CS681 **Computational Number Theory**

Lecture 5 and 6: Chinese Remaindering

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

In the next two lectures, we shall see a very important techinique used in computer science. The technique is called *Chinese Remaindering*. This comes in extremely handy in various arithmetic problems. We shall be looking at the problem of evaluating the determinant as a motivation.

1 Motivation for CRT: The Determinant

We are given an integer square matrix A and we are to find the determinant of the matrix. Before we talk about solving the problem, we need to understand the input as such. How big is the input?

The size is not just n^2 since the entries of the matrix also need to be represented. Hence the input size also depends on the size of the entries in the matrix. Let us call $\lambda = \max_{i,j} |a_{ij}|$. Then each entry in the matrix requires atmost $\log \lambda$ bits and there are n^2 entries. Thus, the input size is $n^2 \log \lambda$. We are looking for an algorithm that runs in time polynomial in the input size.

The naive approach is to do Gaussian Elimination, or the elementary row-operation method done in high-school: pick the first non-zero element in the first row, divide that row by the number (making it 1), and use this row to clear all other entries in that row.

This however has two problems:

1. Involves division and hence manipulating rational numbers.
2. Numbers in the matrix can become huge during gaussian elimination.

Firstly we need to understand why the first point is really a problem. We are given a matrix with just integer entries. We shall see now that that such a matrix will have an integer determinant. Thus, it may not be efficient to have rational number manipulation.

1.1 Integer Matrices have Integer Determinants

The group of permutations over n indices is denoted by S_n. Given any permutation in S_n, we can talk about the sign of the permutation. The definition is based on the fact that every permutation can be written as a product of cycles of length 2.

Lemma 1. *Every permutation σ can be written as a product of disjoint cycles.*

Proof. Start with the index 1. Look at the image of 1 which is $\sigma(1)$ and its image $\sigma(\sigma(1))$ etc. Eventually some $\sigma^i(1) = 1$ since the number of elements is finite. Hence this corresponds to the cycle $\left(1\ \sigma(1)\ \sigma^2(1)\ \cdots\ \sigma^i(1)\right)$. Now look at the next smallest index that has not been covered in this cycle and do the same. Our permutation σ is the product of these cycles and they are clearly disjoint. □

Lemma 2. *Every permutation σ can be written as a product of cycles of length 2.*

Proof. By the previous lemma, it is enough to show that every cycle can be written as a product of 2-cycles. And this is very easy to see. Consider any cycle of the form $(a_1\ a_2\ \cdots\ a_k)$. Easy to check that this is equal to the product $(a_1\ a_2)(a_1\ a_3)\cdots(a_1\ a_k)$. □

Now, if a permutation σ can be represented as a product of m 2-cycles, then we define the sign of σ to be $(-1)^m$. An immediate question is whether this is well defined. That is, suppose a permutation can be represented as a product of 7 such 2-cycles and also as a product of 24 2-cycles in a different way, won't it give two conflicting values for the sign of the permutation. The answer is that such a thing won't happen. It is not too hard to check but we leave this to the interested reader.

(Hint: Every permutation of n indices can be thought of as a tuple $(x_1, \cdots, x_n)$ where the i-th index corresponds to the image of i under the permutation. Now look at the sign of the expression

$$\prod_{i>j}(x_i - x_j)$$

The sign of this expression is an equivalent definition of the sign of the permutation.

Now can you see why a permutation cannot be expressed as a product of s transpositions and t transpositions where s and t are of different parity?)

This is another formula to evaluate the determinant.

$$detA = \sum_{\sigma \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}$$

As an example, any 2×2 matrix has its determinant as $a_{11}a_{22} - a_{12}a_{21}$ where the first term corresponds to the permutation $(1)(2)$ and the second to the permutation $(1\ 2)$.

It is clear from the above formula that a determinant of an integer matrix has to be an integer.

1.2 First Attempt: An Euclidian Approach

Since we are assured that the answer is going to be an integer, it doesn't make much of sense to use rational numbers during our computation. And besides, the denominators can grow really huge through successive row operations.

But the problem of division can be sorted out using a Euclid's Algorithm sort of approach. Consider the first row of the matrix. Suppose each element is a multiple of the least entry, then we are in good shape. There would be no need to divide at all. How do we make sure that this happens? Somehow get the gcd of the numbers as one of the entries!

Pick up the least element in the row, say a_{11}. Now every other $a_{i1} = qa_{11} + r$. Now subtract q times the first row from the i-th row. This essentially reduces every entry to the remainder when divided by a_{11}. Now continue this procedure by picking up the least element until you get the gcd of the numbers and then use it to kill every other entry in the row.

But, this still causes numbers to blow up. While we do operations to work on the first column, the other entries can grow to become too large.

1.3 Second Attempt: The Big Primes Method

One clever trick is to do all computations modulo a prime large enough. Since we know that the determinant is equal to $\sum sign(\sigma) \prod a_{i\sigma(i)}$, this value is atmost $n!\lambda^n$ since there are $n!$ terms and each term can be atmost $M = \lambda^n$. Now choose a prime P larger than this bound M.

Now division reduces to multiplying by the inverse modulo P and this can be done efficiently (this is the reason we want our P to be a prime. We can't choose any arbitrary number since inverses may not exist). Now the

gaussian elimination works and numbers will be bounded by P. Gaussian elimination modulo this prime P will give us a final answer D that is in the range $[0, P-1]$. But this could still mean that there are two choices for the integer determinant. The determinant could either be D or $D-P$. So to get around this small catch, we choose P to be a prime larger than $2M$. In this way, if the value we get is less than $P/2$, we know it is the determinant. Else, it will be $D-P$.

Thus we can solve the determinant problem by doing all computations modulo a large prime. But how do we get a large prime? How do we find a prime larger than the bound M quickly?

By a theorem on the density of primes, a random number between m and $2m$ is a prime with reasonably good probability. Thus we can just pick a random number, test if it is prime, if not pick again. We will hit a prime soon enough.

This however introduces randomness in our algorithm. We would like to have a deterministic polynomial time algorithm. This is where Chinese Remaindering comes in.

2 Chinese Remainder Theorem: Over Integers

Theorem 3. *Let $N = a_1 a_2 \cdots a_k$ such that each pair a_i, a_j are coprime. Then we have the following isomorphism between the two rings.*

$$\mathbb{Z}/(N\mathbb{Z}) \cong \mathbb{Z}/(a_1\mathbb{Z}) \times \mathbb{Z}/(a_2\mathbb{Z}) \times \cdots \mathbb{Z}/(a_k\mathbb{Z})$$

And more so, the isomorphism and the inverse map are computable easily.

Proof. First we look at the following homomorphism

$$\begin{array}{rcl} \phi : \mathbb{Z} & \longrightarrow & \mathbb{Z}/(a_1\mathbb{Z}) \times \mathbb{Z}/(a_2\mathbb{Z}) \times \cdots \mathbb{Z}/(a_k\mathbb{Z}) \\ x & \mapsto & (x \bmod a_1, x \bmod a_2, \cdots, x \bmod a_k) \end{array}$$

It is easy to check that this is indeed a homomorphism of rings. What is the kernel of the map? We are looking at the inverse image of $(0, 0, \cdots, 0)$. This just means that any x in the kernel must be $0 \bmod a_i$ for each i. And since the a_i's are coprime, this inturn means that x must be divisible by N. Thus the kernel of this map is $(N\mathbb{Z})$.

Hence, by the first isomorphism theorem, we have that the induced quotient map is an injective homomorphism:

$$\hat{\phi} : \mathbb{Z}/(N\mathbb{Z}) \hookrightarrow \mathbb{Z}/(a_1\mathbb{Z}) \times \mathbb{Z}/(a_2\mathbb{Z}) \times \cdots \mathbb{Z}/(a_k\mathbb{Z})$$

It's just left to show that the map is not only injective but also surjective; that would establish that it is indeed a homomorphism. Since the rings in the picture are finite rings, we can use a cardinality argument here. The cardinality of the ring on the left is N and so is the cardinality of the ring on the right N (since it is equal to $a_1 a_2 \cdots a_n$). Hence, since the map is injective between two sets of the same finite cardinality, it has to be an isomorphism.

This however will now help when the rings are infinite. For example $\mathbb{R}$ happily sits injectively inside $\mathbb{C}$ but they are clearly not isomorphic. We shall soon be getting to a general setting when such a cardinality argument won't work. Thus we need a more algebraic proof.

Here we shall use a small lemma.

Lemma 4. *We can easily compute elements x_i such that $x_i = 1 \bmod a_i$ and $x_i = 0 \bmod a_j$ for all $i \neq j$. In other words, the image of x_i is the tuple that has 1 on the i-th coordinate and 0 everywhere else.*

Pf: Since all the a_i's are pairwise coprime, a_i is coprime to $\bar{a}_i = \prod_{j \neq i} a_j$. Thus, by euclid's lemma, there exists elements x and y such that

$$x a_i + y \bar{a}_i = 1$$

Going modulo a_i, we get $y\bar{a}_i = 1 \bmod a_i$. And since $y\bar{a}_i$ is divisible by each other a_j, it is $0 \bmod a_j$. Thus this number $y\bar{a}_i$ is our required x_i and hence can be computed easily by the extended euclid's algorithm. □

Now that we have these x_i's, computing the inverse map is very simple. Given a tuple $(z_1, z_2, \cdots, z_k)$, the inverse image is just $\sum_{i=1}^{k} z_i x_i$.

Thus the map $\hat{\phi}$ is indeed an isomorphism and its image and inverse images can be computed easily. □

2.1 Solving Determinant through CRT

We are going to pick up small primes $p_1, p_2, \cdots, p_m$ such that $\prod p_i = N > 2M$ and then use chinese remaindering. How many primes do we need to pick? Since each prime is greater than or equal to two, the product of m distinct primes is clearly greater than 2^m. Thus in order to go larger than

$2M$, we just need to pick $\log 2M$ primes, which is $O(n \log n + n \log \lambda)$ and is clearly polynomial in the input size.

How do we go about picking them? Just keep testing numbers from 2 onwards, check if it is prime, and do this until we have enough primes. How long do we have to go before we get enough primes?

The prime number theorem tells us that the number of primes less than n is $O\left(\frac{n}{\log n}\right)$. With a little bit of calculations, it is easy to see that we would have found our m primes if we go just up till $m^2 \log^2 m$. And since m is polynomial in the input size, so is $m^2 \log^2 m$.

So we just need to look at all numbers up till $m^2 \log^2 m$ where $m = \log(2n!\lambda^n)$, and pick up all the primes. Note all these primes are extremely small, even their magnitude smaller than m which is about $\log M$. In the big prime method, we were picking a prime that required $\log M$ bits to even represent it in binary; its magnitude was about $2^{\log M} = M$. These primes are logarithmically smaller. Hence, to check if the numbers are really primes, you can use even the extremely inefficient exponential time sieve method or something. It also makes sense to store these small primes in the library, precompute them and keep it.

Now that we have these primes, we compute the x_i's as indicated in the lemma using the extended euclid's algorithm. Now we compute the determinant of the matrix A modulo each of these primes p_i using gaussian elimination. Let us say we get our value of the determinant mod p_i as d_i. Once you have done this calculation for each p_i, we get the tuple $(d_1, d_2, \cdots, d_m)$. Now using the x_i's, find the inverse map to get the value of the determinant D mod N. If this value is less than $N/2$, return it. Else, return $D - N$.

3 Chinese Remainder Theorem for Arbitrary Rings

In order to state the theorem for arbitrary rings, we need analogues of divisibility and coprimeness in terms of rings. This can be done using ideals. We say $m \mid n$, or m divides n, if every multiple of n is also a multiple of m. This interms of ideals translates to $m\mathbb{Z}$ containing the ideal $n\mathbb{Z}$.

As for coprimes, we know that two numbers a, b are coprime if there exist x, y such that $xa + yb = 1$. For this, we need a notion of an ideal sum.

Given ideals $\mathfrak{a}, \mathfrak{b}$ of a ring R, we define the sum-ideal as

$$\mathfrak{a} + \mathfrak{b} = \{a + b \ : \ a \in \mathfrak{a}, b \in \mathfrak{b}\}$$

This is also the ideal generated by the union of the two ideals.

Algorithm 1 DETERMINANT: USING CRT

Let $M = n!\lambda^n$ and $m > 2\log M$.
Enumerate the first $m^2 \log^2 m$ numbers and check for primes. Pick the first m primes.
Let $N = p_1 p_2 \cdots p_m$.
for $i = 1$ to m **do**
    Evaluate, using gaussian elimination, $det(A) \bmod p_i$.
    Let $d_i = det(A) \bmod p_i$.
    Evaluate, using extended euclid's algorithm, the x_i as in the lemma.
end for
Let $D = \sum_{i=1}^{m} x_i d_i$.
if $D < N/2$ **then**
    return D.
else
    return $D - N$.
end if

And now we can say that two ideals $\mathfrak{a}$ and $\mathfrak{b}$ are coprime if the ideal $\mathfrak{a} + \mathfrak{b} = R$.

Using these definitions, we have the Chinese Remainder Theorem for arbitrary rings. We state and prove the theorem for two ideals, but the general case is similar.

Theorem 5. *Let R be any commutative ring with identity. Let $\mathfrak{a}$ and $\mathfrak{b}$ be two ideals of R that are coprime to each other. Then we have the following ring isomorphism:*

$$R/(\mathfrak{a} \cap \mathfrak{b}) \cong R/\mathfrak{a} \times R/\mathfrak{b}$$

Proof. The proof is almost exactly the same as the proof of CRT over integers. Consider the following homomorphism.

$$\begin{aligned} \phi : R &\longrightarrow R/\mathfrak{a} \times R/\mathfrak{b} \\ x &\mapsto (x \bmod \mathfrak{a}, x \bmod \mathfrak{b}) \end{aligned}$$

Here, mod corresponds to the coset that contains x. This again is clearly a homomorphism. And the kernel is the set of all elements that ar $0 \bmod \mathfrak{a}$ and $0 \bmod \mathfrak{b}$ which means that the element is present in both $\mathfrak{a}$ and $\mathfrak{b}$. Thus the kernel of the homomorphism is $\mathfrak{a} \cap \mathfrak{b}$.

Thus all that's left to do is to show that the quotient map

$$\hat{\phi} : R/(\mathfrak{a} \cap \mathfrak{b}) \hookrightarrow R/\mathfrak{a} \times R/\mathfrak{b}$$

is also surjective. To show that, we construct the inverse map.

We need to find the inverse image of any given tuple (x, y). Since we know that the two ideals are coprime, $\mathfrak{a} + \mathfrak{b} = R$ and in particular contains the identity element 1.

Hence there exists two elements $a \in \mathfrak{a}, b \in \mathfrak{b}$ such that $a + b = 1$. Just as in the integer case, if we go modulo $\mathfrak{b}$ we see that a is an element that is $0 \bmod \mathfrak{a}$ but $1 \bmod \mathfrak{b}$ and similarly the element b. Now consider the element $xb + ya$. This is easily seen to be $x \bmod \mathfrak{a}$ and $y \bmod \mathfrak{b}$. Hence $xb + ya$ is the inverse image of (x, y).

Thus the map is also surjective and hence is indeed an isomorphism. □

Thus any ring that lets us compute the elements a, b such that $a + b = 1$ will allow CRT to go through. One such example is the ring of polynomials over integers. In this ring, the irreducible polynomial will act as the primes and we will be able to compute the determinant of a matrix with polynomial entries as well.

Another important property is that the units go to the units. That is, if you take any element x in the ring $R/(\mathfrak{a} + \mathfrak{b})$ whose inverse exists, corresponding tuple (a, b) also have the property that a and b are invertible. The proof of this is pretty easy.

Suppose an invertible element x was mapped to (a, b). Then, since the inverse exists, look at the image of the inverse of x. Lets call it (a', b'). By definition, $1 = xx^{-1} \mapsto (aa', bb') = (1, 1)$. And hence a' must be the inverse of a and b' the inverse of b. And thus if x is invertible, so is a, b and viceversa.

Chinese Remainder Theorem is a really powerful tool used in numerous occasions in computer science. We shall see more in the days to come.

CS681 **Computational Number Theory**

Lecture 7: Towards Factorization over Finite Fields

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

We shall slowly move into factorization of univariate polynomials (polynomials with just one variable) over finite fields. We are given a finite field K and a polynomial over one variable X whose coefficients are from K. We are to find the factorization of this polynomial into irreducible factors.

Before we get into this question, we need to first understand if it even makes sense. How can we be sure that such a factorization exists? And even if it did, how do we know if it is unique?

We shall first answer a lot of questions in the algebra related to it before going to factorization as such.

4 Rings, Ideals, Factorization etc.

We know that integers can be uniquely factorized into product of prime powers. However, not all rings are as well-behaved as the integers are. We first need to ask if the algebraic structure has this property of unique factorization. Let us look at an example where this fails.

Look at the set of integers modulo 8. This is called $\mathbb{Z}_8$ and we know that this forms a ring. Suppose we look at polynomials over this ring, polynomials of a single variable X whose coefficients come from $\mathbb{Z}_8$, does this ring have the property of unique factorization? Here is a counter example in $\mathbb{Z}_8[X]$.

$$X^2 - 1 = (X-1)(X+1) = (X-3)(X+3)$$

But $\mathbb{Z}_8$ is a bad ring, in the sense that non-zero elements can multiply to give 0 ($2 \times 4 = 0$ here). As for another example, look at the set of all number of the form $a + b\sqrt{-5}$ where $a, b \in \mathbb{Z}$. This forms a ring and over this ring $6 = 2 \cdot 3 = (1+\sqrt{-5})(1-\sqrt{-5})$.

Hence it's not always true that factorization is unique. However, fortunately for us, we have unique factorization over $K[X]$ whenever K is a field.

Definition 1. *A ring is said to be an integral domain if and only if there are no non-trivial zero divisors. That is, if $a, b \in R$ such that $ab = R$, then either $a = 0$ or $b = 0$.*

The ring $\mathbb{Z}$ is an integral domain but the ring of integers modulo 6 is not (since $2 \times 3 = 0$ over the ring).

Inorder to define factorization, we need a notion of primes over arbitrary rings. Let us first look at the definition of primes over integers. Let us first look at the wrong definition.

A number p is said to be prime if for all a that divides p, either $a = 1$ or $a = p$.

This translates to the ring definition of a maximal ideal and not a prime ideal.

This is the common definition in school but generalization based on this is erraneous. Though it happens to correct over the set of integers, it is not true in general. Here is the right definition.

Definition 2. *A number p is said to be a prime if and only if for all a, b such that p divides ab, either p divides a or p divides b.*

Thus this gives the definition of prime ideals in the setting of rings.

Definition 3. *An ideal $\mathfrak{a} \subseteq R$ is said to be prime if and only if for all $a, b \in R$ such that $ab \in \mathfrak{a}$, either $a \in \mathfrak{a}$ or $b \in \mathfrak{a}$.*

Any element $p \in R$ that generates a prime ideal is called a prime element of R.

Definition 4. *An ideal $\mathfrak{a} \subseteq R$ is said to be maximal if and only if for every ideal $\mathfrak{a}' \supseteq \mathfrak{a}$, either $\mathfrak{a}' = 1R = R$ or $\mathfrak{a}' = \mathfrak{a}$.*

This basically means that no proper ideal of R properly contains $\mathfrak{a}$. Note that not all prime ideals are maximal. We were just lucky that this was true on $\mathbb{Z}$ and hence both definitions of prime numbers were equivalent. This is not true over arbitrary rings.

Definition 5. *An ideal $\mathfrak{a} \subseteq R$ is said to be a principle ideal if the ideal is generated by a single element. That is, $\mathfrak{a} = aR$ for some $a \in R$.*

Definition 6. *An integral domain R is said to be a*

- *principle ideal domain (PID) if every ideal in it is principle (every ideal is generated by a single element).*

- *unique factorization domain (UFD) if every element can be uniquely factorized in to product of prime elements of the ring.*

We already saw an example of a ring (and a domain) that was not a UFD. Here is an example of a ring that is not a PID. Consider a field K and look at the ring of polynomials on two variables X, Y over this field. This is denoted by $K[X, Y]$.

In this field, look at the ideal generated by X and Y. That is, the set of polynomials of the form $Xf(X, Y) + Yg(X, Y)$, those polynomials that do not have a constant term. This is clearly an ideal but this isn't principle.

A similar example is over $\mathbb{Z}[X]$ and the ideal being (p, X) where p is any prime number.

Fact 1. *For any field K, $K[X]$ is a PID.*

Fact 2. *Any PID is also a UFD*

The two facts together tell us that we can indeed talk of factorization of polynomials in $K[X]$. Another useful fact is the following, and this helps us see that factorization makes sense even on multivariate polynomials.

Fact 3. *If R is a UFD, so is $R[X]$.*

The following theorems are very useful.

Theorem 6. *A ring R is a field if and only if the only ideals of R are the 0 ideal and the whole ring R.*

Proof. First we shall show that a field has no non-trivial ideals. Suppose The field had some ideal I that contained some element $x \neq 0$. Since it is a field, the inverse of x exists. Since I is an ideal and $x \in I$ would mean that $xa \in I$ for all $a \in R$ and in particular $xx^{-1} = 1 \in I$. But if $1 \in I$, then for every element a in the field, $1a \in I$ which would then force I to be the entire field.

As for the other direction, suppose the ring R was not a field. We want to show that there exists some non-trivial ideal in this ring. Since we assumed that it isn't a field, there must be some non-zero element a whose inverse does not exist. Look at the ideal generated by it, aR. This ideal certainly contains a and it cannot 1 since if it did, it would mean that a is invertible. And hence this is an ideal that is non-zero and also not the whole of R; a non-trivial ideal. □

Theorem 7. *For any ring R*

1. *if an ideal $\mathfrak{a}$ is prime, then $R/\mathfrak{a}$ is an integral domain.*
2. *if an ideal $\mathfrak{a}$ is maximal, then $R/\mathfrak{a}$ is a field.*

Proof. We have to show that if $\mathfrak{a}$ is prime, then $R/\mathfrak{a}$ is an integral domain. Suppose not, then there exists two non-zero elements a, b such that $ab = 0$ in $R/\mathfrak{a}$. This means that $a \bmod \mathfrak{a} \neq 0$ and $a \bmod \mathfrak{a} \neq 0$ but $ab \bmod \mathfrak{a} = 0$ or in other words $ab \in \mathfrak{a}$ but neither a nor b belongs to $\mathfrak{a}$. This contradicts the assumption that $\mathfrak{a}$ and hence R has to be an integral domain.

As for the case when $\mathfrak{a}$ is maximal, assume that $R/\mathfrak{a}$ is not a field. Then there exists some non-zero element that is not invertible. Look at the ideal generated by this element. As in the earlier theorem, this is a non-trivial ideal (neither 0 nor the entire ring). But in the map from R to $R/\mathfrak{a}$, ideals of $R/\mathfrak{a}$ corresponds to ideals in R that contain $\mathfrak{a}$. Since we just found a non-trivial ideal in $R/\mathfrak{a}$, this would translate to a non-trivial ideal in R that properly contains $\mathfrak{a}$ thus contradicting the maximality of $\mathfrak{a}$. Thus R has to be a field. □

4.1 Some Insights

This is not completely a part of the course but it would be useful to know this to understand factorization. In any ring, we can talk of a tower of prime ideals. What this means is a series of the form $0 \subseteq I_1 \subseteq I_2 \subseteq \cdots \subseteq I_n \subseteq R$ such that each ideal I_j is a prime ideal. The number n is called the Krull Dimension of the ring R.

