10 Factoring Polynomials over Finite Fields

One of the most fundamental operations in algebraic computing is factoring. It is also one where modern computer algebra has seen some of its greatest successes. In this section we look at factoring polynomials over finite fields.

Let R be any integral domain. Recall that an *irreducible* element of a ring R is any $a \in \mathsf{R}$ such that a cannot be written as $b \cdot c$ for non-units $b, c \in \mathsf{R}$. We define a *Unique Factorization Domain* as a ring such that any $a \in \mathsf{R}$ can be written as a product

$$a = a_1 a_2 \cdots a_\ell,$$

where the a_i are irreducible, and unique up to their order and up to multiplication by a unit.

We know that $\mathbb{Z}$ is a unique factorization domain, and so is any polynomial with coefficients from a unique factorziation domain. Over $\mathsf{F}[x]$, F a field, we define the factorization problem as, given $f \in \mathsf{F}[x]$, find distinct, monic, irreducible $f_1, \ldots, f_\ell \in \mathsf{F}[x]$ and $e_1, \ldots, e_\ell \in \mathbb{Z}_{>0}$ such that

$$f = \mathrm{lc}(f) \cdot f_1^{e_1} f_2^{e_2} \cdots f_\ell^{\ell}.$$

Here $\mathrm{lc}(f)$ is the leading coefficient of f, the coefficient of the highest degree term. We will say an f is *squarefree* if it is not divisible by the square of another polynomail (or higher power) of any other polynomial of degree greater than zero.

We will examine this problem for $\mathsf{F} = \mathbb{F}_q$, the finite field with q elements in it. For now you can think of q being prime, so $\mathbb{F}_q = \mathbb{Z}_p$. We break down the problem into three distinct phases (though our final algorithm won't quite do this).

- **Squarefree Factorization.** Find the largest factor of $f \in \mathbb{F}_q[x]$ which is squarefree, generally referred to as the *squarefree part* of f.
- **Distinct Degree Factorization.** Given a squarefree $f \in \mathbb{F}_q[x]$, find $g_1, g_2, \ldots, g_n \in \mathbb{F}_q[x]$ such that g_d is the product of all factors of f which have degree d, for $1 \leq d \leq n$.
- **Equal Degree Factorization.** Given a squarefree $g \in \mathbb{F}_q[x]$, all of whose irreducible factors have some fixed degree d (which is known), find $h_1, \ldots, h_k \in \mathbb{F}_q[x]$ of degree d such that $g = h_1 h_2 \cdots h_k$.

We will look at distinct degree factorization first, followed by equal degree factorization, followed by squarefree factorization (and how to sometimes avoid it).

10.1 Distinct Degree Factorization

Distinct degree factorization is based on the following theorem of Fermat, often referred to as *Fermat's Little Theorem.*

Theorem 10.1 (Fermat's Little Theorem)**.** *For any nonzero $a \in \mathbb{F}_q$ we have $a^{q-1} = 1$, and for all $a \in \mathbb{F}_q$, $a^q = a$ and*

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

Please see the text for a proof.

Let's first look at how to find all the factors of degree one. Assume that $f \in \mathbb{F}_q[x]$ is squarefree. All factors of degree one are also factors of $x^q - x$ by Fermat's Little Theorem, and so to separate all factors of degree one, we can compute

$$g_1 = \gcd(x^q - x, f) \in \mathbb{F}_q[x].$$

In fact, this is useful even when f is not squarefree. If $(x-\alpha)^e$ divides f for any $e \geq 1$, then certainly $(x-\alpha)$ divides $x^q - x$. Thus $\gcd(x^q - x, f)$ is a product of all *unique* degree one factors of f (and does not take into account the exponent on the factor in f, sometimes called its multiplicity).

To compute this GCD efficiently we need to use a slightly different method than the standard one. First compute $b_1 = x^q \text{ rem } f$. Then $\gcd(x^q - x, f) = \gcd(b_1 - x, f)$. To compute b_1, we perform repeated squaring.

Algorithm: `RepeatedSquaring`
Input: ‣ $f \in \mathsf{F}[x]$ of degree n and $u \in \mathsf{F}[x]$ of degree less than n, and $m \in \mathbb{N}$
Output: ‣ $u^m \text{ rem } f \in \mathsf{F}[x]$
(1) Write m in binary:

$$m = 2^k + m_{k-1} \cdot 2^{k-1} + m_{k-2} \cdot 2^{k-2} + \cdots + m_1 \cdot 2 + m_0$$

(2) $w \leftarrow u$; $v \leftarrow 1$;
(3) For $i = 0, 1, 2, \ldots, k$ do
(4) if $m_i = 1$ then $v \leftarrow v * w \text{ rem } f$;
(5) $w \leftarrow w * w$;
(6) Return v.

It is fairly easy to see that the algorithm is correct and that the cost is $O(k) = O(\log m)$ multiplications and divisions with remainders of polynomials of degree n. In other words, the cost is $O(\mathsf{M}(n) \log m)$ operations in F.

