \leftarrow$ Pow $(x, f, e)$
 7          $h_1, \dots, h_{\omega(n)} \leftarrow$ MExp $(\alpha, (p_1, ..., p_{\omega(n)}))$
 8          for $i = 1$ to $k$
 9              do if $\gcd(h_i - x, f) \neq 1$
10                     then return FALSE
11          return TRUE
12     else return FALSE
```

Lemma 3.3. *The algorithm $TEST(f)$ performs*

$$O(n^2 \log(n) \log(q) + n^2 \log(n) \log(\omega(n)) + n^2 \omega(n) \log(n))$$

operations in $\mathbb{F}_q$ to give the correct answer.

Proof. First operations come on line 2 and line 6 is almost the same. Because e is smaller than n we get from lemma 3.1 that Pow(x, f, e) requires $O(n^2 \log(n) \log(q))$ operations. Then lemma 3.2 implies that the algorithm MExp$(\alpha, (p_1, ..., p_{\omega(n)}))$ performs $O(n^2 \log(n) \log(\omega(n)))$. If we use standard Euclid's algorithm to compute gcd which performs $O(n^2 \log(n))$ then the complexity of the last loop is $O(n^2 \omega(n) \log(n))$, because we have to $\omega(n)$ times compute greatest common divisor. When we sum it up we get

$$O(n^2 \log(n) \log(q) + n^2 \log(n) \log(\omega(n)) + n^2 \omega(n) \log(n)).$$

□

Now we can construct a probabilistic algorithm that takes $n \in \mathbb{N}$ as input and outputs irreducible polynomial of degree n:

```
IrredGen($n$)
$a0 \leftarrow R, \dots, a_{n-1} \leftarrow R$
$f \leftarrow a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + x^n$
if IrredTest(f)
    then return $f$
    else go to 1
```

Lemma 3.4. *The algorithm* IrredGen(n) *will perform expected number*

$$O(n^3 \log(n) \log(q) + n^3 \log(n) \log(\omega(n)) + n^3 \omega(n))$$

of operations in $\mathbb{F}_q$

Proof. Straightforward from lemma 3.3 and lemma 2.6. □

3.3 Notes on complexity

Lemma 3.3 can be rewritten in a more generalizing way. Let us denote complexity of multiplication of two polynomials whose degrees are less or equal n $M(n)$ and complexity of computing *gcd* of two polynomials whose degrees are less or equal $G(n)$, then lemmas 3.3 and 3.4 can be rewritten as:

Lemma 3.5. *The algorithm* IrredTest(f) *performs*

$$O(M(n) \log(n) \log(q) + M(n) \log(n) \log(\omega(n)) + G(n)\omega(n))$$

operations in $\mathbb{F}_q$ *to give the correct answer and the algorithm* IrredGen*(n) performs expected number of*

$$O(nM(n) \log(n) \log(q) + nM(n) \log(n) \log(\omega(n)) + nG(n)\omega(n))$$

operations in $\mathbb{F}_q$

Proof. Straightforward. □

For example if we use fast polynomial arithmetics(FFT), then $M(n) = n \log(n)$. I we also use Euclid's algorithm together with fast polynomial arithmetics, then $G(n) = M(n) \log(n) = n \log^2(n)$. Then the algorithm IrredGen(n) will perform expected number

$$O(n^2 \log^2(n) \log(q) + n^2 \log^2(n) \log(\omega(n)) + n^2 \log^2(n)\omega(n))$$

of operations in $\mathbb{F}_q$

Chapter 4

Conclusion

We have presented a probabilistic algorithm for generating irreducible polynomials of degree n over $\mathbb{F}_q$ that performs expected number of

$$O(n^3 \log(n) \log(q) + n^3 \log(n) \log(\omega(n)) + n^3 \omega(n))$$

and we have also shown that it can be reduced to

$$O(n^2 \log^2(n) \log(q) + n^2 \log^2(n) \log(\omega(n)) + n^2 \log^2(n) \omega(n))$$

if we use fast polynomial arithmetics instead of standard "high-school-like" methods. We have proven that the problem of finding a irreducible polynomial can be solved probabilistically in polynomial time and it was proven by Shoup that there is a deterministic polynomial algorithm, but it works only for fields of small characteristics. There are still at least two open problems:

- Can the problem of finding irreducible polynomial of degree n over fields $\mathbb{F}$ of big characteristics solved deterministically with performing $O(n^k)$ operations in $\mathbb{F}$ where k is fixed?
- Can the problem of finding irreducible polynomial of degree n solved probabilistically with performing expected number $O(n^c)$ operations in $\mathbb{F}$ where $c < 2$?

Bibliography

[1] Tůma J., Barto L.: Konečná tělesa, preprint, 2008.

[2] Shoup V.: *A Computational Introduction to Number Theory and Algebra, second version*, Cambridge University Press, 2008.

[3] Lidl R., Niederreiter H.: *Finite Fields*, Cambridge University Press, 1997.