The Krull Dimension is actually a local property but for it is well defined for rings like $K[X_1, X_2, \cdots, X_n]$ (where K is a field) and $\mathbb{Z}[X]$.

If we were to look at $K[X, Y]$, we have the tower $0 \subseteq (X) \subseteq (X, Y) \subseteq K[X, Y]$. The krull dimension of this ring is 2. Similarly the ring of polynomials on n variables over a field K will have a krull dimension of n.

And the ring $\mathbb{Z}[X]$ has the tower $0 \leq (p) \leq (p, X) \leq \mathbb{Z}[X]$ and hence has krull dimension 2. We shall see soon that factorization of polynomials in $\mathbb{Z}[X]$ is so similar to factorization of polynomials in $K[X, Y]$.

We need to understand the concept of finite fields, extensions, etc before we get into factorization. We shall first spend some time on this.

5 Finite Fields

We shall be studying properties of fields that have finite number of elements in them. A few things to keep in mind, we shall prove them soon, is that any finite field has its cardinality to be a power of prime. There cannot exist a finite field whose cardinality is divisible by two distinct primes. And infact, for any prime p and α, there is exactly one field of size p^α. (and note that this isn't true on the infinite setting. $\mathbb{R}$ and $\mathbb{C}$ both have infinite number of elements but are clearly different)

Definition 7. *A field E is called an extension of a field K if E is a field that contains K. This (also) is denoted by E/K.*

There is a notion of a degree of a field extension but one needs to be familiar with vector spaces to completely understand this. We shall dwell a little on it.

5.1 Vector Spaces

Definition 8. *A vector space V over a field K, with an additive structure and multiplication by elements of K (scalars), satisfies the following conditions:*

- *$(V, +)$ is an additive abelian (commutative) group (additive closure, inverse, identity)*
- *For any vector $v \in V$ and scalar $\alpha \in K$, the element αv is also a vector.*
- *For any vectors $u, v \in V$ and scalar $\alpha \in K$, we have $\alpha(u + v) = \alpha u + \alpha v$.*
- *For any vector u and scalars $\alpha, \beta \in K$, we have $(\alpha + \beta)u = \alpha u + \beta u$ and $\alpha(\beta u) = (\alpha\beta)u$.*

Let us look at a few examples to get ourself familiar with this notion. $\mathbb{C}$ forms a vector space over $\mathbb{R}$. Clearly the above properties are satisfied.

Another example is the plane $\mathbb{R}^2$, set of point (x, y) where both coordinates are from the reals. Scalar multiplication is defined as $\alpha(x, y) = (\alpha x, \alpha y)$.

Another example is the ring of polynomials $K[X]$ over K where K is a field. Scalar multiplication is just multiplying every coefficient by the scalar.

Next we need a notion of linear independance.

Definition 9. *A set $\{v_1, v_2, \cdots, v_k\}$ is said to be linearly independant if the only way*

$$c_1v_1 + c_2v_2 + \cdots + c_kv_k = 0$$

can happen for scalars c_i is when all the c_i's are zero themselves. That is, no non-trival linearl combination of these vectors is zero.

For example, let us look at each of our examples stated and find a linearly independant set. In $\mathbb{C}$ over $\mathbb{R}$, look at the set $\{3, 2+i\}$. Suppose $c_1(3) + c_2(2+i) = 0$, then $(3c_1 + 2c_2) + c_2 i = 0$ and this is possible only when both c_1 and c_2 are zero. Hence the set is linearly independant.

And again, look at the set $\{(1,0),(0,1)\}$ in $\mathbb{R}^2$. This again is linearly independant since the only way $c_1(1,0)+c_2(0,1) = (c_1, c_2) = (0,0)$ is when both c_1 and c_2 are zero.

In the third example, look at the set $\left\{1, X, X^2\right\}$. $c_1 + c_2X + c_3X^2$ can be the zero polynomial if and only if all the c_i's are zero.

This is the notion of linear independance. With a little bit of thought, any vector that can be represented as a linear sum from such a set is infact uniquely represented so.

For example, let us assume that $\{v_1, v_2, \cdots, v_k\}$ was a linearly independant set. Let $v = c_1v_1 + c_2v_2 + \cdots + c_kv_k$. Suppose this could be represented as a linear sum in a different way, we shall obtain a contradiction.

$$\begin{aligned} v &= c_1v_1 + c_2v_2 + \cdots + c_kv_k \\ &= c_1'v_1 + c_2'v_2 + \cdots + c_k'v_k \\ \implies 0 &= (c_1 - c_1')v_1 + \cdots + (c_k - c_k')v_k \end{aligned}$$

And if the two representations were indeed different, there is atleast one i such that $c_i \neq c_i' \implies (c_i - c_i') \neq 0$ but this would give a non-trivial linaer combination of the v_i's to become zero. This contradicts our assumption that they were linearly independant. Hence such linear representations are unique.

An example is that every point (x, y) can be represented uniquely as a linear sum of $(1,0)$ and $(0,1)$ (it is just $x(1,0) + y(0,1)$). The students are encouraged to also check it for $\mathbb{C}$ with our linearly independant set being $\{3, 2+i\}$.

Let us look at our example of $K[X]$. We saw that $\left\{1, X, X^2\right\}$ was a linearly independant subset but the term X^5 can never be written as a linear sum of $1, X, X^2$. Thus, the set $\left\{1, X, X^2\right\}$ doesn't cover or *span* X^5. Since X^5

is not spanned by the set $\{1, X, X^2\}$, we can add it to the set and it would still be linearly independant.

We can keep adding elements to our linearly independant set in this way by picking up some vector that is not spanned by it and adding it. This process can go on indefinitely as well. For the moment let us look at the case where this process stops after finite number of steps. Now we have a set that is linearly independant and it also spans the entire space.

An example would be to look at $\mathbb{C}$ over $\mathbb{R}$. Start with 3. The linear span of this is just elements of the form $3c$ where c is a real number. Hence it does not span elements like $2+i$. Hence we can add $2+i$ to this set and still have a linearly independant set. Now this set $\{3, 2+i\}$ is linearly independant and also spans the entire space. Any complex number $a+ib$ is equal to $b(2+i) + \frac{a-2b}{3}3$.

Such a set that spans the space and also is linearly independant is called a *basis* of the vector space V over K. And every vector in the vector space can be expressed as a linear combination of the basis elements, and uniquely so.

The number of basis elements is called the dimension of the vector space. But wait, how do we know that every basis will have the same number of elements? Is it possible that I can find three complex numbers that are linearly independant over $\mathbb{R}$ and span $\mathbb{C}$? The answer is no. It is not so hard to see that all basis must have the same number of elements. Thus the dimension of the vector space is independant of the choice of basis is hence well-defined.

The vector space $K[X]$ over K has infinite dimension and its basis could be chosen as $\{1, X, X^2, \cdots\}$. And a polynomial, say $80 + 2X + 0X^3 + 3X^4$ can be represented by the tuple $(80, 2, 0, 3, 0, 0, 0, \cdots)$ and every polynomial has a corresponding tuple.

Suppose we choose $\{1, i\}$ as a basis for $\mathbb{C}$ over $\mathbb{R}$, then every element $a + bi$ can be expressed as $a(1) + b(i)$. Now we can represent the number $a + bi$ as the tuple (a, b).

So essentially, a vector space V over K is one where each element in it can be represented as a tuple, whose entries come from K. The arity of the tuple is the dimension of the vector space.

Thus, in the finite setting, if V is a finite dimensional (say d-dimensional) vector space over a finite field K, then the number of elements of V is $|K|^d$. This is clear since it is just the number of d-tuples whose entries come from K.

5.2 Field Extensions

Let E/K be a field extension. This just means that both E and K are fields and that E contains K.

Now observe that for all $\alpha, \beta \in K$ and $u, v \in E$, $\alpha(u+v) = \alpha u + \alpha v$ and $(\alpha+\beta)u = \alpha u + \beta u$ etc. Thus all the conditions to call this a vector space hold. Thus, we can think of E as a vector space over K.

An example of this we have seen already. $\mathbb{C}$ is a field that contains $\mathbb{R}$. And $\mathbb{C}$ actually is a vector space over $\mathbb{R}$. Another example would be to look at

$$\mathbb{Q}[\sqrt{2}] = \left\{ a + b\sqrt{2} \ : \ a, b \in \mathbb{Q} \right\}$$

It is easy to check that this is a field and this clearly contains $\mathbb{Q}$. And this also naturally forms a vector space over $\mathbb{Q}$.

Definition 10. *The dimension of E as a vector space over K is called the degree of the extension E/K. This is denoted by $[E : K]$.*

$\mathbb{C}$ over $\mathbb{R}$ is a 2-dimensional extension. $\mathbb{R}$ over $\mathbb{Q}$ is an infinite dimensional extension. $\mathbb{Q}[\sqrt{2}]$ over $\mathbb{Q}$ is a 2 dimensional extension.

Adjoining Elements: An informal discussion

The field $\mathbb{C}$ is just taking $\mathbb{R}$ and adding the element i to it. Once we add i to $\mathbb{R}$, we just take all possible linear combinations, products, inverses to make it a field. We let the set $\mathbb{R} \cup i$ grow into the smallest field containing $\mathbb{R}$ and i. This is formally referred to as $\mathbb{R}(i)$, the field got by adjoining i to $\mathbb{R}$.

It is easy to check that $\mathbb{Q}(\sqrt{2})$ is infact $\mathbb{Q}[\sqrt{2}] = \left\{ a + b\sqrt{2} \ : \ a, b \in Q \right\}$. And similarly one can also check that $\mathbb{Q}(\sqrt[3]{2}) = \left\{ a + b\sqrt[3]{2} + c\sqrt[3]{2^2} \ : \ a, b, c \in \mathbb{Q} \right\}$. From this is it easily seen that $\mathbb{Q}(\sqrt[3]{2})$ is a degree 3 extension over $\mathbb{Q}$.

Given such an adjointed field extension, is it easy to find out the degree? The answer is yes. All we need to do is choose an easy basis for the vector space. For example, let us look again at $\mathbb{Q}(\sqrt[3]{2})$. Let $\alpha = \sqrt[3]{2}$. We want the degree of the extension $\mathbb{Q}(\alpha)/\mathbb{Q}$. Now consider the set $\left\{1, \alpha, \alpha^2, \alpha^3, \cdots ...\right\}$. When does this fail to be a linearly independant subset? We know that $\alpha^3 - 2 = 0$ and hence it loses its linear independance after α^3. This is because α was a root of $X^3 - 2$, a degree 3 polynomial over the $\mathbb{Q}$.

Instead if we were to look any α, any equation of linear dependance would look like $a_0 + a_1\alpha + a_2\alpha^2 + \cdots a_k\alpha^k = 0$ and this would just mean that α is a root of the polynomial $a_0 + a_1X + a_2X^2 + \cdots a_kX^k = 0$. Thus,

the degree of such an extension $\mathbb{Q}(\alpha)/\mathbb{Q}$ is just the degree of the smallest degree polynomial of which α is a root.

$\mathbb{C} = \mathbb{R}(i)$ and i has $X^2 + 1$ as its minimum polynomial and thus $[\mathbb{C} : \mathbb{R}] = 2$. If we were to look at $\mathbb{Q}(\pi)$, π is not a root of any polynomial with coefficients in $\mathbb{Q}$ (this is also referred as 'π is transcendental'). Thus the set $\left\{1, \pi, \pi^2, \cdots\right\}$ would be an infinite linearly independant subset. And hence the extension $\mathbb{Q}(\pi)$ over $\mathbb{Q}$ is of infinite degree.

We shall look at more properties of finite fields and extensions next time.

CS681	**Computational Number Theory**
Lecture 8: More on Finite Fields	
Instructor: Piyush P Kurur	*Scribe: Ramprasad Saptharishi*

We will be spending some time on understanding the structure of fields of finitely many elements. In this class we shall some necessary properties that finite fields (or any field in general) should hold.

6 Characteristic of Finite Fields

Looking at the examples of fields that we know of, we clearly see that $\mathbb{Q}$ and $\mathbb{Z}/p\mathbb{Z}$ are different. Apart from just the cardinality properties, $\mathbb{Z}/p\mathbb{Z}$ has this property of k and $p-k$ cancelling off. Using this as a motivation, let us define what a characteristic of a field is. First we shall state it formally and then look at a better interpretation of it.

Definition 11. *For any ring R, there exists the identity element 1. Consider the homomorphism*

$$\begin{array}{rcl} \phi : \mathbb{Z} & \longrightarrow & K \\ n & \mapsto & n \cdot 1 \end{array}$$

where $n \cdot 1$ simple means adding 1, in R, n times. And $\mathbb{Z}$ being a PID, the kernel of this map will be of the form $m\mathbb{Z}$. This number m is called the characteristic of the field.

In other words, the characteristic is the smallest number m such that m times the identity element is zero.

However, it is possible that $m \cdot 1$ will never be zero. For example, take the rings like $\mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{Q}[x]$ etc. The corresponding map will hence have a trivial kernel and that is the ideal $0\mathbb{Z}$. Hence the characteristics of these rings is 0 and not infinity. This is just the language. Infact, we shall refer characteristic 0 rings as rings of infinite characteristic.

Here is a trivial lemma.

Lemma 8. *Let R be a ring (with identity) of characteristic m and S be a ring that contains R. Then characteristic of S is also m.*

Proof. R has characteristic m implies that $\phi : \mathbb{Z} \longrightarrow R$ has the kernel as $m\mathbb{Z}$. The homomorphism works just on the identity element of R and hence would be exactly the same on S (since S has to share its identity element with R). Thus, since the homomorphism is the same, the kernel has to be the same. □

6.1 Characteristic of Fields

Now, suppose m is the characteristic of any ring R. Then by definition the kernel of the map ϕ is $m\mathbb{Z}$. And by the isomorphism theorem, we know that the following map is injective:

$$\hat{\phi} : \mathbb{Z}/m\mathbb{Z} \longrightarrow R$$

And therefore, in a way, a copy of $\mathbb{Z}/m\mathbb{Z}$ is sitting inside R. Thus, $\mathbb{Z}/m\mathbb{Z}$ is a subring of R.

Now let us look at the characteristic of fields instead of rings. Let us take the identity element and just keep adding it. Either, for some m we have $m \cdot 1 = 0$ or it just keeps going on. If it becomes 0, we know that the field has characteristic m. The other case is the characteristic 0 case.

Now, can it be possible that m is composite? Suppose $m = pq$ where both $p, q < m$. Since we know that $m \cdot 1 = 0$, this means that $(p \cdot 1)(q \cdot 1) = pq \cdot 1 = 0$. And by our assumption, we know that neither $p \cdot 1$ nor $q \cdot 1$ is zero; we just showed the existence of zero divisors in a field! That is not possible. Hence summarizing as a theorem:

Theorem 9. *Any field F must either have 0 characteristic or a characteristic that is a prime.*

Let us pick any field F whose characteristic is a prime p. We know that if we let 1 'generate' a subfield of its own by just adding itself, it would get to $\mathbb{Z}/p\mathbb{Z}$. Thus for any field of prime characteristic, it should contain $\mathbb{Z}/p\mathbb{Z}$. We shall refer to $\mathbb{Z}/p\mathbb{Z}$ by $\mathbb{F}_p$.

For a field of infinite characteristic (characteristic 0, just language), 1 would keep being added on without ever giving a 0. Thus it would generate the entire set of positive integers. And since additive inverses should exist, the negative integers should also belong to the field. And further, because of the multiplicative inverses, all rational numbers should exist. Hence, every field either contains $\mathbb{F}_p$ or $\mathbb{Q}$.

Please keep in mind that finite characteristic does not mean finite cardinality. As a counter example, look at the following set:

$$\mathbb{F}_p(X) = \left\{ \frac{f(X)}{g(X)} \ : \ f, g \in \mathbb{F}_p[X], g \neq 0 \right\}$$

that the set of rational functions over one variable. This field has infinite cardinality and since it contains $\mathbb{F}_p$ has characteristic p.

7 Order of Finite Fields

Now let us take any finite field K. Then this field must have characteristic that is not zero. Why? Since if it did have characteristic zero, it would contain $\mathbb{Q}$ and hence be infinite.

Since the characteristic of this field is finite, say p, it contains $\mathbb{F}_p$. Recall that if K is a field that contains another field F (in our case $\mathbb{F}_p$), then K is an extension of F.

Thus, this tells us that any field of characteristic p is a vector space over $\mathbb{F}_p$. Since we now have a vector space, we can talk of the dimension of this vector space. The dimension cannot be infinite. Why? For it it was, then the basis, which belongs to K, would be an infinite set. And this is an obvious contradiction since we assumed that K was finite.

Thus, the dimension of K over $\mathbb{F}_p$ is finite, say s. And since every element of K can be written as a s-tuple of elements of $\mathbb{F}_p$, this means that the number of elements of K is p^s.

And, from our earlier theorem, we know that the characteristic of a finite field has to be a prime. Hence the order of any finite field has to be a power of a prime.

Theorem 10. *Any finite field has p^s elements where p is a prime and s a positive integer.*

The moment we make such a statement, we have two questions in mind.

- Existence: For every prime p and positive integer s, do we have a field of p^s elements?
- Uniqueness: Can we have two different fields with p^s elements?

We shall soon see that the answer to the first question is yes and the second is no. There is exactly one field of size p^s.

7.1 Creating Extensions of $\mathbb{F}_p$

The general question is how to obtain extension fields of a given field. As a motivation, let us look at $\mathbb{R}$. How do we get an extension field of $\mathbb{R}$? In a sense, we need to increase our domain. Therefore, we need to find elements that don't belong to $\mathbb{R}$. The way to do that is to look at roots of polynomials. There is no root of the polynomial $X^2 + 1$ in $\mathbb{R}$. Hence, just add the root. This is formally done by quotienting.

But there is a small catch before we do that. We now know that we have the complex number i is a root of $X^2 + 1$ but it is of course the root of $(X^2+1)(X^{213} - 3X^{127} + 897)$ as well. This is where the concept of minimal polynomial comes in.

Definition 12. *Let L be an extension of K. For any $\alpha \in L$, the mimimum polynomial of α over K is the monic polynomial of smallest degree over K that has α as a root.*

Again, the questions of existence and uniqueness comes in. Does every α have a minimum polynomial? The answer is no, and the example is π over $\mathbb{Q}$. It doesn't make sense to talk of a minimum polynomial when the number is transcendental.

But suppose the number is not transcendental, that it is a root of some polynomial. Then do we have a unique minimum polynomial? Yes. Since if f and g are two polynomials of smallest degree that has α as a root, then so does $gcd(f, g)$ and the gcd clearly has smaller degree. This then contradicts that f and g had least degree. Thus the minimum polynomial is unique.

And by the same reason, the minimum polynomial must be irreducible. For if f was the minimum polynomial of α and if $f(X) = g(X)h(X)$ then α must be a root of either g or h and their degree is strictly less than f. Thus the minimum polynomial if indeed irreducible.

Another way is the following. We have some $\alpha \in L$ and we want the minimum polynomial of α. Consider the following homomorphism.

$$\begin{array}{rccc} \text{eval}_\alpha : K[X] & \longrightarrow & L \\ f(X) & \mapsto & f(\alpha) \end{array}$$

The map is called the evaluation map since it just evaluates every polynomial at α. The kernel of this map will be an ideal of $K[X]$, a principle ideal. The generator of this ideal is the minimum polynomial.

Here is a way to create extensions. Look at $\mathbb{F}_p[X]$ and take some irreducible polynomial f of it. Then we know that $\mathbb{F}_p[X]/(f)$ is a field since the ideal (f) is maximal as f is irreducible. If the degree of f is d, then this would give is a field of size p^d.

We shall see this in more detail next time.

CS681	Computational Number Theory

Lecture 9: Uniqueness of $\mathbb{F}_{p^m}$

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Last lecture we closed with two questions of existence and uniqueness of fields of order p^m. This lecture, we shall the question of uniqueness and also understand some other properties of extensions.

8 Some More Properties of Field Extensions

Firsly, we need to understand what quotienting means. Suppose we have a field K and we look at the polynomial ring over this field - $K[X]$. Let us take some irreducible polynomial f and look at $K[X]/(f(X))$. What does this mean?

Quotienting means that you consider all occurences of the ideal $(f(X))$ as zero. So in particular, if you look at $f(X)$ in this ring, it would be zero. Hence the variable X can now be thought of as a root of f.

Infact, this is exactly what we do to adjoin roots. Suppose we have a field K and we need to adjoin some element α. We take the minimum polynomial f of α and we look at $K[X]/(f(X))$. Here, essentially, X is α.

This is how we built $\mathbb{C}$. We looked at $\mathbb{R}[X]$ and quotiented it with the ideal generated by $X^2 + 1$. And now note that there is no real distinction between $-i$ or i. Algebraically, $\mathbb{R}(i) \cong \mathbb{R}(-i) = \mathbb{C}$. So the general point to note is that if α and β are both roots of the same irreducible polynomial f, then $K(\alpha) \cong K(\beta) \cong K[X]/(f(X))$.

8.1 Splitting Fields

Until we knew that an object called $\mathbb{C}$ existed, we had no idea if there was an element i such that $i^2 = -1$. So as such, it doesn't make sense to adjoin some element until you know where it is from. When you look at $K[X]/(f(X))$, we said that X can be identified with a root of f, but what root? Where is this root? I know it's not in K (for if it were, $X - \alpha$ is a factor of f and hence can't be irreducible), but where else is it?

This is where splitting fields come in.

Definition 13. *A splitting field of a polynomial f over K is the smallest field extension E/K that contains all the roots of f.*

Or equivalently, it's the smallest field E where the polynomial f factorizes into linear factors.

The first thing to note is that this is not equivalent to adjoining a root. To illustrate the difference, take the field $\mathbb{Q}$ and let us look at the polynomial $X^3 - 2$. This polynomial is irreducible and hence we can talk of the field $\mathbb{Q}[X]/(X^3 - 2)$ and identify the element X with a root of the polynomial say $\sqrt[3]{2}$.

But note that this is *not* the splitting field. The polynomial has other roots, namely $\sqrt[3]{2}\omega$, $\sqrt[3]{2}\omega^2$ where ω is a primitive cube root of unity. And it is clear that $\mathbb{Q}(\sqrt[3]{2})$ is a subfield of the reals and obviously cannot contain the complex number $\sqrt[3]{2}\omega$ and hence cannot be the splitting field of the polynomial.

The splitting field of this polynomial is $\mathbb{Q}(\sqrt[3]{2}, \omega)$ and is a degree 6 extension over $\mathbb{Q}$ whereas the extension by adjoining roots are degree 3 extensions over $\mathbb{Q}$.

