Lecture 8: Factoring Univariate Polynomials over Finite Fields

Algorithmic Number Theory (Fall 2014)
Rutgers University
Swastik Kopparty
Scribe: John Kim

1 Review of Finite Fields

A finite field is a field with a finite number of elements. Let F be a finite field. What can we say about F? Since $1 \in F$, and F is finite, adding 1 to itself enough times eventually gives 0. The smallest number of copies of 1 required to get 0 is called the *characteristic* of F. The characteristic of F must be prime, call it p. It is well known that F has cardinality $q = p^n$ for some positive integer n. We write $F = \mathbb{F}_q$ for the unique finite field with cardinality q (see Fact 3).

Let's start with some basic facts about finite fields:

Fact 1. $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$. *So any finite field of prime cardinality is cyclic.*

Fact 2. $\mathbb{F}_{p^k} = \mathbb{F}_p[x]/ < f(x) >$, *where* $f(x)$ *is any irreducible polynomial of degree* k *in* $\mathbb{F}_p[x]$.

Fact 3. *Let* $q = p^n$ *for a prime* p *and a positive integer* n. *Then there is a unique finite field of cardinality* q. *It turns out* $\mathbb{F}_q = \{x \in \overline{\mathbb{F}_p} \mid x^q = x\}$.

Fact 4. *The multiplicative subgroup* $G = \mathbb{F}_q^*$ *is cyclic.*

Proof of Fact 4. Let $d = \exp(G)$ denote the smallest positive integer such that $g^d = 1 \forall g \in G$. Consider the polynomial $f(x) = x^d - 1$. Then $f(x)$ has at most d distinct roots. But every $g \in G$ is a root of $f(x)$ by the definition of $\exp(G)$. So $f(x)$ has at least $q-1$ distinct roots. Hence $d = q-1$, and so G is cyclic. □

Fact 5. *Let* $q = p^n$ *for some odd prime* p. *If* $d \mid q-1$ *then* $\exists$ *a subgroup* $G \subseteq \mathbb{F}_q^*$ *such that* $[\mathbb{F}_q^* : G] = d$. *It turns out* $G = \{x^d \mid x \in \mathbb{F}_q^*\}$, *the set of all* d*-th powers.*

Fact 6. *Let* $q = p^n$ *for some odd prime* p. *Approximately half of* $\mathbb{F}_q$ *are perfect squares, and consequently, approximately half are not perfect squares.*

2 Open Problems

The following open problems require a $\texttt{poly}(\log p, k)$ or $\texttt{poly}(\log p)$ time algorithm to close them.

Open Problem 1. *Given* p *prime and* k *a positive integer, find an irreducible polynomial of degree* k *over* $\mathbb{F}_p$.

This problem is open even for $k = 2$. There is a randomized algorithm that works.

Open Problem 2. *Given p prime, find a generator of $\mathbb{F}_p^*$.*

This problem is believed to be doable. A random element is a generator with probability $\frac{1}{\log \log p}$.

Open Problem 3. *Given p prime and $b \in \mathbb{F}_p^*$, is b a generator of $\mathbb{F}_p^*$?*

Open Problem 4 (Discrete Log Problem)**.** *Given p prime and $b, c \in \mathbb{F}_p^*$, find d (if any) such that $b^d = c$ in $\mathbb{F}_p$.*

3 Finding k-th Roots in Finite Fields

We begin with the problem of finding k-th roots.

Question 1 (Finding k-th Roots)**.** *Given p prime, $a \in \mathbb{F}_q^*$, and $k \in \mathbb{N}$, does there exist $b \in \mathbb{F}_q^*$ such that $b^k = a$ in $\mathbb{F}_q$?*

If we can factor $x^k - a$, then we can find b. We may assume $k \mid q-1$ for the following reason. By Fact 4, $\mathbb{F}_q^* \cong \mathbb{Z}/(q-1)\mathbb{Z}$. Hence, $x \in \mathbb{F}_q^*$ is a perfect k-th power iff x is a perfect d-th power (where $d = \gcd(k, q-1)$) iff $x^{\frac{q-1}{d}} = 1$.

Let's consider the simpler special case where p is an odd prime and $q = p$, so that $2 \mid q-1$. Can we find square roots?

Question 2 (Finding Square Roots)**.** *Given p odd prime, and $a \in \mathbb{F}_p$, find $b \in \mathbb{F}_p$ such that $b^2 = a$ in $\mathbb{F}_p$?*

Equivalently, what are the roots of $x^2 - a$? We should try and understand more about the roots. Can the roots be the same? If b is a root, then $-b$ is also a root. $b = -b$ iff $2b = 0$ iff $b = 0$ iff $a = 0$. Hence, the roots are the same iff $a = 0$. If $a \neq 0$, then there are 2 distinct roots: $\pm b$.