Now to compute $b_1 = x^q \text{ rem } f$, we perform repeated squaring, and this requires $O(\mathsf{M}(n) \log q)$ operations in $\mathbb{F}_q$. To compute $\gcd(b_1, f)$ requires $O(\mathsf{B}(n))$ operations in $\mathbb{F}_q$.

To find factors of higher degree, we rely on the following theorem, which is a generalization of Fermat's Little Theorem. For the proof, please see the text.

Theorem 10.2. *For any $d \geq 1$, the polynomial $x^{q^d} - x \in \mathbb{F}_q[x]$ is the product of all monic irreducible polynomials in $\mathbb{F}_q[x]$ whose degree divides d.*

Now we can write a complete algorithm for distinct-degree factorization.

Algorithm: `DistinctDegreeFactorization`
Input: ‣ $f \in \mathbb{F}_q[x]$ monic, squarefree of degree $n > 0$
Output: ‣ $g_1, \ldots, g_n \in \mathbb{F}_q[x]$ such that g_d is the product of all irreducible factors of f of degree d.
(1) $h_0 \leftarrow x$, $f_0 \leftarrow f$
(2) for i from 1 to n while $f_i \neq 1$ do
(3) Compute $h_i = h_{i-1}^q$ rem f by repeated squaring
(4) $g_i \leftarrow \gcd(h_i - x, f_{i-1})$
(5) $f_i = f_{i-1}/g_i$
End for;
(6) Return $g_1, g_2, \ldots, g_n$

To see that this works, note that $h_i \equiv h_{i-1}^q \equiv x^{q^i}$ for $i = 1 \ldots n$. At each stage we remove all factors of degree dividing i from f_i. Since we remove lower degree factors before higher degree factors, this means at Step i we only capture factors of degree i (all factors of a degree dividing i have been previously removed).

Each loop of the algorithm requires $O(\mathsf{M}(n)\log q)$ operations in $\mathbb{F}_q$ and there are at most n iterations of the loop, for a total cost of $O(n\,\mathsf{M}(n)\log q)$ operations in $\mathbb{F}_q$.

10.2 Equal degree factorization

Once we have performed distinct degree factorization, we have decomposed our (monic and squarefree) input polynomial $f \in \mathbb{F}_q[x]$ as

$$f = g_1 g_2 \cdots g_n$$

where $g_i \in \mathbb{F}_q[x]$ is the product of all monic irreducible factors of f of degree i.

To proceed, we will assume that q is an odd number (there is a slightly different algorithm when q is even, a power of 2). We need some fairly straightforward theorems, that you may have seen before, or can find Section 25.4 of the text.

Theorem 10.3. *If $h \in \mathbb{F}_q[x]$ is irreducible of degree d, then $\mathbb{F}_q[x]/(h)$ is a finite field with q^d elements. If $g = h_1 h_2 \cdots h_\ell$ for distinct, monic, irreducible polynomials $h_1, \ldots, h_\ell \in \mathbb{F}_q[x]$, then*

$$\frac{\mathbb{F}_q[x]}{(g)} \cong \frac{\mathbb{F}_q[x]}{(h_1)} \times \frac{\mathbb{F}_q[x]}{(h_2)} \times \cdots \times \frac{\mathbb{F}_q[x]}{(h_\ell)}.$$

That is, it is isomorphic to a direct product of finite fields.

Theorem 10.4. *Let $h \in \mathbb{F}_q[x]$ be monic and irreducible of degree d.*

(a) For any nonzero $a \in \mathbb{F}_q[x]$, $a^{q^d-1} \equiv 1 \bmod h$.

(b) For any nonzero $a \in \mathbb{F}_q[x]$, $a^{(q^d-1)/2} = 1$ if $a \equiv b^2 \bmod h$ for some $b \in \mathbb{F}_q[x]$, and $a^{(q^d-1)/2} = -1$ otherwise.

(c) The number of $a \in \mathbb{F}_q[x] \setminus \{0\}$ of degree less than d such that $a^{(q^d-1)/2} = 1$ is $(q^d-1)/2$ (and hence the number such that $a^{(q^d-1)/2} = -1$ is also $(q^d-1)/2$).

A few notes.

- Part (a) above is simply the generalization of Fermat's Little Theorem to all finite fields.
- In Part (b) when $a = b^2$ for some b, we say a is a *quadratic residue*.
- In Part (c), this says that half of the nozero the elements in a finite field are quadratic residues and the other half are non-residues.
- Parts (b) and (c) follow from the fact that all finite fields are cylic. That is, there is one element β in the field such that all non-zero elements in the field are powers of β.