An important theorem is the following:

Theorem 11. *If E and E' are two splitting fields of a polynomial $f(X)$ over K, then $E \cong E'$.*

We omit the proof, but nevertheless the theorem is important and will be used. Interested readers can refer any abstract algebra books for the proof of this fact.

8.2 The Frobenius Map

The binomial theorem takes a very simple form over $\mathbb{F}_p$. We know that

$$(X + Y)^p = \sum_{i=0}^{p} \binom{p}{i} x^i y^{p-i}$$

Lemma 12. *For $i \neq 0, p$, the coefficient $\binom{p}{i}$ is divisible by p.*

Proof. The proof is quite obvious. The coeffient is

$$\binom{p}{i} = \frac{p \cdot (p-1) \cdots (p-i+1)}{1 \cdot 2 \cdots i}$$

and the numerator has a factor of p and the denominator does not. And since this si an integer, the factor of p will remain uncancelled and hence will be divisible by p. □

This then tells us that in the field $\mathbb{F}_p$,

$$(X+Y)^p = X^p + Y^p$$

If you look at it as a automorphism (an isomorphism of $\mathbb{F}_p$ into itself), this shows the map $x \mapsto x^p$ is an automorphism of $\mathbb{F}_p$.

This map $\phi(x) = x^p$ is called the Frobenius map. It is a very important automorphism of finite fields and appears almost everywhere.

8.3 The Multiplicative Group of a Finite Field is Cyclic

Let K be a finite field of p^m elements. We shall show now that the multiplicative group $K \setminus \{0\}$ is a cyclic group.

Infact, we shall show a much stronger theorem.

Theorem 13. *Let K be any field (not necessarily finite) and let A be any finite subgroup of $K^\star = K \setminus \{0\}$. Then the subgroup A is cyclic. (That is, there exists an element $a \in A$ such that every other element in A is a power of a)*

Proof. Let $|A| = M$. We need to show that there exists an element in A of order M. Suppose not, let a be the element of highest order in A. Let the order of a be $m < M$.

Now pick any $x \in A$. We claim that the order of x must divide M. To prove this, let the order of x be n. Let the gcd of m and n be d.

By Euclid's lemma, there exists integers p, q such that $pn+qm = d$. Thus, let us look at the element $g = x^p a^q$. Then $g^m = x^{pm} a^{qm} = a^{qm} = d$ since $qm = d \bmod n$. And similarly $g^n = x^d$. Thus, the order of g is infact equal to $\frac{mn}{d} = lcm(m,n)$. And since we assumed that a was an element of maximum order, the lcm of m and n has to be m and therefore n has to divide m.

Having established this, we see that the order of every element must divide m and therefore $x^m = 1$ for every $x \in A$. And therefore, the polynomial $X^m - 1$ has every element of A as a root. But if we were to assume that $m < M$, then a polynomial of degree m would have $M > m$ roots! And this clearly cannot happen in a field. And therefore, $m = M$ and hence the group A is cyclic. □

9 Uniqueness of $\mathbb{F}_{p^m}$

Now we come to the proof that every field of p^m elements are isomorphic. So essentially, there is only 1 field of size p^m and hence would make sense

to refer to *the* field of size p^m as $\mathbb{F}_{p^m}$.To avoid cluttering of subscripts and superscripts, let $q = p^m$. We need to show that any two fields of size q are isomorphic.

Now let F be any field of order q. Then every element $a \in F$ satisfies the property that $a^q = a$. Hence in particular, every element of the field F is a root of $X^q - X$. And therefore, considering this polynomial over $\mathbb{F}_p$, F is a splitting field of $X^q - X$ over $\mathbb{F}_p$. And by a theorem stated earlier, all splitting fields of a polynomial are isomorphic. And therefore, any two fields of order q are isomorphic.

10 More about $X^q - X$

Here is a very important property of this polynomial $X^q - X$.

Theorem 14. *Let $Irr(K, d)$ be the set of all irreducible polynomials of degree d over K. Then, the following equation holds in $\mathbb{F}_p[X]$:*

$$X^{p^m} - X = \prod_{\substack{f \in Irr(\mathbb{F}_p, d) \\ d|m}} f(X)$$

Proof. We shall show that the LHS is equal to the RHS by comparing the roots on both sides. For the roots to first exist, we shall go to some large field.

Firstly, note the polynomial $X^q - X$ is satisfied by every element of $\mathbb{F}_q$. And by the same degree argument, the roots of the polynomial is precisely all the elements of $\mathbb{F}_q$. We shall first show that every element $\alpha \in \mathbb{F}_q$ is also a root of the RHS.

Since $\alpha \in \mathbb{F}_q$, pick up the minimum polynomial $f(X)$ of α over $\mathbb{F}_p$. We have seen earlier that this polynomial must be irreducible. Hence $\mathbb{F}_p[X]/(f(X))$ corresponds to the field $\mathbb{F}_p(\alpha)$. Let the degree of f be d.

We know that

$$\begin{aligned} [\mathbb{F}_q : \mathbb{F}_p] &= [\mathbb{F}_q : \mathbb{F}_p(\alpha)]\,[\mathbb{F}_p(\alpha) : \mathbb{F}_p] \\ \implies m &= [\mathbb{F}_q : \mathbb{F}_p(\alpha)] \cdot d \\ \implies d \mid m & \end{aligned}$$

And hence, since the degree of this irreducible polynomial divides m, it appears in the product of the RHS as well.

The other way is easy too. Pick up any factor $f(X)$ on the RHS. Let its degree be d. Let α be one of the roots. Then we know that $\mathbb{F}_p(\alpha)$ must be the field $\mathbb{F}_{p^d}$. But since d divides m, this is a subfield of $\mathbb{F}_q$. And therefore every element of $\mathbb{F}_{p^d}$, in particular α, must be in $\mathbb{F}_q$ as well. □

10.1 Extracting Factors

The formula outlined in the previous section is extremely useful in factoring. It helps us pull out factors of the same degree. We shall see a quick sketch here and discuss this in detail next class.

Let us say we have some polynomial f over $\mathbb{F}_p$. How do we extract all linear factors over $\mathbb{F}_p$? The idea is pretty simple. We know that $X^p - X$ splits as linear factors over $\mathbb{F}_p$. And more over, the formal derivative of it is $pX^{p-1} - 1 = -1 \neq 0$ and therefore it has no repeated roots as well.

Then, we have

$$g(X) = gcd(f, X^p - X)$$

Then if $\alpha \in \mathbb{F}_p$ was any root of f, then clearly since α also satisfies $X^p - X$, α will be a root of g as well. And more importantly, since $X^p - X$ is has no repeated roots, g will retain the same property as well. Hence every root of f over $\mathbb{F}_p$ appears exactly once in g.

Having removed all degree 1 factors, we can all extract degree 2 factors by taking the gcd with $X^{p^2} - X$.

We shall discuss this idea of factoring in detail soon.

CS681 **Computational Number Theory**

Lecture 10: Distinct Degree Factoring

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Last class we left of with a glimpse into distant degree factorization. This class, we shall look at the details of it and also how it can be used as an irreducibility test.

11 DDF: The Problem

We are given a polynomial f over a finite field $\mathbb{F}_p$ of degree say n. We want to factor them in to degree 1 factors, degree 2 factors etc. To make this more explicit, let us assume that f factorizes as

$$f = g_{11}g_{12}\cdots g_{1m_1}g_{21}\cdots g_{dm_d}$$

where each g_{ij} is an irreducible factor of f of degree i. Hence given f, we want to return $\prod_j g_{ij}$ for each i. That is, return the factor of f that is the product of all degree i irreducible factors of f. And we want to do this for all i. This is called *distinct degree factorization.* We want an efficient algorithm for this.

But before we start thinking of algorithms, what do we mean by efficient? It is the usual 'polynomial time in input length' but what is the input length? We are looking at $f(X) = \mathbb{F}_p[X]$ and each coeffiecient of the polynomial is from $\mathbb{F}_p$. And since $\mathbb{F}_p$ contains just p elements, we can encode them in binary using $\log p$ bits. Hence the input size is about $n \log p$. Thus we are interested in algorithms that have running time of $(n \log p)^c$ for some constant c.

12 Extracting Square-free Parts

When we said that f factorizes as $g_{11}\cdots g_{dm_d}$, it is very much possible that there are some $gij = g_{ik}$ or in other words the square of g_{ij} divides f. The first step of any factoring algorithm is to remove such multiple factors and

make f *square free* (make sure that f is not divisible by any square).

Suppose we were looking at polynomials over $\mathbb{Z}[X]$ or something. Then we have a very nice way of checking for such multiple factors. We know that if f has a repeated root α , then the derivative f' has α as a root as well. Infact, if $(x-\alpha)^m$ divided f then $(x-\alpha)^{m-1}$ will divide f'. And hence, the gcd of f and f' will have $(x-\alpha)^{m-1}$ as a factor. Thus, we can divide f by this gcd and get rid of higher multiplicity terms.

But in the case of $\mathbb{Z}[X]$, we had an interpretation of real numbers, limits and hence we could define what a derivative is. But when it comes to finite fields, what does differentiation mean? How can we use this technique to get extract the square-free part?

Note that we don't need the notion of a derivative as a tangent or something. That is where the limits come in. What we need is a condition where if some g^m divides f then g^{m-1} should divide f' for the f' that we are going to define. Thus, since we are just in the realms of polynomials, we shall use the rules of differentiation as just a formula.

Thus let D be a map that sends X^m to mX^{m-1}. Now extend this linearly to all polynomials to get

$$D\left(a_0 + a_1X + a_2X^2 + \cdots a_mX^m\right) = a_1 + 2a_2X + \cdots ma_mX^{m-1}$$

Now, we leave the following things to be proven as an exercise.

Exercise

1. $D(f+g) = D(f) + D(g)$
2. $D(fg) = fD(g) + gD(f)$
3. $D(f^m) = mf^{m-1}D(f)$
4. **Theorem:** If h is a factor of f such that $h^m \mid f$ then $h^{m-1} \mid g$. And further, if h^m is the highest power of h that divides f then h^{m-1} is the highest power of h that divides f'.

Once we have these properties, we can extract the square-free part of f easily. Given an f construct the formal derivative f'. Let $g = gcd(f, f')$. The polynomial f/g consists is the square free extraction of f.

There is, however, a small catch here. But we shall get back to it in the end of the class and discuss it in the next class in detail. For the moment, let us proceed with distinct degree factorization.

13 Distinct Degree Factorization

Given $f \in \mathbb{F}_p[X]$ of degree n, we want to get the distant degree factorization of f. That is, we want to get $g_1, g_2, \cdots, g_n$ such that $f = g_1 g_2 \cdots g_n$ where each g is the product of all irreducible factors of f of degree i. And we want to do this efficiently.

The key idea is the formula we proved last class:

$$X^{p^m} - X = \prod_{\substack{f \in \text{Irr}(\mathbb{F}_p, d) \\ d|m}} f(X)$$

Firstly, let us look at the case when $m = 1$. Then $X^p - 1$ is the product of all irreducible polynomials over $\mathbb{F}_p$ of degree 1. And thus, in particular, it contains the degree 1 irreducible factors of f within it.

Thus, $g_1 = gcd(f, X^p - X)$ will be the product of all degree 1 irreducible factors of f as $X^p - X$ has just degree 1 irreducible factors and all those that divide f will be a part of the gcd. Thus, we have obtained the required g_1. Now, call $f_2 = f/g_1$ and now let $g_2 = gcd(f_2, X^{p^2} - X)$ and this will have precisely all degree 2 irreducible factors of f (degree 1 factors won't appear since we have removed all degree 1 factors by dividing by g_1). Thus, we can repeat this procedure.

This is a naive algorithm and has a pretty serious problem. The trouble is that the algorithm would have to compute $gcd(f_i, X^{p^i} - X)$ which is a HUGE degree polynomial. We want our running time as $(n \log p)^c$ but this naive algorithm takes about $O(p)$ time! And this is definitely not acceptable since even finding all linear irreducible factors might take $O(p)$ time! We definitely need to change this. But the polynomial $X^{p^i} - X$ is a nice polynomial and hence allows a nice simple trick to make the algorithm polynomial time.

Let us look at the gcd algorithm. The first step is to compute $X^{p^i} - X \bmod f_i$ and the problem with this is that the degree of the polynomial is too large to apply the naive algorithm. But we can do far better in the case of this special polynomial.

Note that $X^{p^i} - X \bmod f_i = X^{p^i} \bmod f_i - X \bmod f_i$ and the difficulty was in computing $X^{p^i} \bmod f_i$. This can be done quite efficiently using the technique of repeated squaring.

13.1 Repeated Squaring

In general, we have a polynomial f of degree n and we want to calculate $X^M \bmod f$ in time $(n \log M)^c$ for some constant c. The idea is pretty simple.

Firstly assume M to be a power of 2. Then we can do the following. Start with X, and square it. And square it again and so on. The moment the degree goes beyond n, take it modulo f to reduce it's degree back to less than n. And continue this squaring process, for $\log M$ steps. Thus, we would have computed $X^M \bmod f$ in time $(n \log M)^c$.

Now what do we do for M that is not a power of two? The answer is simple again. Just look at the binary representation of M. Say $m = b_0 + b_1 2 + b_2 2^2 + \cdots b_l 2^l$. Then

$$X^M = X^{b_0} (X^{b_1})^{2^1} (X^{b_2})^{2^2} \cdots (X^{b_l})^{2^l}$$

And since each b_i is either 1 or 0, X^{b_i} is either 1 or X. Thus, we can just compute $X^{2^i} \bmod f$ for $0 \leq i \leq \log M$ and multiply all those residues that have b_i as 1.

Algorithm 2 EVALUATE $X^M \bmod f$ USING REPEATED SQUARING

1: Let $M = b_0 + b_1 2 + \cdots b_l 2^l$, the binary representation of M, where $l = \lfloor \log M \rfloor$.
2: $g_0 = 1$ and $g = b_0 X$.
3: **for** $i = 1$ to l **do**
4: $\quad g_i = (g_{i-1})^2 \bmod f$
5: $\quad$ **if** $b_i = 1$ **then**
6: $\quad\quad g = g \cdot g_i \bmod f$
7: $\quad$ **end if**
8: **end for**
9: **return** g

Now, with this repeated squaring algorithm, we can do distinct degree factorization. The algorithm is given at the end of the lecture.

14 A Catch and a Hint

First let us get to the catch that we had mentioned earlier. Our method for extracting the square free part of f involved taking the formal derivative and then the gcd of f with f'. But strange things can happen in a finite field. For example, let us take the polynomial $f(X) = X^{2p} - 3X^p + 5$. The formal derivative is $2pX^{2p-1} - 3pX^{p-1}$ which is 0 in $\mathbb{F}_p$! What do we do now?

It is easy to see that the derivative can become 0 if the only surviving terms of the polynomial have degree that's a multiple of a prime. Then this polynomial $f(X)$ can be thought of as $g(X^p)$ and we can go on to do our algorithm for g instead of f. In our example, $g(X) = X^2 - 3X + 5$ which is well behaved. All we need to do is get the factors of g and replace every occurance of X by X^p. And more over, since we are in $\mathbb{F}_p$, it is easy to check that $f(X) = g(X^p) = (g(X))^p$.

So much for the small catch. Now here is a hint. The algorithm for distinct degree factorization immediately gives us an algorithm to test if a given polynomial over $\mathbb{F}_p[X]$ is irreducible. The reduction is very straightforward and the students are encouraged to think about it. We shall discuss this in the next class.

Algorithm 3 DISTINCT DEGREE FACTORIZATION

Input: $f(X) \in \mathbb{F}_p[X]$ of degree n.

1: $f_0 = f$.
2: **for** $i = 1$ to n **do**
3: Using repeated squaring, compute $s_i = X^{p^i} \bmod f_{i-1}$
4: Compute $g_i = gcd(f_{i-1}, s_i - X)$.
5: $f_i = f_{i-1}/g_i$
6: **end for**
7: **return** $\{g_1, g_2, \cdots, g_n\}$.

CS681	Computational Number Theory

Lecture 11: Cantor-Zassenhaus Algorithm

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

In this class, we shall look at the Cantor-Zassenhaus randomized algorithm for factoring polynomials over $\mathbb{F}_p$. We shall do it for the case when $p \neq 2$. The case when $p = 2$, which isn't too different from the other case, would be given as an exercise for the students to solve.

15 Irreducibility Testing

We left off last class with a hint that the distinct degree factorization infact gives a straightforward irreducibility test. Here is the explicit answer.

We are given an f and we need to check if this polynomial is irreducible or not. First check if it is square free. If it isn't, immediately reject it. Else, proceed to compute the distinct degree factors of f. If the degree of f is n, the DDF algorithm returns $g_1, g_2, \cdots, g_n$ such that each g_i is the product of irreducible factors of f of degree d.

Now, suppose f was irreducible, then clearly every $g_i = 1$ for $1 \leq i \leq n-1$ and $g_n = f$. Thus just check if the returned g_i's satisfy this condition.

15.1 Generating Irreducible Elements

Suppose are given a positive integer d, we want to efficiently find an irreducible polynomial of degree d over $\mathbb{F}_p$. Now before we get into this, why is this important?

The answer is that this is the only way we can construct the field $\mathbb{F}_{p^d}$. There are lots of applications where we need to do arithmetic over a field of large size and $\mathbb{F}_p$ would be a candidate only if the prime p is large. And finding such a large prime is hard and inefficient.

Instead, we pick a small prime p and try and find an irreducible polynomial of degree d. Once we do that, we have $\mathbb{F}_p[X]/(f(X))$ which is isomorphic to $\mathbb{F}_{p^d}$. This is precisely finding irreducible polynomials of a given

degree is very useful.

To generate an irreducible polynomial of degree d, we shall just randomly pick a polynomial of degree d. It can be argued that the probability that this polynomial is irreducible is pretty high. And since we even have a deterministic test to check if a polynomial f is irreducible, we just repeat this procedure: Pick an $f \in \mathbb{F}_p[X]$ of degree d at random and repeat this if the irreducibility test says that this polynomial is not irreducible.

All we need to do is to argue that the density of irreducible polynomials is large.

Theorem 15. *Let $I(d)$ be the number of irreducible polynomials of degree d over $\mathbb{F}_p$. Then*

$$I(d) = \frac{p^d}{d} + O\left(\sqrt{p}\right)$$

And therefore,

$$\Pr_{f \in \mathbb{F}_p[X], deg(f)=d}[f \in \text{Irr}(\mathbb{F}_p, d)] \geq \frac{\frac{p^d}{d} + O(\sqrt{p})}{p^d} \geq \frac{1}{d}$$

As for the proof of the theorem, here is a sketch of it.

Proof. (sketch) We know that

$$X^{p^m} - X = \prod_{\substack{f \in \text{Irr}(\mathbb{F}_p, d) \\ d|m}} f(X)$$

Comparing the degrees on both sides,

$$p^m = \sum_{d|m} I(d) \cdot d$$

Equations of this kind can be inverted using the *Möbius Inversion.*

Lemma 16 (Möbius Inversion)**.** *If we have any equation of the form*

$$f(m) = \sum_{d|m} g(d)$$

then

$$g(m) = \sum_{d|m} \mu(d) f(m/d)$$

Exercise: Read up on the Möbius Inversion.

With the inversion formula, once can complete the proof of the theorem by taking $f(m) = p^m$ and $g(m) = I(m) \cdot m$. □

16 The Cantor-Zassenhaus Algorithm

Now we get to factoring a polynomial over $\mathbb{F}_p$. Given a polynomial of degree f over $\mathbb{F}_p$, it is enough to get one non-trivial[1] factor of f.

As we said in the last few lectures, the first thing to do is to check if f is square free. If it isn't we can just return the square-free part of f as a factor and be done. If it is square-free, we compute the distinct degree factorization of f. If f turns out to be irreducible, we just return "irreducible." Else, we have to proceed to find a non-trivial factor of f.

We shall factor each g_i returned by the DDF algorithm separately. Hence, we now assume that we have an $f \in \mathbb{F}_p[X] = g_1 \cdots g_m$ such that each g_i is irreducible, distinct and of the same degree d.

Here enters our old friend Chinese Remaindering. Since $f = g_1 \cdots g_m$, we know that

$$\mathbb{F}_p[X]/(f(X)) \cong \mathbb{F}_p[X]/(g_1(X)) \times \cdots \times \mathbb{F}_p[X]/(g_m(X))$$

Now note that each g_i is an irreducible polynomial of degree d. And therefore, $\mathbb{F}_p[X]/(g_i(X))$ is isomorphic to $\mathbb{F}_{p^d}$. Hence the product just looks like

$$\mathbb{F}_p[X]/(f(X)) \cong \mathbb{F}_{p^d} \times \cdots \times \mathbb{F}_{p^d}$$

And further, we know that

$$(\mathbb{F}_p[X]/(f(X)))^\star \cong \mathbb{F}_{p^d}^\star \times \cdots \mathbb{F}_{p^d}^\star$$

Now what do zero divisors, say g, in $\mathbb{F}_p[X]/(f)$ look like? When you take the image under the chinese remaindering, it should go to some tuple $(a_1, a_2, \cdots, a_m)$ where some $a_i = 0$. Further, if this zero divisor is non-trivial (0 is a trivial zero-divisor, useless), some other $a_j \neq 0$. What does this mean? g has a 0 in coordinate i which means that g is divisible by g_i, and hence $g \neq 1$. And also, g is non-zero at coordinate j and therefore g_j does not divide g and hence $g \neq f$. Thus, $gcd(g, f)$ is certainly not f nor 1 and hence is a non-trivial factor of f.

[1] trivial factors of f are 1 and f. Factors though they may be, are useless for us.

Therefore, the problem of finding factors reduces to the problem of finding zero divisors in $\mathbb{F}_p[X]/(f(X))$.

16.1 Finding Zero-Divisors

The idea is the following. We cross our fingers and pick a polynomial $a(X)$ of degree less than n at random. This is some element from $\mathbb{F}_p[X]/(f(X))$. If we are extremely lucky we might just get $gcd(a, f) \neq 1$, and this already gives us a non-trivial factor of f and we are done. Hence, lets assume that a is not a zero-divisor of the ring. And therefor, a must be an element of $(\mathbb{F}_p[X]/(f))^\star$, an invertible element.

Note that since we do not know the factors g_i, we do not know the chinese remainder map. We just know that a map exists, we don't know how to compute it. But suppose someone secretly told us that one of the coordinates of a under the chinese remainder map is -1, then what can we do?

Look at the images of $a(X) + 1$. The images of this is just 1 added to every coordinate of the image of $a(X)$. And since someone told us that one of the coordinates was -1, that coordinate in the image of $a(X)+1$ must be zero! Which means that, $a(X) + 1$ is a zero divisor. However it is possible that all the coordinates is -1 and that would just make $a(X) + 1 = 0$ which is useless.