If we can find a polynomial $R(x)$ such that:

1. b is a root of $R(x)$
2. $-b$ is not a root of $R(x)$

Then the $\gcd(R(x), x^2 - a) = x - b$, giving us the value of one of the roots b (and hence both of the roots).

Consider the polynomial $R(x) = x^{\frac{p-1}{2}} - 1$, which has the following properties:

1. $R(x)$ has $\frac{p-1}{2}$ roots (around half of $\mathbb{F}_p$ are roots)
2. $R(x)$ is a sparse polynomial

It is entirely possible that both b and $-b$ are roots of $R(x)$ or neither are roots of $R(x)$. To get a deterministic algorithm that finds the square root of a, we want a polynomial that selects exactly one of $\pm b$ to be a root. If instead, we try a randomized algorithm, we only require a polynomial that selects exactly one of $\pm b$ to be a root half the time. Berlekamp realized that by applying a randomized affine shift to $x^2 - a$, we can essentially randomize the two roots. Then exactly one of the randomized roots will be a root of $R(x)$ with probability $\frac{1}{2}$.

3.1 Berlekamp's Algorithm for Finding Square Roots in $\mathbb{F}_p$

Let p be an odd prime. To factor $x^2 - a$ in $\mathbb{F}_p$, we first pick $c, d \in \mathbb{F}_p$ uniformly at random. Define $G(x) = (cx+d)^2 - a$, the randomized affine shift of $x^2 - a$. Then the roots of $G(x)$ are $\frac{b-d}{c}$ and $\frac{-b-d}{c}$. As c and d are independent uniform random, $\frac{b-d}{c}$ and $\frac{-b-d}{c}$ are also independent uniform random. Hence, we have come up with a new quadratic whose roots are independent uniform random, and finding the roots of this new quadratic will tell us the roots of the original quadratic $x^2 - a$.

Next, we compute the $\gcd(R(x), G(x)) = \gcd(x^{\frac{p-1}{2}} - 1, G(x))$. As $\deg(G(x)) = 2$, the Euclidean Algorithm finishes in at most 3 steps, the first of which is the most computationally difficult. We must somehow compute $x^{\frac{p-1}{2}} - 1 \pmod{G(x)}$. Note that it is enough to find $x^{\frac{p-1}{2}} \pmod{G(x)}$. We may do this using the standard method of repeated squaring involving the binary representation of $\frac{p-1}{2}$. Note that we used the sparseness of $R(x)$ in this step. For sparse $R(x)$, the repeated squaring computation only happens very few times.

As discussed earlier, the gcd will be linear with probability $\frac{1}{2}$. This gives us a root of $G(x)$, which in turn allows us to get a root of $x^2 - a$.

Note that the algorithm only works for p an odd prime. When $p = 2$, the polynomial $x^{2^{n-1}} - 1$ has only a single root as $x \mapsto x^2$ is an automorphism. This suggests the following question:

Question 3. *In F_{2^n}, is there a sparse polynomial with around half the field as roots?*

Yes, the trace polynomial $x + x^2 + x^4 + \cdots + x^{2^{n-1}}$.

4 General Factoring

Our plan for factoring any polynomial is this:

1. Remove squared factors
2. Make all irreducible factors have the same degree
3. Berdekamp's randomness trick

4.1 Removing Squared Factors

We use the following lemma to identify squared factors:

Lemma 1. *If $a(x)^2 \mid f(x)$, then $a(x) \mid \gcd(f(x), f'(x))$.*

Proof. Suppose $f(x) = a(x)^2 b(x)$. Then:

$$f'(x) = 2a(x)a'(x)b(x) + a(x)^2 b'(x).$$

So $a(x) \mid f'(x)$. □

To remove squared factors of $f(x)$, compute the $g(x) = \gcd(f(x), f'(x))$. Consider $\frac{f(x)}{g(x)}$ and repeat gcd trick to remove its squared factors, etc.

This algorithm makes progress as long as $f'(x) \neq 0$. When does the algorithm get stuck? We need to know when $f'(x) = 0$.

Write $f(x) = \sum_{i=0}^{d} a_i x^i$. Then $f'(x) = \sum_{i=0}^{d} i a_i x^{i-1}$. If $f'(x) = 0$, then $p \mid i a_i$ for $0 \leq i \leq d$. So either $a_i = 0$ or $p \mid i$. So we may write:

$$f(x) = \sum_{i=0}^{m} a_{ip} x^{ip} = (b(x))^p,$$

where $b(x) = \sum_{i=0}^{m} (a_{ip})^{\frac{1}{p}} x^i = \sum_{i=0}^{m} (a_{ip})^{p^{n-1}} x^i$.