Let's assume for now that $f = h_1 h_2$ for irreducible $h_1, h_2 \in \mathbb{F}_q[x]$ of degree d (so $\deg f = 2d$). Then we know that

$$\frac{\mathbb{F}_q[x]}{(f)} \cong \frac{\mathbb{F}_q[x]}{(h_1)} \times \frac{\mathbb{F}_q[x]}{(h_2)}.$$

If we choose a random polynomial $u \in \mathbb{F}_q[x]$ of degree less than $2d$ that is relatively prime with $f = h_1 h_2$, then this is exactly the same as choosing random $u_1, u_2 \in \mathbb{F}_q[x] \setminus \{0\}$ of degree less d, that is, $u \equiv u_1 \bmod h_1$ and $u \equiv u_2 \bmod h_2$, by the Chinese Remainder Theorem. In other words, choosing a random $u \in \mathbb{F}_q[x]$ of degree less than $2d$, that is not divisible by either h_1 or h_2, is the same as choosing a random nonzero element in each of $\mathbb{F}_q[x]/(h_1)$ and $\mathbb{F}_q[x]/(h_2)$.

Now, if we compute $u^{(q^d-1)/2}$, this is the same as computing $u_1^{(q^d-1)/2} \bmod h_1$ and $u_2^{(q^d-1)/2} \bmod h_2$. In each case these are ± 1 with probability $1/2$, independently. Suppose that

$$u_1^{(q^d-1)/2} \equiv 1 \bmod h_1, \text{ and } u_2^{(q^d-1)/2} \equiv -1 \bmod h_2.$$

or equivalently that

$$u^{(q^d-1)/2} \equiv 1 \bmod h_1, \text{ and } u^{(q^d-1)/2} \equiv -1 \bmod h_2.$$

This means that h_1 divides $u^{q^d-1} - 1$, but h_2 does not. Thus, we now have the basis for an algorithm:

(1) Choose a nonzero random $u \in \mathbb{F}_q[x]$ of degree less than $2d$.
If $\gcd(u, f) \neq 1$ then $\gcd(u, f)$ is either h_1 or h_2.

(2) Compute $v = u^{(q^d-1)/2} \operatorname{rem} f$.

(3) Compute $\gcd(v-1, f)$

With probability at least $1/2$, these three steps results in splitting f. If not, repeat.

This works similarly when $f \in \mathbb{F}_q[x]$ is any product of ℓ distinct monic irreducible factors of known degree d, for any $\ell \geq 2$. In this case

$$\frac{\mathbb{F}_q[x]}{(f)} \cong \frac{\mathbb{F}_q[x]}{(h_1)} \times \frac{\mathbb{F}_q[x]}{(h_2)} \times \cdots \times \frac{\mathbb{F}_q[x]}{(h_\ell)}$$

We then choose a random nonzero $u \in \mathbb{F}_q[x]$ of degree less than $\deg f$. if $\gcd(u,f) \neq 1$ then we have split f. Otherwise, $\gcd(u,f) = 1$ and the choice of u is the same as randomly (and uniformly) choosing nonzero elements $u_i = (u \operatorname{rem} h_i) \in \mathbb{F}_q[x]/(h_i)$ for each i in $\mathscr{I} = \{1,\ldots,\ell\}$. Suppose that $\mathscr{I} = \mathscr{I}_1 \cup \mathscr{I}_2$, where if $i \in \mathscr{I}_1$ then u_i is a quadratic residue in $\mathbb{F}_q[x]/(h_i)$, and if $i \in \mathscr{I}_2$ then u_i is not a quadratic residue in $\mathbb{F}_q[x]/(h_i)$. Then for all $i \in \mathscr{I}_1$, $h_i \mid u^{(d-1)/2} - 1$ and for all $i \notin \mathscr{I}_1$, $h_i \nmid u^{(d-1)/2)} - 1$. Thus,

$$\gcd(f, u^{(q^d-1)/2} - 1) = \prod_{i \in \mathscr{I}_1} h_i.$$

Assuming that we are "lucky" (which happens with probability at least 1/2), then we split f.

10.3 A complete factoring algorithm for $\mathbb{F}_q[x]$

To complete our factoring algorithm we need to handle an number of cases which we have not addressed so far, specifically non-monic polynomials and non-squarefree polynomials.

Non-monic polynomials are easy: simply write $f = \operatorname{lc}(f) \cdot \bar{f}$, where $\bar{f} \in \mathbb{F}_q[x]$ is monic. Then factor $\bar{f}$. For non-squarefree polynomials, most of what we have done so far works just fine. We proceed in stages for $i = 1,2,\ldots,n$. At stage i we determine all factors of degree i, along with their multiplicities, and remove them from f before continuing on.

Thus, to complete our algorithm, at the end of stage i we simply take each irreducible factor of f of degree i that we have computed, figure out how many times it divides f, and remove all factors of degree i from f before proceeding to stage $i+1$. Determining multiplicities and removing factors can be done simply by repeated quotient with remainder.

The total cost of our factoring algorithm is dominated by the cost of equal degree factorization, and requires an expected number of $O(n(\mathsf{M}(n)\log q + \mathsf{B}(n)))$ operations in $\mathbb{F}_q$ to completely factor a polynomial $f \in \mathbb{F}_q[x]$ of degree n.