Now, how do we make sure that there is some -1 in one of the coordinates and not everywhere? Use the fact that each element of the product is $\mathbb{F}_{p^d}$. We know that $\mathbb{F}^\star_{p^d}$ is an abelian group of order $p^d - 1$. And therefore, for every element b in this group, $b^{p^d-1} = 1$.

We need a -1, and therefore we look at the square-root of it. Since we know that $b^{p^d-1} = 1$, $b^{(p^d-1)/2} = \sqrt{1}$ which can either be 1 or -1.[2] Let us just call $(p^d - 1)/2 = M$.

Thus, we have a simple procedure. Pick up some random polynomial $a(X)$ of degree less than n. Check if you are lucky by computing $gcd(a, f)$ and checking if it is 1. Else, compute $a(X)^{(p^d-1)/2} \bmod f(X)$ using repeated squaring. Now if a was mapping to $(a_1, a_2, \cdots, a_m)$, then a^M would be mapped to $(a_1^M, \cdots, a_m^M)$. And we just saw that each of a_i^M is either 1 or -1.

Claim 17. *Each $a_i^M = 1$ with probability $1/2$, and they are independent.*

[2] there can't be any other square-roots of 1. This is because square roots of -1 satisfy the equation $X^2 - 1 = 0$ and this equation can have only 2 roots in a field

Proof. Since the chinese remainder map is an isomorphism and each g_i's are distinct, they are clearly independent. To check that the probability that $b^M = 1$ is $1/2$, we look at the following map.

$$\begin{aligned} \psi : \mathbb{F}_{p^d}^\star &\longrightarrow \{1, -1\} \\ b &\longrightarrow b^M \end{aligned}$$

Note that the set $\{1, -1\}$ form a group under multiplication. Infact it can be identified with the group $\mathbb{Z}/2\mathbb{Z}$.

Exercise: Prove that the map ψ is indeed a group homomorphism.

And therefore, the kernel of this map ψ is a subgroup of $\mathbb{F}_{p^d}^\star$ of all elements b such that $b^m = 1$. The other coset of this kernel is the set of elements b such that $b^M = -1$. Since these two are cosets, they are of equal size. Hence a randomly chosen b will have $b^M = 1$ with probability $1/2$. □

And therefore, each a_i is 1 or -1 with probability $1/2$. Thus the probability that all the coordinates are 1 or all the coordinates are -1 is just $1/2^m$. Thus with probability atleast $1 - 2^{m-1}$, we have some vector that has 1s at certain places and -1s at the rest. Thus, now we are in the case when someone had secretly told us that some coordinate is -1.

And therefore, we can pick a random polynomial $a(X)$, raise it to the power M modulo f, and add 1 to it. With probability atleast $1 - 2^{m-1}$, this will be a zero-divisor and hence $gcd(a^M + 1, f)$ will be a non-trivial factor of f.

So here is the algorithm:

Algorithm 4 CANTOR-ZASSENHAUS ALGORITHM FOR FACTORING

Input: A polynomial $f \in \mathbb{F}_p[X]$ of degree n.

if f is not square-free **then**
  return the square-free part
end if
Compute the distinct degree factors of f. Call them $\{g_1, g_2, \cdots, g_n\}$.
if $g_n \neq 1$ **then**
  return IRREDUCIBLE
end if
for all $g_i \neq 1$ **do**
  EqualDegreeFactorize(g_i, d)
end for

Algorithm 5 EQUALDEGREEFACTORIZE

Input: A polynomial $f \in \mathbb{F}_p[X]$ of degree n all of whose m irreducible factors are of degree d.

1: Pick a random polynomial $a(X)$ of degree less than n.
2: **if** $gcd(a, f) \neq 1$ **then**
3: **return** $gcd(a, f)$
4: **end if**
5: Let $M = (p^d - 1)/2$.
6: Using repeated squaring, compute $a'(X) = a(X)^M + 1$.
7: **if** $a' \neq 0$ and $gcd(a', f) \neq 1$ **then**
8: **return** $gcd(a', f)$ {Happens with probability atleast $1 - 2^{m-1}$}
9: **end if**
10: Repeat algorithm with a different choice for a.

CS681	Computational Number Theory

Lecture 12: Berlekamp's Algorithm

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Last class we say a randomized algorithm for factoring univariate polynomials over a finite field. This class we shall look at another algorithm for factoring. This was given by Berlekamp.

17 Berlekamp's Algorithm

We are given a polynomial $f(X) \in \mathbb{F}_p[X]$. As in all factoring algorithms, the first thing to do is make f square free. Once we have done this, the polynomial is of the form

$$f = f_1 f_2 \cdots f_m$$

where each f_i is a distinct irreducible factor of f. Then, chinese remaindering tells us that

$$R = \mathbb{F}_p[X]/(f) = (\mathbb{F}_p[X]/(f_1)) \times (\mathbb{F}_p[X]/(f_2)) \times \cdots \times (\mathbb{F}_p[X]/(f_m))$$

Let the degree of f_i be d_i and the degree of f be n.

17.1 The Frobenius Map

Here enters the frobenius map again. Consider the following function from R to itself.

$$\begin{array}{rcl} T : R & \longrightarrow & R \\ a & \mapsto & a^p \end{array}$$

The first thing to note here is that all elements of $\mathbb{F}_p$ are fixed in this map because we know that elements $\mathbb{F}_p$ satisfy $X^p - X = 0$. And further, we also saw the special case of binomial theorem that said $(X + Y)^p = X^p + Y^p$.

To understand this map T better, let us understand R. We have defined $R = \mathbb{F}_p[X]/(f)$ which is basically polynomials over $\mathbb{F}_p$ modulo f. And clearly, every element of R has degree atmost $n-1$ and therefore a polynomial of the form $a_0 + a_1X + \cdots a_{n-1}X^{n-1}$ can be thought of as the vector $(a_0, a_1, \cdots, a_{n-1})$. Thus, the ring R is a vector space of dimension n over $\mathbb{F}_p$.

Now, notice that the map T described above is $\mathbb{F}_p$-linear. By this, we mean that for all $\alpha, \beta \in \mathbb{F}_p$ and $u, v \in R$, we have $T(\alpha u + \beta v) = \alpha T(u) + \beta T(v)$. If we think of these elements of $\mathbb{F}_p$ as scalars, they can be 'pulled out of' T.

Therefore, it's enough to know the image of each X^i by the map.

$$\begin{aligned} p(X) &= a_0 + a_1X + \cdots + a_{n-1}X^{n-1} \\ T(p(X)) &= a_0 + a_1T(X) + \cdots + a_{n-1}T(X^{n-1}) \end{aligned}$$

17.2 The Berlekamp Sub-algebra

Now let B be the map $T - I$ where I is the identity map (maps everything to itself). Then B sends any element $a \in R$ to $a^p - a$. Now define $\mathcal{B} = \ker(B) = \ker(T - I)$. It is easy to check that the kernel of any linear map from one vector space into another (in this case R to R) forms a subspace of the vector space. Hence $\mathcal{B}$ is a subspace of the vector space R.

This space $\mathcal{B}$ is called the Berlekamp sub-algebra.

What does this space look like? Let a be any element in $\mathcal{B}$ and therefore is an element of R. Let the chinese remainder theorem map this to the tuple $(a_1, a_2, \cdots, a_m)$. And therefore $a^p - a = (a_1^p - a_1, \cdots, a_m^p - a_m)$. And since $a \in \mathcal{B}$, each of the $a_i^p - a_i$ must be 0. Now, $a_i^p - a_i$ is an element of $\mathbb{F}_p[X]/(g_i) \cong \mathbb{F}_{p^{d_i}}$ and therefore $a_i^p - a_i = 0$ can happen only if $a_i \in \mathbb{F}_p$.[3]

And therefore, each element of the tuple will infact be an element of $\mathbb{F}_p$ and therefore

$$\mathcal{B} \cong \mathbb{F}_p \times \cdots \times \mathbb{F}_p$$

And since $\mathcal{B}$ is a product of m copies of $\mathbb{F}_p$, $\mathcal{B}$ is an m dimensional subspace of R over $\mathbb{F}_p$.

[3] the elements of $\mathbb{F}_{p^d}$ that satisfy $X^p - X = 0$ are precisely those elements of $\mathbb{F}_p$

17.3 Finding a Basis

A basis for R is obvious, $\{1, X, X^2, \cdots, X^{n-1}\}$ but how do we find a basis for $\mathcal{B}$? Let us say T acts on R as

$$T(X^i) = \sum_{j=0}^{n-1} \alpha_{ji} X^j$$

then we can think of T as a the matrix $(\alpha_{ji})_{i,j}$. Thus, thinking of polynomials in R as a tuple of coefficients described above, then the action of T is just left multiplication by this matrix.

Thus, the matrix for B would be $\hat{B} = (\alpha_{ji})_{i,j} - I$. Hence the kernel of this map is just asking for all vectors v such that $\hat{B}v = 0$. And therefore, a basis for $\mathcal{B}$ can be obtained by gaussian elimination of $\hat{B}$.

Once we have a basis $\{b_1, b_2, \cdots, b_m\}$, we can pick a random element of $\mathcal{B}$ by just picking m random elements a_m from $\mathbb{F}_p$ and $\sum a_i b_i$ would be our random element from $\mathcal{B}$.

Any element a in $\mathcal{B}$ gets mapped to $\mathbb{F}_p \times \cdots \times \mathbb{F}_p$ by the Chinese remainder theorem. And therefore, we can use the Cantor-Zassenhaus idea there: $a^{\frac{p-1}{2}}$ corresponds to a vector of just 1s and -1s.

So here is the final algorithm.

Algorithm 6 BERLEKAMP FACTORIZATION

Input: A polynomial $f \in \mathbb{F}_p[X]$ of degree n

1: Make f square-free.
2: Let R be the ring $\mathbb{F}_p[X]/(f)$, considered as a n dimensional vector space over $\mathbb{F}_p$.
3: Construct the matrix of transformation $\hat{B}$ corresponding to the map $a \mapsto a^p - a$.
4: Use gaussian elimination and find a basis $\{b_1, b_2, \cdots, b_m\}$ for the berlekamp subalgebra $\mathcal{B}$.
5: Pick $\{a_1, \cdots, a_{m-1}\} \in_R \mathbb{F}_p$ and let $b = \sum a_i b_i$.
6: **if** $gcd(b^{\frac{p-1}{2}} + 1, f)$ is non-trivial **then**
7: **return** $gcd(b^{\frac{p-1}{2}} + 1, f)$ {Happens with probability atleast $1 - 2^{m-1}$}
8: **end if**
9: Repeat from step 5.

CS681	Computational Number Theory

Lecture 13: Codes: An Introduction

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Over the next few lectures we shall be looking at codes, linear codes in particular. In this class we shall look at the motivation and a glimpse at error detection. The details shall be done in the lectures to come.

18 Codes

Suppose you have two parties Alice and Bob who wish to communicate over a channel which could potentially be unreliable. What we mean by unreliable is that some parts of the message could be corrupt or changed. We want to make sure that the recepient can detect such corruption if any, or sometimes even recover the message from the corrupted.

We can assume that the channel sends allows sending some strings over a fixed finite alphabet (a finite field, or bits, or the english alphabet etc). The two questions we need to address here is detection and correction.

18.1 Block Codes

What if Alice needs to send a message, in say english, and the channel has a different alphabet, say binary strings. Then we need some way of converting strings of one alphabet into another. This is achieved by looking at blocks of code.

In the example of english to binary, we could look at ascii codes. Each letter would correspond to a *block* of letters in the channel.

Of course, not all blocks of bits could correspond to meaningful sentences or messages. A block code is in general just a subset of strings. To formally define it:

Definition 14. *Let Σ be the fixed finite alphabet for the channel of communication. A block code $\mathcal{C}$ of length n over this communication is a subset of Σ^n.*

Elements of $\mathcal{C}$ are called code words.

Definition 15. *For any two strings x and y of the same length, the hamming distance is defined as the number of indices that x and y differ in.*

$$d(x, y) = |\{i \ : \ x_i \neq y_i\}|$$

Definition 16. *The distance of a code $\mathcal{C}$ is the minimum distance between its codewords. That is,*

$$d(\mathcal{C}) = \min_{x \neq y} d(x, y)$$

In a sense, the distance of a code is a measure of how much one needs to change to alter one code to another. For example, if the distance of a code was say 5, then it means that there are two strings (messages) x and y that differ at just 5 places. Now suppose x was sent through the channel and those 5 bits were changed due to the noise in the channel, then Bob would receive the message y from Alice while she had sent x. And since y was also a message, Bob could completely misinterpret Alice's message.

We would like codes to have large distance so that it takes a lot of corruption to actually alter one codeword into another. From this, we have a simple observation.

Observation 18. *Let d be the distance of a code $\mathcal{C}$. Then the code is $d-1$-detectable, or if the channel corrupts atmost $d-1$ letters of the message, then the other party can detect that the code word has been corrupted.*

Proof. As we remarked earlier, since the distance is d, it takes atleast d corruptions to change one code word into another. Therefore, on any code word x, something less than d corruptions cannot change it to another codeword. Therefore, if the string Bob received was a codeword, then he knows for sure that Alice had sent that string for sure. □

Therefore, if the channel changed atmost t bits, any code with distance atleast $t + 1$ would allow error detection. But of course, if Bob received "I hobe you", he knows that the message was corrupted but he has no way of determining whether the message was "I love you" or "I hate you". In order to correct t errors, you need a more than just a distance of $t + 1$.

Observation 19. *If $\mathcal{C}$ is a code of distance atleast $2t + 1$, then any message that is corrupt by atmost t bits can be recovered. Or in other words, the code is t-correctable.*

Suppose Alice had sent some codeword x and let us say this was altered through the channel and Bob received it as y. Given that atmost t bits were altered, we want to show that Bob can infact recover the message.

Since Bob knew that atmost t bits are corrupted, he looks at all codewords at a hamming distance of atmost t from y.[4] Now clearly x is a codeword that is present at a hamming distance at most t from y. If x was the only such code word, then Bob knows for sure that the message Alice sent has to be x.

But it must be the case that x is the only such codeword. Suppose not, say there was some other codeword $x' \neq x$ at a distance atmost t from y. Now since x and y differ at most t places and y and x' at atmost t, by the triangle inequality x and x' differ at atmost $2t$ places. But this contradicts the assumption that the distance of the code is atleast $2t + 1$.

Or in other words, if you want to move from one codeword to another through corruption, you need to corrupt $2t + 1$ bits atleast. And therefore if you corrupt just t place, you are definitely closer to where you started from than any other codeword.

But of course, it does not make sense to look at all code words of distances less than t from y to decode. Even besides that, how do we even figure out if a given word is a code word or not. So two important properties that we would want the code to have is efficient detection of codewords and effecient decoding.

19 Linear Codes

Recall that a block code $\mathcal{C}$ is an arbitrary subset of Σ^n. These codes could have no structure underlying them and that inherently makes detecting if a string is a codeword hard. Hence comes the idea for linear codes.

Since our alphabet is finite, we shall assume that the alphabet is infact a finite field $\mathbb{F}_q$. Now our space is $\mathbb{F}_q^n$ which is infact a vector space over $\mathbb{F}_q$ of dimension n. Instead of looking at arbitrary subsets of this space, linear codes restrict themselves to subspaces of $\mathbb{F}_q^n$.

Definition 17. *A $[n, k, d]_q$ linear code $\mathcal{C}$ such that $\mathcal{C}$ is a k-dimensional subspace of $\mathbb{F}_q^n$ and has distance d.*

That is, if $x, y \in \mathcal{C}$ then so is $\alpha x + \beta y$ for all $\alpha, \beta \in \mathbb{F}_q$.

To intuitively understand the parameters, we would be encoding messages of length k with codes of length n so that it can error-correct up to $d/2$ errors. Thus we want k to be as close to n as possible and also try to

[4] can be thought of as putting a ball of radius t around y

make d large. It is not possible to arbitrarily increase both but we want some reasonable values.

To see an example of a linear code, consider our field to be $\mathbb{F}_2$. Then if $\mathcal{C} = \{(0,0,0),(1,1,1)\}$ then this is a $[3,1,3]_2$ linear code.

Definition 18. *The weight of any string x is the number of indices of x that have a 1 in it.*

$$wt(x) = |\{i \ : \ x_i = 1\}|$$

Then we have the following easy observation.

Observation 20. *If $\mathcal{C}$ is a linear code, then*

$$d(\mathcal{C}) = \min_{x \neq 0} wt(x)$$

Proof. By definition, $d(\mathcal{C}) = d(x,y)$ but $d(x,y) = wt(x-y)$. Note that since we are looking at a linear code, $x, y \in \mathcal{C}$ also tells us that $x - y \in \mathcal{C}$. Therefore, for every $x \neq y$ we have a corresponding $0 \neq z = x - y$ that is a codeword whose weight is exactly the distance between x and y. □

19.1 Detection

Suppose we have a linear code $\mathcal{C}$ and given a string x we want to check if this is in the code or not. Since we know that our code is a subspace of $\mathbb{F}_q^n$, we can represent $\mathcal{C}$ using a basis. Let us say we are looking at $[n,k,\_]_q$ codes and our basis be $\{b_1, b_2, \cdots, b_k\}$ where each $b_i \in \mathbb{F}_q^n$.

The idea is that we want to construct a matrix H such that $Hx = \bar{0}$ if and only if $x \in \mathcal{C}$. Thus, in terms of tranformations, we want a linear map H such that the kernel of this map is precisely (nothing more, nothing less) $\mathcal{C}$. The question is, how do we find such a matrix?

We first find a transformation that achieves what we want and then try and figure out what the matrix of the transformation should look like. We have a basis $\{b_1, b_2, \cdots, b_k\}$ for our code. Let us extend this first to a basis $\{b_1, b_2, \cdots, b_n\}$ of $\mathbb{F}_q^n$.

Thus, every vector $v \in \mathbb{F}_q^n$ can be written as a unique linear combination of b_is. We do the obvious transformation: map all b_i where $i \leq k$ to 0 and the rest of the basis elements to identity.

$$\begin{aligned} v &= \alpha_1 b_1 + \alpha_2 b_2 + \cdots + \alpha_k b_k + \alpha_{k+1} b_{k+1} + \cdots + \alpha_n b_n \\ Hv &= \alpha_{k+1} b_{k+1} + \cdots + \alpha_n b_n \end{aligned}$$

Now a vector v will be in the kernel of H if and only if all α_i where $i > k$ is zero. Then v has to be a linear combination of just the basis elements of $\mathcal{C}$ and therefore must itself be a codeword.

Now comes the question of how to compute the matrix of transformation of H. Suppose it turns out that that the basis we chose was actually the standard basis $\{e_1, e_2, \cdots, e_n\}$. Then what can we say about the matrix H? Then clearly, it should map all vector $(\alpha_1, \alpha_2, \cdots, \alpha_n)$ to $(0, 0, \cdots, 0, \alpha_{k+1}, \cdots, \alpha_n)$. And this matrix is just

$$\hat{H} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}_{n \times n}$$

where the I is the identity matrix of order $n - k$. But it's unlikely that we start with such a nice basis. The good news is that we can easily move from one basis to another. We just want a way to send each b_i to e_i so that we can then use the transformation $\hat{H}$ to send it to 0 if $i \leq k$ and keep it non-zero otherwise. Instead of sending b_i to e_i, the other direction is easy to compute.

What if we ask for a matrix B such that $Be_i = b_i$? This is easy because Be_i is just the i-th column of the matrix B. Thus the matrix is just $[b_1 b_2 \cdots b_n]$ where each b_i is now expanded as a column vector; just place each basis element side by side as a column vector and that is the transformation. Now that we have a matrix B that sends e_i to b_i, how do we get a matrix that sends b_i to e_i? Take B^{-1}!

Now look at $\hat{H}B^{-1}$ as a transformation. $\hat{H}B^{-1}b_i = \hat{H}e_i$ which is 0 if $i \leq k$ and e_i otherwise. Thus this matrix $\hat{H}B^{-1}$ is the matrix we were after: a matrix whose kernel is precisely the code $\mathcal{C}$.

CS681	Computational Number Theory

Lecture 14: BCH Codes

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

We shall look a a special form of linear codes called cyclic codes. These have very nice structures underlying them and we shall study BCH codes.

20 General Codes

Recall that a linear code $\mathcal{C}$ is just a subspace of $\mathbb{F}_q^n$. We saw last time that by picking a basis of $\mathcal{C}$ we can construct what is known as a parity check matrix H that is zero precisely at $\mathcal{C}$.

Let us understand how the procedure works. Alice has a message, of length k and she wishes to transmit across the channel. The channel is unreliable and therefore both Alice and Bob first agree on some code $\mathcal{C}$. Now how does Alice convert her message into a code word in $\mathcal{C}$? If Alice's message could be written as $(x_1, x_2, \cdots, x_k)$ where each $x_i \in \mathbb{F}_q$, then Alice simply sends $\sum_{i=1}^{k} x_i b_i$ which is a codeword.

Bob receives some y and he checks if $Hy = 0$. Assuming that they choose a good distance code (the channel cannot alter one code into another), if Bob finds that $Hy = 0$, then he knows that the message he received was untampered with.

But what sort of errors can the channel give? Let us say that the channel can change atmost t positions of the codeword. If x was sent and x' was received with atmost t changes between them, then the vector $e = x' - x$ can be thought of as the error vector. And since we assumed that the channel changed atmost t positions, the error vector can have weight atmost t.

This then means that Alice sent an x and Bob received $x + e$. Bob runs the parity check matrix on $x + e$ to get $H(x + e) = Hx + He = He$. The quantity He is called the *syndrome*, which is just the evaluation of Bob's received message by the parity check matrix. If the syndrome is zero, Bob knows that the received word is a valid codeword.

Of course, in order to determine what the actual message was, Bob needs to figure out what e is (for then he knows the message was $x' - e$) but recovering e from He is still a hard thing. It is not clear how this can be done efficiently on a general setting.

21 Cyclic Codes

Definition 19. *A cyclic code $\mathcal{C}$ is a linear code such that if $(c_0, c_1, \cdots, c_{n-1})$ is a codeword, then so is $(c_{n-1}, c_0, c_1, \cdots, c_{n-2})$. To put it algebraically, the space of codewords is invariant under cyclic shifts.*

If course any codeword that is shifted by i places, to the left or the right, will also be a codeword. In order to be able to see the strong structure behind them, we need a different perspective on $\mathbb{F}_q^n$.

21.1 Codewords as Polynomials

Given a vector $(c_0, c_1, \cdots, c_{n-1})$, we can associate a polynomial naturally which is $c(X) = c_0 + c_1 X + \cdots + c_{n-1}X^n$. This is just interpretting the vector space $\mathbb{F}_q^n$ as the additive group of the ring $\mathbb{F}_q[X]/(f(X))$ where f is a polynomial of degree n, since they are both isomorphic.

The ring picture has the extra multiplicative structure which is very useful here. Suppose we have a codeword $c = (c_0, \cdots, c_{n-1})$, what can we say about the codeword $c' = (c_{n-1}, c_0, \cdots, c_{n-2})$? As a polynomial, $c = c_0 + c_1 X + \cdots + c_{n-1}X^{n-1}$ and $c' = c_{n-1} + c_0 X + \cdots + c_{n-2}X^{n-1}$. So essentially we just took the polynomial c and multiplied by X. The last term $c_{n-1}X^n$, however, was changed to c_{n-1}. How do we achieve this? Do the multiplication modulo $X^n - 1$ which is just identifying X^n by 1.

Thus, cyclic shifts is just multiplication of polynomials in $\mathbb{F}_q[X]/(X^n - 1)$ by powers of X. With this observation, the following theorem summarizes the strong underlying structure in cyclic codes.

Theorem 21. *Any cyclic code $\mathcal{C}$ is an ideal of $R = \mathbb{F}_q[X]/(X^n - 1)$. And conversely, every ideal is a cyclic code*

Proof. Let us prove the easier converse first. Let $f(X) \in R$ be an element of the ideal $\mathcal{C}$. Then it follows that for any polynomial $a(X)$, $a(X)f(X) \in \mathcal{C}$ and in particular $X^i f(X) \in \mathcal{C}$. But we already say that multiplying by powers of X was just shifting and therefore our code is also invariant under shifts.

The other direction is straightforward too. We want to show that given a cyclic code $\mathcal{C}$, for any code word $f(X)$ and any polynomial $a(X)$, $a(X)f(X) \in \mathcal{C}$.

$$\begin{aligned} a(X)f(X) &= (a_0 + a_1X + \cdots + a_{n-1}X^{n-1})f(X) \\ &= a_0f(X) + a_1(Xf(X)) + \cdots + a_{n-1}(X^{n-1}f(X)) \\ &= a_0f_0(X) + a_1f_1(X) + \cdots + a_{n-1}f_{n-1}X \qquad X^if(X) \text{ is shifting} \\ &= f'(X) \in \mathcal{C} \end{aligned}$$

□

Suppose $X^n - 1$ factorizes into irreducible polynomials over $\mathbb{F}_q$, say

$$X^n - 1 = g_1g_2 \cdots g_k$$

Then it easy to check that infact all ideals of R are principle, of the form $g(X)R$ where $g(X)$ is a factor of $X^n - 1$. And hence, we have a simple corollary to above theorem.

Corollary 22. *Every cyclic code $\mathcal{C}$ is just the set of multiples of a single polynomial $g(X) \in R$.*

This polynomial is called the generator polynomial. Let us say we pick a factor $g(X)$ of $X^n - 1$ and let its degree be d. What can we say about the dimension of the code $(g(X))$? For this, we will need the rank-nullity theorem.

Theorem 23 (Rank-Nullity)**.** *If T is a linear map from a between two vector spaces V and W, then rank(T) + nullity$(T) = \dim V$ where rank(T) is defined to be the dimension of the image of V and nullity the dimension of the kernel.*

Now look at the map $\phi : R \longrightarrow R/(g(X))$. This, being a homomorphism of rings will also be a linear map on the additive groups which are vector spaces. The dimension of R is n and the dimension of the image, which is $R/(g(X))$, is d. And therefore, the dimension of the kernel which is $\mathcal{C} = (g(X))$ is $n - d$.

What about the parity check matrix? That is extremely simple here. Since the ideal is generated by a single polynomial $g(X)$, we just need to check if any given polynomial is in the code or not by just checking if g divides it. Thus, just the modulo operation is the parity check. This can be written as a matrix as well but the idea is clear.

22 BCH Codes

BCH[5] codes is an example of a cyclic code that is widely studied in coding theory. In order to get a cyclic code, we just need to get the generating polynomial of that code.

Instead of asking for the polynomial in terms of the coefficient, what if we identify the polynomial by the roots instead? This is the general idea of a BCH code.

We are working in a vector space of dimension n over $\mathbb{F}_q$ and identifying cyclic codes as ideals of $R = \mathbb{F}_q[X]/(X^n - 1)$. Let us further impose the constraint that the roots of $X^n - 1$ are distinct by making sure $gcd(n, q) = 1$ so that the derivative is non-zero.

Let ζ be a primitive n-th root of unity in R and look at the set $\{\zeta, \zeta^2, \cdots, \zeta^d\}$ where $d < \phi(n)$ (to prevent some $\zeta^i = \zeta^j$)[6]. Now we ask for the smallest degree polynomial g that has ζ^i as a root for $1 \leq i \leq d$. This polynomial is going to be our generating polynomial for the cyclic code.

The parity check matrix of a BCH code is pretty simple. Note that if $c(X) \in \mathcal{C}$, then $c(X)$ is a multiple of $g(X)$ and in particular $c(X)$ will also have the ζ^i as roots. And therefore, all we need to check is if $c(\zeta^i) = 0$ for all $1 \leq i \leq d$. Now interpretting $c(X)$ as a vector $(c_0, c_1, \cdots, c_{n-1})$ of coefficients, the parity check reduces to the following matrix multiplication.

$$\begin{bmatrix} 1 & \zeta & \zeta^2 & \cdots & \zeta^{n-1} \\ 1 & \zeta^2 & (\zeta^2)^2 & \cdots & \zeta^{2^{n-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^d & (\zeta^d)^2 & \cdots & (\zeta^d)^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Note that the parity check matrix H is a $(n - d) \times d$ matrix.

22.1 Distance of a BCH Code

Suppose $g(X)$ is the generating polynomial for the set being the first d powers of ζ, what can we say about the distance of the cyclic code $(g(X))$?

Theorem 24. *A BCH code obtained by considering the first d powers of ζ has distance $d + 1$.*

[5] Bose, Ray-Chaudhuri, Hocquenghem

[6] why won't $d < n$ suffice?

Proof. We would like to show that the minimum weight of the code $\mathcal{C} = (g(X))$ has to be atleast $d+1$. Suppose not, then there is a codeword c such that the weight of c is less than or equal to d. Then this polynomial has atmost d positions with non-zero entries. Let us denote those coefficients by $\{c_{i_1}, c_{i_2}, \cdots, c_{i_d}\}$ and say in increasing order of indices.

We just need to check that for each $1 \leq k \leq d$

$$\sum_{j=1}^{d} c_{i_j} (\zeta^k)^{i_j} = 0$$

But the above equation corresponds to the following matrix product

$$\begin{bmatrix} \zeta^{i_1} & \zeta^{i_2} & \cdots & \zeta^{i_d} \\ (\zeta^{i_1})^2 & (\zeta^{i_2})^2 & \cdots & (\zeta^{i_d})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (\zeta^{i_1})^d & (\zeta^{i_2})^d & \cdots & (\zeta^{i_d})^d \end{bmatrix} \begin{bmatrix} c_{i_1} \\ c_{i_2} \\ \vdots \\ c_{i_d} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Note that the $d \times d$ matrix is essentially in the form of a vandermonde matrix:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix}$$

and it is well known that the determinant of this matrix is $\prod_{i<j}(x_i - x_j)$ and therefore non-zero if each x_i is distinct as in our case of ζ^{i_j}. Therefore, $Hc = 0$ and H being invertible forces that c has to be the zero vector as well!

Therefore, the only codeword that can have weight less than or equal to d is the zero vector. And therefore the minweight of the BCH code is atleast $d+1$. □

Now that we have this, we can use $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ to get a guarantee that our code has distance atleast d. This is called the *designed distance* of the BCH code. Note that the actual distance you could be larger than d. We just have a guarantee that it is atleast d but the could potentially give you codes of larger distance. There are examples of BCH codes with the actual distance larger than the designed distance.

CS681 **Computational Number Theory**

Lecture 15 and 16: BCH Codes: Error Correction

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

In these two lectures, we shall see how error correction is done in a BCH code.

(most of the class was spent on discussing the solutions for the mid-semester examination)

23 Error Correction in a BCH Code

Recall that a cyclic code is one where the space of codewords is also invariant under cyclic shifts. Last class we identified this with ideals of the ring $\mathbb{F}_q[X]/(X^n - 1)$. We also said that this is an principle ideal domain when $gcd(n, q) = 1$ and therefore every cyclic code can be identified by the polynomial that generates the ideal.

For the BCH code, we pick a principle root of unity ζ and the generator of code is chosen to be the least degree polynomial that has $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ and we argued that the distance of that cyclic code is guaranteed to be at lease d.

How about decoding? Suppose Alice sent a message and that was corrupted at atmost $\lfloor \frac{d}{2} \rfloor$ places, can Bob recover the message efficiently? The answer is yes, and we shall see how. Most of the details shall be done in the next class.

23.1 The Locator and Correction Polynomials

Alice is going to send some polynomial whose degree is bounded by $n-1$, say $c(X) = c_0 + c_1X + \cdots + c_{n-1}X^{n-1}$ and Bob would receive $c(X) + e(X)$ where e is the error. Suppose the channel can corrupt at most $\lfloor \frac{d-1}{2} \rfloor$ coefficients, we know that the number of non-zero coefficients of $e(X)$ is less than $d/2$. Let $M = \{i \ : \ e_i \neq 0\}$. And hence, $|M| = t \leq \frac{d-1}{2}$.

Now look at the two polynomials:

$$\begin{aligned} u(Y) &= \prod_{i \in M} (1 - \zeta^i Y) \\ v(Y) &= \sum_{i \in M} e_i \zeta^i Y \prod_{i \neq j \in M} (1 - \zeta^j Y) \end{aligned}$$

The polynomial $u(X)$ is called the *locator polynomial* and $v(Y)$ is called the *correction polynomial.* Suppose we have $u(Y)$, how do we find out which places of the message are corrupted? This is clear because $u(\zeta^{-i}) = 0$ if and only if $i \in M$ and therefore by just checking $u(\zeta^{-i})$ for all i, we would know precisely at what places the corruption happened.

OK, now we know where the corruption has happened. How do we find out what that coefficient of $e(Y)$ was so that we can recover the message? This is where $v(Y)$ comes in. Notice that v isn't too different from the formal derivative of u. By the chain rule, we can show that

$$u'(Y) = -\sum_{i \in M} \zeta^i \prod_{i \neq j \in M} (1 - \zeta^j Y)$$

So first find out, using the detection polynomial, the places at which the error has occured. Suppose i was one of the places, what can we say about $v(\zeta^{-i})$? Note that every term in the sum other than i will be killed as there would be the term $(1 - \zeta^i Y)$ in the product that is zero when $Y = \zeta^{-i}$. And therefore,

$$\begin{aligned} v(\zeta^{-i}) &= e_i \zeta^i \zeta^{-i} \prod_{i \neq j \in M} (1 - \zeta^j \zeta^{-i}) \\ &= e_i \prod_{i \neq j \in M} (1 - \zeta^{j-i}) \end{aligned}$$

What about $u'(\zeta^{-i})$? For the same reason, the only surviving term in the summation would be the one with i. Therefore,

$$\begin{aligned} v(\zeta^{-i}) &= e_i \prod_{i \neq j \in M} (1 - \zeta^{j-i}) \\ u'(\zeta^{-i}) &= -\zeta^i \prod_{i \neq j \in M} (1 - \zeta^{j-i}) \\ \implies \frac{v(\zeta^{-i})}{u'(\zeta^{-i})} &= -\frac{e_i}{\zeta^i} \end{aligned}$$

And we are done! Running over all detected places, we can completely recover the polynomial $e(X)$ and therefore the actual message.

All that's left to do is find out how to compute the polynomials $u(Y)$ and $v(Y)$. Once we do that, we can locate the errors and also correct them. Finding these two polynomials is the key.

23.2 Computing the Polynomials

There are two important things to note here.

- We don't need to find u and v exactly. Any αu and αv, where α is some constant, would do. We just want u and v up to a scale.
- The degree of u and v is $|M| = t \leq \frac{d-1}{2}$

And remember, all that Bob has is the knowledge that the code is constructed from $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ and the received word $r(X)$. From this, he needs to compute $u(Y)$ and $v(Y)$ to error-correct.

First, let us look at the following rational function

$$w(Y) = \frac{v(Y)}{u(Y)} = \sum_{i \in M} \frac{e_i \zeta^i Y}{1 - \zeta^i Y}$$

At this point, let us make an outrageous mathematically incorrect Taylor expansion but justify it later in the lecture. Note that we have a term of the form $\frac{1}{1-\zeta^i Y}$ and we are going to expand this as a geometric series.[7]

[7] mathematically minded people are requested to clench their fists and tolerate this for a while. it will be justified soon.

Then, we have

$$\begin{aligned}
w(Y) &= \sum_{i \in M} \frac{e_i \zeta^i Y}{1 - \zeta^i Y} \\
&= \sum_{i \in M} e_i \zeta^i Y \left(\sum_{k=0}^{\infty} (\zeta^i Y)^k \right) \\
&= \sum_{k=0}^{\infty} Y^{k+1} \left(\sum_{i \in M} e_i (\zeta^i)^k \zeta^i \right) \\
&= \sum_{k=0}^{\infty} Y^{k+1} \left(\sum_{i \in M} e_i (\zeta^{k+1})^i \right) \\
&= \sum_{k=0}^{\infty} Y^{k+1} e(\zeta^{k+1}) \\
&= \sum_{k=1}^{\infty} Y^k e(\zeta^k)
\end{aligned}$$

The first $d-1$ coefficient of $w(Y)$ can be found out easily as we can find $e(\zeta^k)$ easily. Bob has the received code word $r(X) = c(X) + e(X)$. He doesn't know what $e(X)$ or $c(X)$ is but all he needs to do is compute $r(\zeta^k)$. Note that since c is the message, c is a multiple of $g(X)$ and ζ^k is a root of g and hence c. Therefore, $r(\zeta^k) = c(\zeta^k) + e(\zeta^k) = e(\zeta^k)$.

Justifying the Mathematical Sin

Of course, you just cannot write every $\frac{1}{1-x}$ as $1 + x + x^2 + \cdots$. For example, $\frac{1}{1-2}$ is certainly not $1 + 2 + 2^2 + \cdots$. So how do we justify it here?

Now the first thing is that we cannot hope to do anything better than $d-1$ coefficients of $w(Y)$ since we just know that $c(X)$ has $\zeta, \zeta^2, \cdots, \zeta^{d-1}$ as roots. Therefore, we shall focus on finding $w(Y)$ up till the $(d-1)$-th coefficient. By this, we just mean that we are finding $w(Y) \bmod Y^d$, which is just making $Y^d = 0$ in the expression.

Now note that $1 - (\zeta^i Y)^d = 1 - \zeta^{id} Y^d = 1$ and also that $(1-x)$ divides $(1-x^d)$ and hence $(1 - \zeta^i Y)$ divides $(1 - (\zeta^i Y)^d) = 1$. Which means that there exists some polynomial $p(Y)$ such that $(1 - \zeta^i Y) p(Y) = 1$ and hence $(1 - \zeta^i Y)$ is invertible modulo Y^d.

Hence, we can rework as follows:

$$\begin{aligned} w(Y) &= \sum_{i \in M} \frac{e_i \zeta^i Y}{1 - \zeta^i Y} \bmod Y^d \\ &= \sum_{i \in M} e_i \zeta^i Y \cdot \left(1 - \zeta^i Y\right)^{-1} \bmod Y^d \end{aligned}$$

and it is easy to check that the inverse of $(1-\zeta^i Y)$ modulo Y^d is $\sum_{k=0}^{d-1} (\zeta^i Y)^k$ and hence we have the rest of the equations going through.

$$w(Y) = \sum_{k=1}^{d-1} Y^k e(\zeta^k) \bmod Y^d$$

OK, we now have $w(Y)$, how do we use that to get u and v? The idea is to solve a system of equations to get u and v. Remember that both u and v are degree t polynomials and moreover the constant term in u is 1 and the constant term in v is 0.

Here we shall give an intuitive reasoning and not go into the details of the method. Suppose $u(Y) = 1 + u_1 Y + \cdots + u_t Y^t$ and $v(Y) = v_1 Y + \cdots v_t Y^t$, then we can just think of coefficients as some parameters to evaluate. Using the values of $w(Y) \bmod Y^k$ for all $1 \leq k \leq d$, we can solve for u_i, v_i by writing a system of equations. And since the number of parameters we need to solve for is $2t$ and this is less than or equal to the number of equations we have, it can be done efficiently.

There is infact another approach to solve for u and v using something called the Berlekamp-Welch Decoder. We won't be covering this in the course but just to tell you that the locator and corrector polynomials can be computed efficiently from the received word $r(X)$.

Hence using such efficient algorithms we can compute u and v. Then we just use u to locate the errors and then v to correct them at those places. And from the analysis, it is clear that we can't hope to correct more than t errors as we would then have more parameters than equations and there may not exist a solution to that set of equations.

Thus, a BCH code of designed d can be error corrected if the number of errors is bounded above by $\lfloor \frac{d-1}{2} \rfloor$.

CS681	Computational Number Theory

Lecture 17: Primality is in NP ∩ coNP

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

We shall get into primality testing for integers in the next few classes. We shall build up the details starting with showing that it is in NP ∩ coNP, discuss randomized algorithm, and then finally get into deterministic polynomial time testing.

We shall prove Pratt's result that it is in NP ∩ coNP.

24 Pratt's Theorem

The problem is the following: we are given a number N as input and we want to check if this is prime.

Remember that the input is given in binary. It would be trivial if N was specified in unary in which case the input size is N and hence primality testing in $O(N^c)$ is trivially accomplished by checking every number less than N if it's a factor or not.

The input is provided in binary and therefore we are looking for an algorithm that runs in time polynomial in the input size, which is $\log N$.

Recall the definition of the classes NP and coNP.

Definition 20. NP *is the class of languages L such there exists a polynomial time verification scheme $A(x, y)$ such that $x \in L$ if and only there exists a witness y such that $|y| < |x|^c$ for some constant c and $A(x, y) = 1$.*

coNP *is the class of languages L such that $\bar{L} \in$ NP.*

To get a more intuitive picture, NP is the class of problems that have very short proofs or witnesses. Though it might not be clear if $x \in L$, given a witness y, it is easy to check that (x, y) is a proper solution. For example, sudoku. It might be hard to find a solution but once someone gives us a solution, it is easy to check if the solution is correct.

One could also think of this as guessing a witness y and verifying it using A.

Here is an obvious observation:

Observation 25. *Primality testing is in* coNP.

This equivalent to saying that checking if a number N is composite is in NP which is immediate since the witness is the factor d of the number. Hence, our verification scheme $A(N, d)$ is just checking if d divides N.

Pratt showed that primality testing is infact in NP.

Theorem 26. *Primality testing is in* NP.

Proof. Note that the group $(\mathbb{Z}/N\mathbb{Z})^\star$ is of order $N-1$ if and only if N is prime. And more over, it is a cyclic group of order $N-1$ if and only if N is a prime. Thus, we shall find a witness or a certificate that the group is cyclic.

How do we show that a group is cyclic? We guess a generator a. If we are able to show that $a^n \neq 1$ for any $n < N-1$, we are done. Note that $a^{N-1} = 1$ anyway. Therefore, we just need to check that $a^{(N-1)/p_i} \neq 1$ for every prime divisor p_i of $N-1$.

Therefore, we not only guess the generator a, we guess the factors $p_1, p_2, \cdots, p_k$ of $N-1$. But how do we know that the guessed p_is are indeed primes? We guess its witnesses too; induction! Aren't we going in circles? Actually no since the numbers p_is are quite small and it still won't blow up the size of the final certificate.

Let us try and see how large the witness/certificate can get. How large can the prime factors of $N-1$ be? Since N is prime , $N-1$ is certainly composite (unless N was 2, a worthless case which can be handled right at the beginning). The largest factor of $N-1$ can be of size atmost $\sqrt{N}$. How many factors can there exist? Atmost $\log(N-1)$ of them. Thus if $N-1 = p_1 p_2 \cdots p_k$ then our witness would be $(a, p_1, p_2, \cdots, p_n)$ and the certificates of each of the p_is. The input is of size $\log N$ and let $S(l)$ be the size of the witness for input of length l. Then:

$$\begin{aligned} S(\log N) &= \log^2 N + S(\log p_1) + S(\log p_2) + \cdots + S(\log p_k) \\ &= \log^2 N + (\log p_1)^c + (\log p_2)^c + \cdots + (\log p_k)^c \\ &\leq \log^2 N + (\log p_1 + \log p_2 + \cdots + \log p_k)^c \\ &= \log^2 N + (\log(N-1))^c \\ &= O((\log N)^c) \end{aligned}$$

And since the witness is just polynomially bounded in the size of the input, we can guess the entire certificate and verify. Thus primality testing is in NP. □

And since primality is in NP and coNP, it is in NP ∩ coNP.

CS681	Computational Number Theory

Lecture 17: Primality is in NP ∩ coNP

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

We shall get into primality testing for integers in the next few classes. We shall build up the details starting with showing that it is in NP ∩ coNP, discuss randomized algorithm, and then finally get into deterministic polynomial time testing.

We shall prove Pratt's result that it is in NP ∩ coNP.

25 Pratt's Theorem

The problem is the following: we are given a number N as input and we want to check if this is prime.

Remember that the input is given in binary. It would be trivial if N was specified in unary in which case the input size is N and hence primality testing in $O(N^c)$ is trivially accomplished by checking every number less than N if it's a factor or not.

The input is provided in binary and therefore we are looking for an algorithm that runs in time polynomial in the input size, which is $\log N$.

Recall the definition of the classes NP and coNP.

Definition 21. NP *is the class of languages L such there exists a polynomial time verification scheme $A(x, y)$ such that $x \in L$ if and only there exists a witness y such that $|y| < |x|^c$ for some constant c and $A(x, y) = 1$.*

coNP *is the class of languages L such that $\bar{L} \in$ NP.*

To get a more intuitive picture, NP is the class of problems that have very short proofs or witnesses. Though it might not be clear if $x \in L$, given a witness y, it is easy to check that (x, y) is a proper solution. For example, sudoku. It might be hard to find a solution but once someone gives us a solution, it is easy to check if the solution is correct.

One could also think of this as guessing a witness y and verifying it using A.

Here is an obvious observation:

Observation 27. *Primality testing is in* coNP.

This equivalent to saying that checking if a number N is composite is in NP which is immediate since the witness is the factor d of the number. Hence, our verification scheme $A(N, d)$ is just checking if d divides N.

Pratt showed that primality testing is infact in NP.

Theorem 28. *Primality testing is in* NP.

Proof. Note that the group $(\mathbb{Z}/N\mathbb{Z})^\star$ is of order $N-1$ if and only if N is prime. And more over, it is a cyclic group of order $N-1$ if and only if N is a prime. Thus, we shall find a witness or a certificate that the group is cyclic.

How do we show that a group is cyclic? We guess a generator a. If we are able to show that $a^n \neq 1$ for any $n < N-1$, we are done. Note that $a^{N-1} = 1$ anyway. Therefore, we just need to check that $a^{(N-1)/p_i} \neq 1$ for every prime divisor p_i of $N-1$.

Therefore, we not only guess the generator a, we guess the factors $p_1, p_2, \cdots, p_k$ of $N-1$. But how do we know that the guessed p_is are indeed primes? We guess its witnesses too; induction! Aren't we going in circles? Actually no since the numbers p_is are quite small and it still won't blow up the size of the final certificate.

Let us try and see how large the witness/certificate can get. How large can the prime factors of $N-1$ be? Since N is prime , $N-1$ is certainly composite (unless N was 2, a worthless case which can be handled right at the beginning). The largest factor of $N-1$ can be of size atmost $\sqrt{N}$. How many factors can there exist? Atmost $\log(N-1)$ of them. Thus if $N-1 = p_1 p_2 \cdots p_k$ then our witness would be $(a, p_1, p_2, \cdots, p_n)$ and the certificates of each of the p_is. The input is of size $\log N$ and let $S(l)$ be the size of the witness for input of length l. Then:

$$\begin{aligned} S(\log N) &= \log^2 N + S(\log p_1) + S(\log p_2) + \cdots + S(\log p_k) \\ &= \log^2 N + (\log p_1)^c + (\log p_2)^c + \cdots + (\log p_k)^c \\ &\leq \log^2 N + (\log p_1 + \log p_2 + \cdots + \log p_k)^c \\ &= \log^2 N + (\log(N-1))^c \\ &= O((\log N)^c) \end{aligned}$$

And since the witness is just polynomially bounded in the size of the input, we can guess the entire certificate and verify. Thus primality testing is in NP. □

And since primality is in NP and coNP, it is in NP ∩ coNP.

CS681	Computational Number Theory

Lecture 18: Quadratic Reciprocity

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Polynomial factorization and randomized primality testing were one of the first examples of the power of randomization. Two standard algorithms for primality testing (randomized) are the Miller-Rabin test and the Solovay-Strassen test.

We shall build some theory on quadratic reciprocity laws before we get into the Solovay Strassen test.

26 Quadratic Reciprocity

The reciprocity laws are closely related to how primes split over *number fields.* Let us first understand what these number fields are.

Definition 22. *An algebraic integer over $\mathbb{Q}$ is an element ζ such that it is a root of a monic polynomial in $\mathbb{Z}[X]$. For example, the number $\frac{1}{2} + i\frac{\sqrt{3}}{2}$ is an algebraic integer as it is a root of $x^2 - x + 1$.*

A number field is a finite extension of $\mathbb{Q}$. One could think of this as just adjoining an algebraic number to $\mathbb{Q}$.

Note that number fields are strange objects. They may not even be UFDs. We saw the example when we consider $\mathbb{Q}(\sqrt{-5})$, the number 6 factors as both 3×2 and $(1+\sqrt{-5})(1-\sqrt{-5})$. However, if one were to consider factorization over ideals, they form unique factorizations.

26.1 The Legendre Symbol

Fix an odd prime p. We want to study equations of the form $X^2 - a$ over $\mathbb{F}_p$. What does it mean to say that this has a solution in $\mathbb{F}_p$? It means that a has a square-root in $\mathbb{F}_p$ or a is a square in $\mathbb{F}_p$. The legendre symbol captures that.

Definition 23. *For $a \in \mathbb{F}_p$, the legendre symbol $\left(\frac{a}{p}\right)$ is defined as follows:*

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } p \mid a \\ -1 & \text{if } a \text{ is not a square modulo } p \\ 1 & \text{if } a \text{ is a square modulo } p \end{cases}$$

Proposition 29.

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$$

The proof is fairly straight forward; just consider them case by case when they are $-1, 0, 1$.

Thus, the above proposition tells us that the $\left(\frac{\cdot}{p}\right)$ is a homomorphism from $\mathbb{Z}/p\mathbb{Z}$ to $\{-1, 0, 1\}$.

Another observation is that since $\mathbb{F}_p^\star$ is cyclic, there is a generator b. Then we can write $a = b^t$. We then have,

$$a = \begin{cases} 0 & \text{if } p \mid a \\ -1 & \text{if } a \text{ is not a square modulo } p \\ 1 & \text{if } a \text{ is a square modulo } p \end{cases}$$

and therefore $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}}$.

Note that $x^2 = y^2 \implies x = y$ or $x = -y$ and therefore, the number of squares in $\mathbb{F}_p^\star$ is exactly $\frac{p-1}{2}$. And if the generator of the group is a quadratic non-residue (not a square), then any odd power of the generator is also a non-residue.

27 Quadratic Reciprocity Theorem

Theorem 30. *Let p and q be odd primes (not equal to each other). Then*

$$\begin{aligned} \left(\frac{2}{p}\right) &= (-1)^{\frac{p^2-1}{8}} \\ \left(\frac{p}{q}\right)\left(\frac{q}{p}\right) &= (-1)^{\frac{p-1}{2}\frac{q-1}{2}} \end{aligned}$$

Proof. We shall prove the part of $\left(\frac{2}{p}\right)$ in this class and do the other in the next. The idea is to go to a field extension (if necessary) and evaluate certain elements in two ways to get what we want. In the case of $\left(\frac{2}{p}\right)$ we shall go to the extension $\mathbb{F}_p(i)$ where i is a square root of -1.

Firstly note that this needn't be a proper extension at all. For example, in $\mathbb{F}_5$, we already have a root of -1 which is 2. Infact, for every prime of the form $1 \bmod 4$, we have a square root of -1. So we will go to an extension if necessary.

Now set $\tau = 1 + i$. We know that $\tau^2 = 1 - 1 + 2i = 2i$ and $\tau^p = 1 + i^p$ in $\mathbb{F}_p$. We could also evaluate τ^p as $\tau \cdot (\tau^2)^{\frac{p-1}{2}}$. Also $(1+i)^{-1} = \frac{1-i}{2}$.

$$\begin{aligned} 1 + i^p &= \tau^p \\ &= \tau(2i)^{\frac{p-1}{2}} \\ &= (1+i)2^{\frac{p-1}{2}} i^{\frac{p-1}{2}} \\ \implies \frac{(1+i^p)(1-i)}{2} &= 2^{\frac{p-1}{2}} i^{\frac{p-1}{2}} \\ \implies \frac{1 + i^p - i - i^{p+1}}{2} &= \left(\frac{2}{p}\right) i^{\frac{p-1}{2}} \end{aligned}$$

Case 1: When $p = 1 \bmod 4$

Then $i \in \mathbb{F}_p$ and the above equation reduces to

$$\begin{aligned} \frac{1 + i - i + 1}{2} &= \left(\frac{2}{p}\right)(-1)^{\frac{p-1}{4}} \\ \implies \left(\frac{2}{p}\right) &= (-1)^{\frac{p-1}{4}} \end{aligned}$$

Case 2: When $p = 3 \bmod 4$

$$\begin{aligned} \frac{1 - i - i - 1}{2} &= \left(\frac{2}{p}\right) i^{\frac{p-1}{2}} \\ \implies i^3 &= \left(\frac{2}{p}\right) i^{\frac{p-1}{2}} \\ \implies \left(\frac{2}{p}\right) &= i^{\frac{1-p}{2}+3} = i^{\frac{8-(1+p)}{4}} \\ &= (-1)^{\frac{p+1}{4}} \end{aligned}$$

Therefore,

$$\left(\frac{2}{p}\right) = \begin{cases} (-1)^{\frac{p-1}{4}} & p = 1 \bmod 4 \\ (-1)^{\frac{p+1}{4}} & p = 3 \bmod 4 \end{cases}$$

and combining the two, we get

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$$

□

The proof of the other part is very similar. We consider a similar τ and evaluate τ^p in two different ways to get to our answer. We shall do this in the next class.

CS681	Computational Number Theory

Lecture 19: Quadratic Reciprocity (contd.)

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Last class we proved one part of the Quadratic Reciprocity Theorem. We shall first finish the proof of the other part and then get to a generalization of the Legendre symbol - the Jacobi symbol.

28 Proof of Reciprocity Theorem (contd.)

Recall the statement of the theorem. If $p \neq q$ are odd primes, then

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$$

The idea of the proof is just the same. We choses $\tau = 1 + i$ last time and we got a $2^{\frac{p-1}{2}}$ term while computing τ^p. It would be the same here as well.

Let ζ be a principle q-th root of unity. We shall work with the field $\mathbb{F}_p(\zeta)$. Let

$$\tau = \sum_{a \in \mathbb{F}_q^\star} \left(\frac{a}{q}\right) \zeta^a$$

Just as before, we compute τ^2.

$$\begin{aligned}
\tau^2 &= \left(\sum_{a \in \mathbb{F}_q^\star} \left(\frac{a}{q}\right) \zeta^a\right)\left(\sum_{b \in \mathbb{F}_q^\star} \left(\frac{b}{q}\right) \zeta^b\right) \\
&= \sum_{a,b \in \mathbb{F}_q^\star} \left(\frac{a}{q}\right)\left(\frac{b}{q}\right) \zeta^{a+b} \\
&= \sum_{a,b \in \mathbb{F}_q^\star} \left(\frac{a}{q}\right)\left(\frac{b^{-1}}{q}\right) \zeta^{a+b} \\
&= \sum_{a,b \in \mathbb{F}_q^\star} \left(\frac{ab^{-1}}{q}\right) \zeta^{a+b}
\end{aligned}$$

Let us do a change of variable, by putting $c = ab^{-1}$

$$\begin{aligned}\tau^2 &= \sum_{c,b\in\mathbb{F}_q^\star}\left(\frac{c}{q}\right)\zeta^{bc+b}\\ &= \sum_{-1\neq c\in\mathbb{F}_q^\star}\left(\frac{c}{q}\right)\sum_{b\in\mathbb{F}_q^\star}(\zeta^{c+1})^b + \sum_{b\in\mathbb{F}_q^\star}\left(\frac{-1}{q}\right)\end{aligned}$$

Since both p and q are primes and $-1 \neq c \in \mathbb{F}_q^\star$, ζ^{c+1} is also a principle q-th root of unity. And therefore, $\sum_b(\zeta^{c+1})^b = -1$. Therefore,

$$\tau^2 = \sum_{-1\neq c\in\mathbb{F}_q^\star}\left(\frac{c}{q}\right) + (q-1)\left(\frac{-1}{q}\right)$$

Look at the first term. If one were to sum over all elements of $\mathbb{F}_q^\star$, then half of the $\left(\frac{c}{q}\right)$ would be 1 and the other would be -1 thus fully cancelling off. Since we are just excluding $c = -1$, the first term is just $\left(\frac{-1}{q}\right)$.

$$\tau^2 = \left(\frac{-1}{q}\right) + (q-1)\left(\frac{-1}{q}\right) = q\left(\frac{-1}{q}\right)$$

Now to evaluate τ^p.

$$\begin{aligned}\tau^p &= \sum_{a\in\mathbb{F}_q^\star}\left(\frac{a}{q}\right)\zeta^{ap}\\ &= \sum_{c\in\mathbb{F}_q^\star}\left(\frac{cp^{-1}}{q}\right)\zeta^c \qquad (c = ap)\\ &= \left(\frac{p^{-1}}{q}\right)\sum_{c\in\mathbb{F}_q^\star}\left(\frac{c}{q}\right)\zeta^c\\ &= \left(\frac{p}{q}\right)\tau\\ \tau^p &= \tau(\tau^2)^{\frac{p-1}{2}}\\ \implies \left(\frac{p}{q}\right)\tau &= \tau q^{\frac{p-1}{2}}\left(\frac{-1}{q}\right)^{\frac{p-1}{2}}\\ &= \tau\left(\frac{q}{p}\right)(-1)^{\frac{p-1}{2}\frac{q-1}{2}}\\ \implies \left(\frac{p}{q}\right)\left(\frac{q}{p}\right) &= (-1)^{\frac{p-1}{2}\frac{q-1}{2}} \quad \square\end{aligned}$$

29 Jacobi Symbol

The legendre symbol $\left(\frac{m}{n}\right)$ can be naturally generalized to the case when m and n are odd and coprime numbers.

Definition 24. *Suppose $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$. Then the Jacobi symbol, also represented as $\left(\frac{m}{n}\right)$, is defined as follows*

$$\left(\frac{m}{n}\right) = \begin{cases} 0 & \text{if } (m,n) \neq 1 \\ \prod_{i=1}^{k} \left(\frac{m}{p_i}\right)^{\alpha_i} & \text{otherwise} \end{cases}$$

The Jacobi symbol also satisfies some nice multiplicative properties

- $\left(\frac{m_1 m_2}{n}\right) = \left(\frac{m_1}{n}\right)\left(\frac{m_2}{n}\right)$
- $\left(\frac{m}{n_1 n_2}\right) = \left(\frac{m}{n_1}\right)\left(\frac{m}{n_2}\right)$

Using the above properties, we can get a generalize the theorem on the legendre symbol as well.

Theorem 31. *If m, n are odd numbers such that $(m,n) = 1$, then*

$$\begin{aligned} \left(\frac{2}{n}\right) &= (-1)^{\frac{n^2-1}{8}} \\ \left(\frac{m}{n}\right)\left(\frac{n}{m}\right) &= (-1)^{\frac{m-1}{2}\frac{n-1}{2}} \end{aligned}$$

We shall prove this theorem in the next class. The proof will just be using induction on the factors of m and n.

WARNING: Please note that the Jacobi symbol $\left(\frac{m}{n}\right)$ doesn't say anything about whether or not m is a square modulo n. For example, if $n = p_1 p_2$ and m was chosen such that m is not a square modulo p_1 or p_2. Then the Jacobi symbol $\left(\frac{m}{p_1 p_2}\right) = (-1)(-1) = 1$ but m is not a square in $p_1 p_2$.

CS681	Computational Number Theory

Lecture 20 and 21: Solovay Strassen Primality Testing

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Last class we stated a similar reciprocity theorem for the Jacobi symbol. In this class we shall do the proof of it, discuss the algorithm, and also do the Solovay-Strassen primality testing.

30 Proof of the Reciprocity of $\left(\frac{m}{n}\right)$

The proof will just be induction on m. Recall the statement of the theorem

$$\begin{aligned}\left(\frac{2}{n}\right) &= (-1)^{\frac{n^2-1}{8}} \\ \left(\frac{m}{n}\right)\left(\frac{n}{m}\right) &= (-1)^{\frac{m-1}{2}\frac{n-1}{2}}\end{aligned}$$

We shall just prove the second part here. The first part uses the same technique. Let us assume that the theorem is true for all $m' < m$. If m is a prime, we do induction on n.

Suppose $m = m_1 m_2$, then

$$\begin{aligned}\left(\frac{m_1 m_2}{n}\right)\left(\frac{n}{m_1 m_2}\right) &= \left(\frac{m_1}{n}\right)\left(\frac{n}{m_1}\right)\left(\frac{m_2}{n}\right)\left(\frac{n}{m_2}\right) \\ &= (-1)^{\frac{n-1}{2}\left(\frac{m_1-1}{2}+\frac{m_2-1}{2}\right)}\end{aligned}$$

From now on, the work shall be happening on the exponent and let us just denote $\frac{n-1}{2}E$ for the exponent of -1. We want to evaluate $E \bmod 2$ since we are looking at (-1) power the exponent and only the parity matters.

Let $m_1 = 4k_1 + b_1$ and $m_2 = 4k_2 + b_2$ where $b_1, b_2 = \pm 1$ since m is odd.

$$\begin{aligned} E &= \frac{4k_1 + 4k_2 + b_1 + b_2 - 2}{2} \\ &= \frac{b_1 + b_2 - 2}{2} \bmod 2 \\ \frac{m-1}{2} &= \frac{(4k_1 + b_1)(4k_2 + b_2) - 1}{2} \\ &= 8k_1k_2 + 2k_1b_2 + 2k_2b_1 + \frac{b_1b_2 - 1}{2} \\ &= \frac{b_1b_2 - 1}{2} \bmod 2 \end{aligned}$$

And now it is easy to check that for $b_1, b_2 = \pm 1$,

$$\frac{b_1b_2 - 1}{2} = \frac{b_1 + b_2 - 2}{2} \bmod 2$$

and therefore, $E = \frac{m-1}{2} \bmod 2$ and hence,

$$\left(\frac{m}{n}\right)\left(\frac{n}{m}\right) = (-1)^{\frac{n-1}{2}E} = (-1)^{\frac{n-1}{2}\frac{m-1}{2}} \quad \square$$

31 Algorithm to compute $\left(\frac{m}{n}\right)$

The reciprocity laws give a polynomial time algorithm to compute the Jacobi symbol $\frac{m}{n}$. Note that $\left(\frac{m}{n}\right)$ depends only on $m \bmod n$ and therefore we can reduce m modulo n and compute. When $m < n$, we use the reciprocity to get $\left(\frac{n}{m}\right)$ and we reduce again.

The bases cases (cases when either of them is 1 or $gcd(m, n) > 1$ or $m = 2^k m'$ or $n = 2^k m'$ etc) are omitted[8].

The running time of this algorithm is $(\log m \log n)^{O(1)}$.

32 Solovay Strassen Primality Testing

The general philosophy of primality testing is the following:

- Find a property that is satisfied by exactly the prime numbers.

[8] the TEXsource file of this lecture note has them commented out. Uncomment them and recompile if needed

Algorithm 7 JACOBI SYMBOL $\left(\frac{m}{n}\right)$

```
//base cases omitted
if m > n then
    return (m mod n / n)
else
    return (-1)^((m-1)/2 (n-1)/2) (n/m)
end if
```

- Find an efficient way to check if the property is satisfied by arbitrary numbers.
- Show that for any composite number, one can "easily" find a witness that the property fails.

In the Solovay-Strassen algorithm, the property used is the following.

Proposition 32. *n is prime if and only if for all $a \in (\mathbb{Z}/n\mathbb{Z})^\star$,*

$$\left(\frac{a}{n}\right) = a^{\frac{n-1}{2}}$$

And with the following claim, we have the algorithm immediately.

Claim 33. *If n was composite, then for a randomly chosen from $(Z/n\mathbb{Z})^\star$,*

$$\Pr_{a \in (\mathbb{Z}/n\mathbb{Z})^\star}\left[\left(\frac{a}{n}\right) \neq a^{\frac{n-1}{2}}\right] \geq \frac{1}{2}$$

Thus, the algorithm is the following.

All that's left to do is prove the claim. For that, let us look at a more general theorem which would be very useful.

Theorem 34. *Let ψ_1 and ψ_2 be two homomorphisms from a finite group G to a group H. If $\psi_1 \neq \psi_2$, that is there is atleast one $g \in G$ such that $\psi_1(g) \neq \psi_2(g)$, then ψ_1 and ψ_2 differ at atleast $|G|/2$ points.*

This intuitively means that two different homomorphisms can either be the same or have to be very different.

Proof. Consider the set

$$H = \{g \in G \ : \ \psi_1(g) = \psi_2(g)\}$$

Algorithm 8 SOLOVAY-STRASSEN: check if n is prime

1: Pick a random element $a < n$.
2: **if** $gcd(a, n) > 1$ **then**
3: **return** COMPOSITE
4: **end if**
5: Compute $a^{\frac{n-1}{2}}$ using repeated squaring and $\left(\frac{a}{n}\right)$ using the earlier algorithm.
6: **if** $\left(\frac{a}{n}\right) \neq a^{\frac{n-1}{2}}$ **then**
7: **return** COMPOSITE
8: **else**
9: **return** PRIME
10: **end if**

Note that clearly 1 belongs to H and if $a, b \in H$, then so is ab as $\psi_1(ab) = \psi_1(a)\psi_1(b) = \psi_2(a)\psi_2(b) = \psi_2(ab)$. Inverses are inside as well and therefore, H is a subgroup of G. Also since $\psi_1 \neq \psi_2$, they differ at atleast one point say g_0. Then $g_0 \notin H$ and hence H is a proper subgroup of G.

By Lagrange's theorem, $|H|$ divides $|G|$ and since $|H| < |G|$, $|H|$ can atmost be $|G|/2$. Since every element in $G \setminus H$ is a point where ψ_1 and ψ_2 differ, it follows that ψ_1 and ψ_2 differ at atleast $|G|/2$ points. □

The claim directly follows from the theorem since both the Jacobi symbol and the map $a \mapsto a^{\frac{n-1}{2}}$ are homomorphisms and hence will differ in atleast half of the elements of $(\mathbb{Z}/n\mathbb{Z})^\star$.

Thus, the Solovay-Strassen algorithm has the following error bounds:

- If n is a prime, the program outputs PRIME with probability 1.
- If n is not a prime, the program outputs COMPOSITE with probability atleast $\frac{1}{2}$.

Of course, the confidence can be boosted by making checks on more such a's.

All that's left to do is to prove the proposition.

33 Proof of the Proposition ??

We want to show that if n is not a prime, there the two homomorphisms $a \mapsto a^{\frac{n-1}{2}}$ and $a \mapsto \left(\frac{a}{n}\right)$ are not the same. Thus, it suffices to find a single $a \in (\mathbb{Z}/n\mathbb{Z})^\star$ such that $\left(\frac{a}{n}\right) \neq a^{\frac{n-1}{2}}$.

Case 1: n is not square free

Suppose n had a prime factor p such that p^2 divides n. Recall that for all $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, the Euler ϕ function evaluates to:

$$\phi(n) = \prod_{i=1}^{k} p_i^{\alpha_i - 1}(p_i - 1)$$

And hence, if $p^2 \mid n \implies p \mid \phi(n)$. Now look at the multiplicative group $(\mathbb{Z}/n\mathbb{Z})^\star$, this has $\phi(n)$ elements. A theorem of Cayley tells us that if $p \mid |G|$ then G has an element of order p.[9] Let g be an element of order p in $(\mathbb{Z}/n\mathbb{Z})^\star$.

What is the value of $g^{\frac{n-1}{2}}$? Can this be ± 1? If it were ± 1, then $g^{n-1} = 1$. This means that the order of g divides $n-1$, or $p \mid n-1$ which is impossible since $p \mid n$. And therefore, $g^{\frac{n-1}{2}} \neq \pm 1$ and therefore, certainly cannot be $\left(\frac{g}{n}\right)$ which takes values only ± 1 for all g coprime to n.

Thus g is a witness that $\left(\frac{g}{n}\right) \neq g^{\frac{n-1}{2}}$.

Case 2: n is a product of distinct primes

Now n will be square-free if and only if it is a product of distinct primes. Suppose $n = p_1 p_2 \cdots p_k$

Suppose there is some some a such that $a^{\frac{n-1}{2}} \neq \left(\frac{a}{p_1}\right)$, are we done? Yes we are. We can use such a a to find a g such that $g^{\frac{n-1}{2}} \neq \left(\frac{g}{n}\right)$.

By the Chinese Remainder Theorem, we know that $(\mathbb{Z}/n\mathbb{Z})^\star \cong (\mathbb{Z}/p_1\mathbb{Z})^\star \times \cdots \times (\mathbb{Z}/p_k\mathbb{Z})^\star$. Let g be the element in $(\mathbb{Z}/n\mathbb{Z})^\star$ such that $g \mapsto (a, 1, 1, \cdots, 1)$ by the CRT map. By the definition of the Jacobi Symbol,

$$\left(\frac{g}{n}\right) = \prod_{i \in 1}^{k} \left(\frac{g}{p_i}\right) = \prod_{i=1}^{k} \left(\frac{g \mod p_i}{p_i}\right) = \left(\frac{a}{p_1}\right)\left(\frac{1}{p_2}\right)\cdots\left(\frac{1}{p_k}\right) = \left(\frac{a}{p_1}\right)$$

[9] actually it is more. It says that for every prime power $p^\alpha \mid |G|$, there is a subgroup of order p^α in G.

And $g^{\frac{n-1}{2}} = (a^{\frac{n-1}{2}}, 1, \cdots, 1)$. What we know is that $a^{\frac{n-1}{2}} \neq \left(\frac{a}{p_1}\right)$. Suppose $\left(\frac{a}{p_1}\right) = 1$, then $\left(\frac{a}{p_1}\right) = \left(\frac{g}{n}\right) = 1$. But $g^{\frac{n-1}{2}}$ on the other hand looks like $(a^{\frac{n-1}{2}}, 1, \cdots, 1)$ and we know that $\left(\frac{a}{p_1}\right) = 1 \neq a^{\frac{n-1}{2}}$. Therefore, $g^{\frac{n-1}{2}}$ looks like $(*, 1, \cdots, 1)$ where the first coordinate is *not* 1. And therefore, this is not 1. Therefore $\left(\frac{g}{n}\right) \neq g^{\frac{n-1}{2}}$.

Suppose $\left(\frac{a}{p_1}\right) = -1$, then things are even simpler. $\left(\frac{g}{n}\right) = -1$ but $g^{\frac{n-1}{2}}$ looks like $(*, 1, \cdots, 1) \neq -1$. Therefore $\left(\frac{g}{n}\right) \neq g^{\frac{n-1}{2}}$.

And of course, it works for any prime factor p of n. Thus, the bad case is when for all a and for all prime factors p_i, $\left(\frac{a}{p_i}\right) = a^{\frac{n-1}{2}}$. Since n is composite, there are at least 2 distinct prime factors p_1 and p_2. Pick $a \in (\mathbb{Z}/p_1\mathbb{Z})^\star$ which is a quadratic residue ($\left(\frac{a}{p_1}\right) = 1$) and a $b \in (\mathbb{Z}/p_2\mathbb{Z})^\star$ that is a non-residue ($\left(\frac{b}{p_2}\right) = -1$). Now look at the element $g \in (\mathbb{Z}/n\mathbb{Z})^\star$ that maps to $(a, b, 1, 1, \cdots, 1)$ by the chinese remainder theorem.

Now $g^{\frac{n-1}{2}} = (a^{\frac{n-1}{2}}, b^{\frac{n-1}{2}}, 1, \cdots, 1) = (1, -1, 1, \cdots 1)$ which is not ± 1. And hence clearly, $\left(\frac{g}{n}\right) \neq g^{\frac{n-1}{2}}$.

That completes the proof of correctness of the Solovay-Strassen primality test.

CS681	Computational Number Theory

Lecture 22 : Towards the AKS Primality Test

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

Our next goal is to do the deterministic polynomial time primality test that was found by Manindra Agrawal, Neeraj Kayal and Nitin Saxena in 2002. It was a remarkable achievement and it also gave an example of how certain problems that are open for a long time can have such beautiful, elegant solutions.

34 Derandomization Approaches and the Riemann Hypothesis

We had very good randomized algorithms like the Solovay-Strassen test which we discussed last time, or the Rabin-Miller test.[10] But people wanted a deterministic algorithm for primality testing that runs in polynomial time.

However, it was known that under some strong assumptions, the above algorithms can be derandomized and made into a deterministic algorithm. If the *Extended Riemann Hypothesis* (ERH) is true, then we can completely remove randomness from the Solovay-Strassen test (or the Miller-Rabin test) and make it a deterministic polynomial time algorithm.

This conjecture is widely believe to be true and has been a very important long-standing open problem for a long time. Infact, Clay Mathematical Institute has a prize of a million dollars for anyone who solves it. Of course, if someone proves the riemann hypothesis, we already have a deterministic primality test. But trying to prove the ERH for a primality test is like trying to uproot a whole tree to get a fruit on top of it; completely avoiding the tree and instead using some stick would be better.

Let us have a small discussion on what the ERH, or the Riemann Hypothesis is.

[10] another randomized algorithm for primality testing. We will see this as an assignment.

34.1 The Zeta Function

Riemann introduced a function called the *Riemann Zeta Function.* The Riemann Hypothesis is to do with the roots of this function. Some of you might have seen the equalities like

$$\begin{aligned} 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} \cdots &= \frac{\pi^2}{6} \\ 1 + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} \cdots &= \frac{\pi^4}{90} \end{aligned}$$

One question is what happens when we vary the exponent of the summation. What can we say about

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

Note that there are diverging series like $\zeta(s)$ since

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \log n$$

and therefore if n goes to infinity, the sum goes to infinity as well; the infinite some diverges. Infact, the series $\zeta(s)$ converges for all real s if and only if $|s| > 1$.

Now why not s when it is complex? Why do we restrict ourselves to just real exponents? Then we can define $\zeta(s)$ to be the function when s is any complex number. There again, the infinite sum issues props up. It can be shown that $\sum \frac{1}{n^s}$ converges for all s such that the real part of s, denoted by $\Re(s)$ is greater than 1.

And hence $\zeta(s)$ is well defined for all numbers s such that $\Re(s) > 1$. A pictorial way to look at it is if you consider the complex plane, it is well-defined for all point to the right of the line $x = 1$.

Suppose we consider functions over real values, we have this notion of continuity. A function being continuous at a point x_0 essentially means that irrespective of whether we approach x_0 from the left (also denoted by left hand limit) or the right (right hand limit), the value should coincide with the functional value at x_0. This is sometimes also written as

$$\lim_{x \to x_0^-} f(x) = \lim_{x \to x_0^+} f(x) = f(x_0)$$

In the case of complex functions, we have a function over a plane. So it is not just a line and hence just left or right approaches. In the complex case, it is said to be continuous if no matter what part you take to approach x_0, the limits should match $f(x_0)$.

Similarly for derivatives in the real case, we want the derivative on the left hand side to match the derivative on the right hand side. In the complex case, the derivative must be the same on all directions.

A complex function that satisfies these conditions is called an *analytic function.* We say function is analytic over a region D if the above properties hold for every point in D.

Analytic functions have very strong properties like not just the first derivative but all higher order derivatives exist, a whole bunch of properties. It imposes a lot of restriction on the function.

Riemann showed that the zeta function is analytic in the region $\Re(s) > 1$.

Suppose we have a function f that we define over a small domain D over the compex plane. And over this domain, suppose the function f is analytic. We haven't defined the function outside this domain at all. A process known as *analytic continuation* can be used to extend the domain of this function.

Formally speaking, g (whose domain is D_g) is an analytic continuation of f (whose domain is D_f) if the following properties hold:

- g is analytic over D_g.
- $D_f \subseteq D_g$.
- For all $z \in D_f$, $f(z) = g(z)$.

In simple words, g extends f to a larger domain keeping in mind that if f was already defined at a point z, then g shouldn't change that value; g should coincide with f wherever it is defined already.

There is a remarkable results that if g_1 and g_2 both analytically extend f independently, then essentially $g_1 = g_2$. Therefore, the analytic continuation of a function f is uniquely determined; we can hence talk of *the* analytic continuation of f.

The zeta function is actually the analytic continuation of the function

$\sum \frac{1}{n^s}$. It is however written as just

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

The analytic continuation now extends the domain to the entire complex plane except the point $z = 1$. Thus our $\zeta(s)$ is a function defined over the entire complex plane except 1; there is a simple pole at the point $z = 1$.

34.2 Why is this important?

Now what is the importance of this function? What does it give us? The answer is that it essentially captures the factorization of integers in it.

Every n can be factorized as a product of primes $p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$ and hence

$$\frac{1}{n^s} = \frac{1}{(p_1^{\alpha_1})^s (p_2^{\alpha_2})^s \cdots (p_k^{\alpha_k})^s}$$

And therefore, every term in the zeta function can be written as such a term on the RHS. And therefore,

$$\sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \in \text{Primes}} \left(1 + \frac{1}{p^s} + \frac{1}{p^{2s}} + \cdots\right) = \prod_{p \in \text{Primes}} \left(\frac{1}{1 - \frac{1}{p^s}}\right)$$

This is not a rigerous proof; mathematicians will stand on their head and complain that such infinite sum/product manipulation is an unforgivable sin. However, the final equality is true; can be made rigerous.

This relation between the zeta function and the prime product is one of the many things that make it important.

34.3 The Riemann Hypothesis

The analytically continued function has a lot of roots in $\mathbb{C}$. Infact, it has a root at every negative integer. These are considered as trivial roots since they provide us with no consequence. It has been shown that all the non-trivial zeroes lie in the critical strip of $\{s : 0 < \Re(s) < 1\}$, the strip between the y-axis and the line $x = 1$.

The *Riemann Hypothesis* is the conjecture that all the non-trivial zeroes of the zeta function lie on the line $x = \frac{1}{2}$. That is, any non-trivial root s must

satisfy $\Re(s) = \frac{1}{2}$.

As of now, all the roots that have been discovered lie on this line. But we do not have a way of proving, or disproving, that all the non-trivial roots lie on this line.

34.4 The Extended Riemann Hypothesis

This is a slight generalization of the zeta function. A character χ is a periodic[11] function is multiplicative. That is, $\chi(mn) = \chi(m)\chi(n)$.

For example, $\chi = \left(\frac{a}{n}\right)$ for a fixed a is a character.

The generalized zeta function is the analytic continuation of

$$L(\chi, s) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$$

Clearly, when $\chi(n) = 1$ for all n, this is just the zeta function.

The Extended Riemann Hypothesis is the conjecture that for any character χ, all the non-trivial zeroes of the L function lie on the line $\Re(s) = \frac{1}{2}$.

34.5 Derandomizing using ERH

One of the major consequences of ERH, that is used in a lot of places is that, for any prime p, the first quadratic non-residue is less than $O(\log^2 p)$.

So, in essence, the witness for the Solovay-Strassen test can be found without going too far. One just need to go till up to polylog(n) to get a witness. This thus derandomizes the Solovay-Strassen test.

As remarked earlier, the RH or the ERH is a really strong conjecture. One doesn't need to go as far as the ERH to get a primality test. Agrawal-Kayal-Saxena show that primality can be solved in deterministic polynomial time without using any high-end mathematical machinery. The give a very elegant and simple algorithm.

35 The AKS Primality Test: The Idea

All primality tests have the same form:

[11] there exists a number k such that $\chi(n + k) = \chi(n)$ for all n

1. Find a property such that it is satisfied only by primes.

2. Try to verify that property

In the Solovay-Strassen test, it was the properties of the Legendre symbol that we checked. The following proposition is the property used in the AKS test.

Proposition 35. *The equation "$(X + a)^n = X^n + a \pmod n$" is true for all a if and only if n is prime.*

Proof. When n is a prime, then the above equation is just the binomial theorem for prime numbers. Therefore the equation is indeed true for all a when n is a prime.

Suppose n is not a prime, then we must show that the two polynomials $(X + a)^n$ and $X^n + a$ are not the same. Suppose $n = pq$ where p is a prime and $q \neq 1$, then look at the following binomial term:

$$\binom{n}{p} = \frac{n \cdot (n-1) \cdot (n-2) \cdots (n-(p-1))}{p \cdot (p-1) \cdot (p-2) \cdots 1}$$

Let p^m be the largest power of m that divided n. The only term in the numerator that is divisible by p is the first term since it is a product of p consecutive integers and therefore only one of the can be divisible by p. Therefore the largest power or p that divides the numerator is p^m since it divides n and none of the other terms are divisible by p. The denominator has a factor of p and therefore will cancel off one factor of p from the numerator. Therefore the largest power of p that divides $\binom{n}{p}$ is p^{m-1}. And since $p^m \nmid \binom{n}{p}$, clearly $n \nmid \binom{n}{p}$ and therefore this terms survives.

Since $p \neq n$, this isn't the last term in the binomial series. Therefore, the two polynomials are different. □

There is one major problem here - how do we check if the two polynomials are the same? We can compute them by repeated squaring. But note that the polynomial $(X + a)^n$ has n terms. Checking if every term other than the first and the last is zero would take atleast $O(n)$ time adn therefore will be exponential in the input size which is $\log n$. We are looking for an algorithm that runs in $(\log n)^{O(1)}$ time.

The natural fix to this is computing the equation modulo a polynomial of small degree. Instead of computing $(X + a)^n \bmod n$, we compute $(X + a)^n \bmod (X^r - 1), n$ where $r \leq \log^c n$.

If n was a prime, then $(X+a)^n = X^n+a \bmod n$ and therefore $(X+a)^n = X^n + a \bmod (X^r - 1, n)$. The tricky part is the converse. It is very well possible that some composite number could satisfy this equation since we are going modulo a very small degree polynomial. What AKS does here is try this for different a's; they check if $(X + a)^n = X^n + a \bmod (X^r - 1, n)$ for a fixed r and $a \in (1, 2, \cdots, l)$ where $l \leq \log^{c'} n$. They then argue that if all these tests go through, n has to be a prime or a power of a prime.

Checking if n is a power of a prime is easy and that would conclude the algorithm.

35.1 Checking if $n = p^k$ for $k \geq 2$

Now $p \geq 2$ and therefore k can be at most $\log n$. Suppose we fix a k and want to find if $n = p^k$ for some p, how do we do that? Just a binary search.

Try $(n/2)^k$. If this is equal to n, you already got it. If it is less than n, then you know that if an a exists, it must be greater than $n/2$. Recurse on that half.

Thus, the number of steps is $\log^2 n$, which is polynomial in the input size.

We need to understand some properties of *cyclotomic extensions* over finite fields. We shall do those next time and that should fix the choice of l, r and other constants that come in the process.

CS681 **Computational Number Theory**

Lecture 23 and 24 : The Cyclotomic Polynomial

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

Overview

We started looking at the AKS primality test in the last class. The idea was to check if $(X+a)^n - X^n - a = 0 \pmod{n, X^r - 1}$ for many values for a. We first need to understand the polynomial $X^r - 1$ and extensions associated with it.

36 The polynomial $X^r - 1 = 0$

The roots of this polynomial are the r-th roots of unity. Over $\mathbb{Q}$ these are the complex numbers $e^{\frac{2\pi i}{r}}$ where $0 \leq i \leq r-1$.

Note that the roots of unity form a group under multiplication. If ζ_1 and ζ_2 are roots, then so is $\zeta_1\zeta_2$. Infact, it forms a cyclic group. And therefore, we can talk about a generator of this group.

Definition 25. *An r-th root of unity ζ is called a primitive r-th root of unity if ζ is a generator of the group of roots.*

Or in other words, ζ is a number such that $\zeta^r = 1$ and $\zeta^t \neq 1$ for any $t < r$.

Now, let ζ be a primitive r-th root of unity. What are the other primitive roots? Any other root ζ' can be written as ζ^t. Suppose the $\gcd(r,t) = d$ then look at $\zeta'^{r/d}$. Since d divides r, the exponent is an integer.

$$\zeta'^{r/d} = \zeta^{tr/d} = (\zeta^r)^{t/d} = 1$$

Therefore, the primitive r-th roots of unity are ζ^t where t is coprime with r. And therefore, there are $\varphi(r)$ of them. Let us consider the following polynomial:

$$\Phi_r(X) = \prod_{\zeta \text{ primitive}} (X - \zeta)$$

This is called the r-th cyclotomic polynomial. Clearly, the degree of $\Phi_r(X)$ is $\varphi(r)$.

36.1 Cyclotomic Polynomials over $\mathbb{Q}$

Let us restrict our attention to the field of rational numbers. We want to study $\prod(X - \zeta)$. The first important property is the following:

Proposition 36. *The coefficients of $\Phi_r(X)$ are rational numbers.*

Proof. The idea is quite simple. The equation $X^r - 1$ has all the r-th roots of unity as roots and certainly includes the primitive roots as well. Therefore, $\Phi_r(X)$ divides $X^r - 1$. So the idea is to eliminate all the non-primitive roots of unity.

Note that since the roots of unity form a group under multiplication, for any root of unity ζ the order of ζ divides r. So the order could either be r or some proper factor of r. We are interested in only the ζs whose order is equal to r.

For every ζ, if the order of ζ is t, then $t \mid r$ and ζ is a root of $\Phi_t(X)$ by definition. And running over all the roots of unity, we have

$$X^r - 1 = \prod_{d|r} \Phi_d(X)$$

And therefore,

$$\Phi_r(X) = \frac{X^r - 1}{\prod_{d|r, d \neq r} \Phi_d(X)}$$

By induction, if we assume that $\Phi_d(X)$ has rational coefficients, for all $d < r$, it follows that $\Phi_r(X)$ has rational coefficients too. □

To see a few examples,

1. $\Phi_1(X) = X - 1$
2. $\Phi_2(X) = \frac{X^2-1}{X-1} = X + 1$
3. $\Phi_3(X) = \frac{X^3-1}{X-1} = X^2 + X + 1$
4. $\Phi_4(X) = \frac{X^4-1}{(X+1)(X-1)} = X^2 + 1$

And if you notice, the coefficients are not only rationals but infact integers. This can be got from the earlier proof using the following very useful lemma.

Lemma 37 (Gauss Lemma)**.** *Let f be a monic polynomial with integer coefficients. If f is irreducible in $\mathbb{Z}$, that is there are no polynomials g, h with integer coefficients such that $f(X) = g(X)h(X)$, then f is irreducible over $\mathbb{Q}$ as well.*

Therefore if f and g are integer monic polynomials such that there g divides f over $\mathbb{Q}$, that is there exists a polynomial h with rational coefficients such that $f(X) = g(X)h(X)$, the g infact divides f over $\mathbb{Z}$ or h infact has only integer coefficients. Thus all the above divisions will only yeild integer polynomials and hence $\Phi_r(X)$ is an integer polynomial.

Another important property of cyclotomic polynomials is that they are irreducible over $\mathbb{Q}$. We shall prove this soon. But what's important is that it needn't be so in the case of finite fields. For example, if $r = p - 1$ and we looked at $\Phi_r(X)$ in $\mathbb{F}_p$. Note that $\Phi_r(X)$ is a factor of $X^r - 1 = X^{p-1} - 1$ which inturn is a factor of $X^p - X$ and this completely splits.

However, if we can show that $\Phi_r(X)$ was irreducible over some prime p, then it has to be irreducible over $\mathbb{Q}$. Because if it were reducible as $f(X) \cdot g(X)$ over $\mathbb{Q}$, then we can reduce the equation $\Phi_r(X) = f(X)g(X) \bmod p$ and get a factorization in $\mathbb{F}_p$.

36.2 Cyclotomic polynomials over $\mathbb{F}_p$

Let ζ be a primitive r-th root of unity over $\mathbb{F}_p$. Note that ζ could very well be in $\mathbb{F}_p$ itself; when $r = p - 1$ for example. In any case, consider the field extensio $\mathbb{F}_p(\zeta)$ over $\mathbb{F}_p$ by just adjoining ζ to $\mathbb{F}_p$. Let us say the degree of this extension $[\mathbb{F}_p(\zeta) : \mathbb{F}_p] = d$.

Recall that the degree of a field extension $[K(\alpha) : K]$ is the degree of the $K(\alpha)$ as vector space over K and therefore is equal to the degree of the minimum polynomial of α over K.

Therefore, if $\mu(X)$ is the minimum polynomial of ζ over $\mathbb{F}_p$, $\mu(\zeta) = a_0 + a_1\zeta + \cdots a_d\zeta^d = 0$ and therefore the set $1, \zeta, \zeta^2, \cdots, \zeta^{d-1}$ is the largest linearly independent subset. Therefore, $[\mathbb{F}_p(\zeta) : \mathbb{F}_p] = \deg \mu(X) = d$.

Now, ζ is a root. What about the other roots of this polynomial? Recall our old friend Fröbenius. The automorphism $a \mapsto a^p$ fixes every element in $\mathbb{F}_p$.

$$\begin{aligned}
\mu(\zeta) &= a_0 + a_1\zeta + \cdots + a_d\zeta^d = 0 \\
\implies (\mu(X))^p &= 0 \\
&= \left(a_0 + a_1\zeta + \cdots + a_d\zeta^d\right) \\
&= a_0^p + a_1^p\zeta^p + \cdots + a_d^p\zeta^{dp} \\
&= a_0 + a_1\zeta^p + \cdots + a_d(\zeta^p)^d \\
&= \mu(\zeta^p)
\end{aligned}$$

and hence, ζ^p is also a root. Applying the map again, we can show $\zeta, \zeta^p, \zeta^{p^2}, \cdots, \zeta^{p^{d-1}}$ are all roots of $\mu(X)$.

One can't go up to more than d such applications because the degree of μ is bounded by d and therefore can have only d roots. And if we were to apply the Fröbenius map again, then we would end up in ζ again. Or in other words,

$$\zeta^{p^d} = \zeta \implies \zeta^{p^d-1} = 1 \implies r \mid p^d - 1 \implies p^d = 1 \bmod r$$

Since d was the first place where this sort of wraparound happened, d is the least such number such that $p^d = 1 \bmod r$ which implies that d is the order of p modulo r. This is denoted by $d = ord_r(p)$.

Thus, the degree of the the minimum polynomial of ζ over $\mathbb{F}_p$ for any primitive r-th root is $ord_r(p)$. And since we just specified ζ as any primitive root, it follows that every primitive root has the degree of its minimum polynomial as $ord_r(p)$.

But with a little thought, this will tell us that the approach that $\Phi_r(X)$ is irreducible in $\mathbb{F}_p$ for some p may not work. Suppose this was true, then the minimum polynomial of ζ must infact be $\Phi_r(X)$. And from what we have proved above, the degree of this polynomial must be $ord_r(p)$ and therefore $\varphi(r) = ord_r(p)$. Now notice that if we consider $(\mathbb{Z}/r\mathbb{Z})^\star$, then the size of this group is $\varphi(r)$ and if $ord_r(p) = \varphi(r)$, then p generates the group $(\mathbb{Z}/r\mathbb{Z})^\star$. But not all $(\mathbb{Z}/n\mathbb{Z})^\star$s are cyclic and therefore such a p may not exist at all.

But let us just take it on faith that the cyclotomic polynomial is irreducible. The proof is not hard but would be a significant digression. The interested reader can look it up online or on any abstract algebra text. But what is important that any primitive r-th root has a minimum polynomial of degree $ord_r(p)$ in $\mathbb{F}_p$.

37 AKS Primality Test: A sketch

Now we have enough machinary to go ahead with the primality test. The algorithm would be the following:

1. Find two numbers r and l based on some requirements
2. Do some small preprocessing
3. For all $1 \leq a \leq r$, check if the following identity holds

$$(X + a)^n = X^n + a \pmod{n, X^r - 1}$$

Use repeated squaring to evaluate the LHS and RHS and check.

If any of the test failed, output COMPOSITE.

4. If all the above tests succeeded, output PRIME

Our preprocessing steps would be the following:

- Check if n is a perfect power of some number. If it is some non-trivial power, output COMPOSITE. This will rule out the case that $n = p^k$ for $k \geq 2$.
- For each $2 \leq d \leq r$, check if $\gcd(n, d) = 1$. If you find a factor, output COMPOSITE.

And the parameters will be fixed soon and we shall see that both r and l are less than $(\log n)^c$ for some constant c. Therefore, the preprocessing steps take only polylog time, checking if the identity holds for each a in that range also takes polylog time and therefore the entire algorithm runs in time polynomial in $\log n$.

Further, if n was indeed a prime, the algorithm would definitely output PRIME. The tricky part is to show that if n had atleast 2 distinct prime factors, the algorithm will catch it.

To prove the correctness of this algorithm, we would be studying 2 rings:

- $R = \frac{\mathbb{Z}[X]}{(n, X^r - 1)} = (\mathbb{Z}/n\mathbb{Z}[X]) / (X^r - 1)$
- $R = \frac{\mathbb{Z}[X]}{(p, h(X))} = (\mathbb{Z}/p\mathbb{Z}[X]) / (h(X))$ where p is a prime factor of n and $h(X)$ is an irreducible factor of $\Phi_r(X)$ over $\mathbb{F}_p$. Note that we have shown that $t = \deg(h(X)) = ord_r(p)$.

Suppose the algorithm said n was a prime even when it had p as a prime factor of n. What we would do is construct a group $\mathcal{G}$ and get bounds on the size of $\mathcal{G}$ in two different ways. We will show that $|\mathcal{G}| \geq \binom{t+l}{t-1}$. And also, if n had at least 2 distinct prime factors, then $|\mathcal{G}| \leq n^{\sqrt{t}}$. Thus, unifying the two, we have

$$\binom{t+l}{t-1} \leq \mathcal{G} \leq n^{\sqrt{t}}$$

And then, with suitable choice of r and l, we shall show that the lower bound is larger than the upper bound which would give the required contradiction. That is the general idea.

We shall see this in detail in the next class.

CS681	Computational Number Theory

Lecture 25 : The AKS Primality Test

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

We shall do the AKS primality test this class.

38 The Deterministic Primality Test

The algorithm is going to be the following:

1. Find two numbers r and l based on some requirements
2. Check if n is a perfect power. If it is, output COMPOSITE.
3. Check if any of the numbers less than l have a non-trivial gcd with n. If they do, output COMPOSITE.
4. For all $1 \leq a \leq l$, check if the following identity holds

   $$(X + a)^n = X^n + a \pmod{n, X^r - 1}$$

   Use repeated squaring to evaluate the LHS and RHS and check.

   If any of the test failed, output COMPOSITE.
5. If all the above tests succeeded, output PRIME

We shall figure out what the requirement of r,l as we go along. That would give a better picture of why we want those properties.

The algorithm will be in a way that if n was indeed a prime then the algorithm would answer correctly. We only need to make sure that the algorithm doesn't make a mistake on composite n. We shall ensure that if a composite n passes all the tests, then n has to be a power of a prime. And since we saw that testing of a number was a perfect power is easy, that is one of our preprocessing steps and hence such n will be eliminated. We shall now choose parameters in a way that no composite n can pass all the test.

Let us assume that n is composite, has p as a proper divisor and is not a power of p. And let us assume that n passes all the tests. Note that p has to

be larger than l since we have already made sure that n does not have any small factor in step 3.

The test is basically checking the identity $(X+a)^n = X^n+a \pmod{n, X^r-1}$, which is just checking if two terms in the ring $R = \mathbb{Z}[X]/(n, X^r-1) = \frac{\mathbb{Z}/n\mathbb{Z}[X]}{(}X^r - 1)$. Rings are a little hard to handle; would be easier to go to a field.

38.1 Moving to a Field

Instead of looking at $R = \frac{\mathbb{Z}/n\mathbb{Z}[X]}{(X^r-1)}$, we shall look at the field $\mathbb{F}_p[X]/(h(X))$ where $h(X)$ the minimal polynomial of the primitive r-th root of unity. Note that this just starting with R, going modulo the ideal generated by p, then going modulo the ideal generated by $h(X)$. The ideal generated by p is a prime ideal in R and therefore $R/(p)$ became an integral domain. Then $h(X)$ generates a maximal ideal being irreducible, and therefore we get the field $\mathbb{F} = \mathbb{F}_p[X]/(h(X))$.

The important thing to note here is that, if equations were satisfied in R, they have to be satisfied in $\mathbb{F}$ as well as we are just going modulo some ideals. This is like saying, if $a = b$ in Z, then of course they are equal $\mod p$ as well. Thus let us work with this field. Now the next thing to note that in $\mathbb{F}_p[X]/(h(X))$, it is just $\mathbb{F}_p(\zeta)$ and X plays the role of ζ. Thus, if some n survived all the tests, then for each $1 \leq a \leq l$

$$(X + a)^n = X^n + a \bmod p, h(X)$$

Let us characterize this property of n by the following definition.

Definition 26. *A number m is called introspective for a polynomial f if*

$$f(X)^m = f(X^m) \quad \textit{in } \mathbb{F}$$

As the name suggests, it says that m is curious with respect to X; such powering can be pulled inside the brackets. The tests basically mean that n is introspective for $f(X) = (X + a)$. The following two observations are trivial.

Observation 38. *If m_1 and m_2 are introspective for f, then so is $m_1 m_2$.*

Observation 39. *If m is introspective for f and g, then m is introspective for fg.*

Since n passed the tests, we know that n is introspective for $(X + a)$ for all $1 \leq a \leq l$. And also note that, by the binomial theorem in $\mathbb{F}_p[X]$,

$$(X + a)^p = X^p + a$$

for any a. Therefore, p is introspective for $(X + a)$ as well. Therefore, by the two observations, and $n^i p^j$ is introspective for $\prod_{1 \leq a_i \leq l}(X + a_i)$. Let us create two groups to capture this property.

38.2 The two groups

Let G be the group generated by n and p modulo r. Or in other words, G is the set of numbers of the form $n^i p^j$ modulo r.

Similarly, let $\mathcal{G}$ be the group generated by $\{(X + a) \ : \ 1 \leq a \leq l\}$ in $\mathbb{F}$. And by the two observations, any element of G is introspective to any element of $\mathcal{G}$.

$$\begin{aligned} G &= \langle n, p \rangle \subseteq (\mathbb{Z}/r\mathbb{Z})^\star \\ \mathcal{G} &= \langle \{(X + a) \ : \ 1 \leq a \leq l\} \rangle \subseteq \mathbb{F} \end{aligned}$$

Let $|G| = t$. Since G is a subgroup of $(\mathbb{Z}/r\mathbb{Z})^\star$, $t \leq r$. Now note that the group generated by n is a subgroup of G and its cardinality is $ord_r(n)$. Therefore $t > ord_r(n)$.

We shall now get two bounds on the size of $\mathcal{G}$.

38.3 An Upper Bound on $|\mathcal{G}|$

Not that every element of $\mathcal{G}$ is actually a product of $(\zeta + a)$'s since X is ζ in $\mathbb{F}$. In order to get a bound on the size of $\mathcal{G}$ let us restrict ourselves to just products of distinct $(\zeta + a)$'s. Set $l = t - 1$.

For every subset K of $1, 2, \cdots l$, we can construct a polynomial $f_K(X) = \prod_{i \in K}(X + i)$. And there are 2^t such polynomials. These polynomials are clearly distinct as each of them has a different set of roots. But what happens if we substitute ζ in them? Is it possible that for $K \neq K'$, $f_K(\zeta) = f_{K'}(\zeta)$?

Suppose they were equal, then note that $f_K(\zeta)^m = f_{K'}(\zeta)^m$ for any m, and in particular any $m \in G$. If m is in G, then $f_K(\zeta^m) = f_K(\zeta)^m = f_{K'}(\zeta)^m = f_{K'}(\zeta^m)$. Thus, if $g(X) = f_K(X) - f_{K'}(X)$, then ζ^m is a root of g for every $m \in G$. But since $|G| = t$, this means that $g(X)$ has t roots. But $g(X)$ is a polynomial of degree at most l which is less than t. Such a polynomial cannot have t roots unless it is the zero polynomial. Thus,

$f_K(\zeta)$s are distinct. Since there are 2^{t-1} possible $f_K(X)$s possible, each of this would give a distinct $f_K(\zeta)$ in $\mathcal{G}$. Therefore,

$$|\mathcal{G}| \geq 2^{t-1}$$

38.4 A Lower Bound for $|\mathcal{G}|$

Look at the set $S = \left\{n^i p^j \ : \ 0 \leq i, j \leq \sqrt{t}\right\}$. And if n wasn't a power of p, this set S has $(1+\sqrt{t})^2 > t$ elements. Now, considering them modulo r, they are a subset of $|G|$ and by pigeon hole principle, there must be some $m_1 \neq m_2 \in S$ such that $m_1 = m_2 \bmod r$ and therefore $m_1 = m_2 + kr$. Then, if $f(\zeta) \in \mathcal{G}$

$$f(\zeta)^{m_2} = f(\zeta)^{m_1+kr} = f(\zeta^{m_1+kr}) = f(\zeta^{m_1}) = f(\zeta)^{m_1}$$

Thus, if we were to consider $g(X) = X^{m_1} - X^{m_2}$, then every $f(\zeta) \in \mathcal{G}$ is a root of $g(X)$. Note that degree of $g(X)$ is at most the max of m_1, m_2. And $m_1 = n^i p^j \leq n^{\sqrt{t}} n^{\sqrt{t}} = n^{2\sqrt{t}}$. And since the degree of g is is at most $n^{2\sqrt{t}}$, the number of roots can also be only that much. Therefore

$$|\mathcal{G}| \leq n^{2\sqrt{t}}$$

38.5 Conflicting Bounds

Combining the two bounds, we have

$$2^{t-1} \leq |\mathcal{G}| \leq n^{2\sqrt{t}}$$

Now if we can ensure that the lower bound is larger than the upper bound, we would get the contradiction we are looking for; that would rule out the possiblity that a composite number passed all the tests.

Thus we want $2^{t-1} > n^{2\sqrt{t}}$. Taking logs, $t - 1 > 2\sqrt{t} \lg n$. And if $t > 4 \log^2 n$, that should be to make the bounds for $|\mathcal{G}|$ contradict.

Thus all we need to do now is find an r so that $ord_r(n) > 4 \log^2 n$.

38.6 Getting an r such that $ord_r(n) > 4 \log^2 n$

What we shall do to get an r is just try $1, 2, \cdots$ until it is satisfied. But how long would we have to go until we hit a good r?

Look at the following number

$$B = \prod_{i=1}^{4\log^2 n} (n^i - 1) < n^{\log^4 n}$$

If r was a prime number not dividing this B, then the order of n modulo r cannot be less than $4\log^2 n$. The number of prime factors of B is at most $\log^4 n \log n = \log^5 n$. And therefore, the $\log^5 n+1$-th prime would definitely not divide B. Therefore, we can just enumerate all primes starting from 2 and get the $\log^5 n + 1$-th prime. And this search is assured to be within $\log^6 n$ by the prime number theorem. Therefore, we are in good shape.

39 The Final Algorithm

Thus, putting all the pieces together.

Algorithm 9 AKS PRIMALITY TEST

1: Check if the input number n is a perfect power of some number. If so, **return** COMPOSITE.
2: Find the least prime r such that $ord_r(n) > 4\log^2 n$. Let $l = ord_r(n) - 1$. In the process, check if any of those r's have a non-trivial factor with n. If yes, **return** COMPOSITE.
3: **for** $1 \leq a \leq l$ **do**
4: **if** $(X + a)^n \neq X^n + a \bmod (n, X^r - 1)$ **then**
5: **return** COMPOSITE.
6: **end if**
7: **end for**
8: **return** PRIME.

The total time complexity is about $O(\log^{12} n)$. The current best is about $O(\log^6 n)$.

CS681	**Computational Number Theory**
Lecture 26 : Hensel Lifting	
Instructor: Piyush P Kurur	*Scribe: Ramprasad Saptharishi*

We first saw two algorithms to factor univariate polynomials over finite fields. We shall now get into bivariate factoring over finite fields. Before that, we need to look at a very important and powerful tool called Hensel Lifting.

40 Hensel Lifting

The intuition behind Hensel Lifting is the following - you have a function for which you need to find some root. Suppose you have an x very close to a root x_0 in the sense that there is a small error. The question is how can you use x and the polynomial to get a closer approximation?

Recall the Newton Rhapson Method you might have done in calculus to find roots of certain polynomials. Let us say f is the polynomial and x_0 is our first approximation of a root. We would like to get a better approximation. For this, we just set $x_1 = x_0 + \varepsilon$. And by the Taylor Series,

$$\begin{aligned} f(x_1) = f(x_0 + \varepsilon) &= f(x_0) + \varepsilon f'(x_0) + \varepsilon^2 \frac{f''(x)}{2!} + \cdots \\ &= f(x_0) + \varepsilon f'(x_0) + O(\varepsilon^2) \end{aligned}$$

Ignoring the quadratic error terms, we want a better approximation. Thus, in a sense, we would want $f(x_1)$ to be very close to 0. To find the right ε that would to the trick, we just set $f(x_1) = 0$ and solve for ε. With just some term shuffling, we get

$$\epsilon = -\frac{f(x_0)}{f'(x_0)} \implies x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

But one crucial property we need here is that $f'(x_0)$ is not zero for otherwise division doesn't make sense. In the same spirit, we shall look at version 1 of the Hensel Lifting.

40.1 Hensel Lifting: Version 1

Theorem 40. *Let p be a prime and c and positive integer, and let f any polynomial. Suppose we have a solution x that satisfies*

$$f(x) = 0 \bmod p^c \qquad , \qquad f'(x) \neq 0 \bmod p$$

then we can "lift" x to a better solution $x^\star$ that satisfies

$$f(x^\star) = 0 \bmod p^{2c} \qquad , \qquad x^\star = x \bmod p^c$$

It is of course clear that if $f(x^\star) = 0 \bmod p^{2c}$ then $f(x^\star) = 0 \bmod p^c$ but the converse needn't be true. Therefore, $x^\star$ is a more accurate root of f. The proof of this is entirely like the proof of the Newton Rhapson Method.

Proof. Set $x^\star = x + hp^c$. We need to find out what h is. Just as in newton rhapson,

$$\begin{aligned} f(x^\star) = f(x + hp^c) &= f(h) + hp^c f'(x) + (hp^c)^2 \frac{f''(x)}{2!} + \cdots \\ &= f(h) + hp^c f'(x) + O((hp^c)^2) \\ &= f(h) + hp^c f'(x) \bmod p^{2c} \end{aligned}$$

Since we want $f(x^\star) = 0 \bmod p^{2c}$, we just set the LHS as zero and we get

$$h = \frac{f(x)}{p^c f'(x)}$$

Note that $f(x) = 0 \bmod p^c$ and therefore it makes sense to divide $f(x)$ by p^c. Thus our $x^\star = x + hp^c$ where h is defined as above and by definition $x^\star = x \bmod p^c$. □

Another point to note here is that since $x^\star = x \bmod p^c$, $f(x^\star) \neq 0 \bmod p$ as well. Therefore, we could lift even further. And since the accurace doubles each time, starting with $f(x) = 0 \bmod p$, i lifts will take us to an $x^\star$ such that $f(x^\star) = 0 \bmod p^{2^i}$.

Hensel Lifting allows us to get very good approximations to roots of polynomials. The more general version of Hensel Lifting plays a very central role in Bivariate Polynomial Factoring.

40.2 Hensel Lifting: Version 2

In the first version of the Hensel Lifting, we wanted to find a root of f. Finding an α such that $f(\alpha) = 0 \bmod p$ is as good as saying that we find a factorization $f(x) = (x - \alpha)g(x) \bmod p$. And also, the additional constraint that $f'(\alpha) \neq 0 \bmod p$ is just saying that α is not a repeated root of f or in other words $(x - \alpha)$ does not divide g. With this in mind, we can give the more general version of the Hensel Lifting.

Theorem 41. *Let R be a UFD and $\mathfrak{a}$ any ideal of R. Suppose we have a factorization $f = gh \bmod \mathfrak{a}$ with the additional property that there exists $s, t \in R$ such that $sg + th = 1 \bmod \mathfrak{a}$. Then, we can lift this factorization to construct $g^\star, h^\star, s^\star, t^\star$ such that*

$$\begin{aligned} g^\star &= g \bmod \mathfrak{a} \\ h^\star &= h \bmod \mathfrak{a} \\ f &= g^\star h^\star \bmod \mathfrak{a}^2 \\ s^\star g^\star + t^\star h^\star &= 1 \bmod \mathfrak{a}^2 \end{aligned}$$

Further, for any other g', h' that satisfy the above four properties, there exists a $u \in \mathfrak{a}$ such that

$$\begin{aligned} g' &= g^\star(1 + u) \bmod \mathfrak{a}^2 \\ h' &= h^\star(1 - u) \bmod \mathfrak{a}^2 \end{aligned}$$

Therefore, the lifted factorization in some sense is unique.

Proof. (sketch) Set $g^\star = g + te$ and $h^\star = h + se$. Now solve for e and that should do it. Finding $s^\star, t^\star$ is also similar. (painful!) □

Here is a more natural way is to look at this. What we want is a solution to the curve $XY = f$ where f is the function we want to factorize. Let us call $F(X, Y) = f - XY$. We have X, Y as solutions such that $F(X, Y) = f - XY = e$. Now

$$\begin{aligned} F(X + \Delta X, Y + \Delta Y) &= f - (X + \Delta X)(Y + \Delta Y) \\ &= f - XY - (X\Delta Y + Y\Delta X) + O(\Delta^2) \\ &= F(X, Y) - (X\Delta Y + Y\Delta X) \\ &= e - (X\Delta Y + Y\Delta X) \end{aligned}$$

Further, we also know that $sX + tY = 1$ and therefore, if we just set $\Delta X = se$ and $\Delta Y = te$, we have

$$F(X + \Delta X, Y + \Delta Y) = e - e(sX + tY) = 0 \bmod \Delta^2$$

One should also be able to look at the lifts of s and t as solving appropriate equations. In the next class, we shall look at this technique put to use in Bivariate Factorization.

CS681	Computational Number Theory

Lecture 27 : Bivariate Factoring

Instructor: Piyush P Kurur *Scribe: Ramprasad Saptharishi*

We shall now look at Bivariate Factoring. This is an instance where there are a lot of corner cases that need to be handled but we shall make some preliminary assumptions on the polynomial. We shall soon see how we can ensure those properties as well.

41 Bivariate Factoring

We are given a polynomial $f \in K[X, Y]$ that we would like to factor. Before we go into factorization as such, we first need to figure out how to do the $\gcd$ algorithm over two variables.

41.1 The Euclidian Algorithm for $K[X, Y]$

Lets say we are given two polynomials $f(X, Y)$ and $g(X, Y)$. How do we compute their $\gcd$? Remember that when we had univariate factoring, the euclidian algorithm would divide. But over $K[X, Y]$, how do we do that?

Hence, instead of looking at these polynomials over $K[X, Y]$, we look at them as univariate polynomials rational functions over one variables, namely $K(X)[Y]$. Thus coefficients could now be of the form $p(X)/q(X)$. This then give rise to another problem, can the denominator keep getting larger? That is a serious issue since during the process, we could potentially be doubling the degree at each step

But clearly, the final answer cannot have a really large X-degree; it clearly must be bounded by the X degrees of f and g. So what we do instead is do these computations modulo a number of irreducible polynomials. We just need to make sure that the product of all these irreducible polynomials have degree larger than the X degree of both f and g.

This would then let us compute the $\gcd$ of two bivariate polynomials.

Now we can think of f as an element in $K[X][Y]$, a polynomial in Y each of whose coefficients are polynomials in X. The first things we do is pull out the gcd of all the coefficients. And we also express f as an element of $K[Y][X]$ and repeat the same thing. Then we shall assume that f is

squarefree. This needs justification but we shall do this later. But for the moment let us assume that this can be done and proceed.

Now we have $f(X,Y) \in K[X][Y]$. We shall put $Y = 0$ to get a univariate polynomial in X. Since we have already done factoring of univariate polynomials, we shall factorize $f(X,0)$ into coprime factors $f(X,0) = g_0 h_0$.

Note that this is just saying

$$f = g_0 h_0 \bmod Y$$

since $Y = 0$ is precisely taking $\bmod Y$. And we further have the property, because the factors are coprime[12] that $sg_0 + th_0 = 1 \bmod Y$. Thus we can Hensel Lift this factorization for k steps to get

$$f = g_k h_k \bmod Y^{2^k}$$

We would now argue that after sufficient hensel lifting, we can retrieve a factor of f.

Claim 42. *If f was not irreducible, hen there is an irreducible factor of f, say g such that $g = g_0 l_0 \bmod Y$*

Proof. This is obvious. Suppose f factorizes into irreducible factors as $f_0 f_1 \cdots f_l$. Then clearly,

$$f = f_0 f_1 \cdots f_l \bmod Y$$

And therefore g_0 must divide one of these factors and whatever factor it may be, set that as g. Therefore, $g = g_0 l_0 \bmod Y$ □

Now start with $g = g_0 l_0 \bmod Y$ and lift this to k steps. What we know is that $\deg_X(g) < \deg_X(f)$ and $\deg_Y(g) \leq \deg_Y(f)$. What we have is just $f = g_k h_k \bmod Y^{2^k}$. We shall use this to try and get hold of a factor.

Suppose we look at the following equation to solve:

$$\begin{aligned} \tilde{g} &= g_k \tilde{l} \bmod Y^{2^k} \\ \deg_X(g) &< \deg_X(f) \\ \deg_Y(g) &\leq \deg_Y(f) \end{aligned}$$

Firstly, does this system of equations have a solution? Indeed; we just say that $g = g_0 l_0 \bmod Y$ can be lifted to k steps to give $g = g_k l_k \bmod Y^{2^k}$

[12] we shall get back to this

and clearly g and l_k satisfy the system of equations. Therefore, we are guaranteed that at least one such solution exists if f is not irreducible. Now, how do we compute such a solution?

Notice that we already know g_k and once we have g_k, if we think of the coefficients of $\tilde{g}$ and $\tilde{l}$ as variables, the above is just a system of linear equations. Thus, one could solve that system using gaussian elimination to get some solution $\tilde{g}$. It might very well be possible that $\tilde{g} \neq g$. But from a solution $\tilde{g}$, how do we retrieve a factor?

Look at the two equations $f = g_k h_k \bmod Y^{2^k}$ and $\tilde{g} = g_k \tilde{l} \bmod Y^{2^k}$. And in essence, both f and $\tilde{g}$ share a common factor modulo Y^{2^k}. Therefore, if one were to consider them as polynomials of X with coefficients coming from $K[Y]$, this means that the X-resultant of f and $\tilde{g}$ is zero modulo Y^{2^k}. By the X resultant we mean the resultant, considered as polynomials in $K[Y][X]$.

Therefore

$$\text{Res}_X(f, \tilde{g}) = 0 \bmod Y^{2^k}$$

The resultant would be a polynomial in Y. What is the maximum degree of this polynomial possible? Since we are looking at the determininant, the size of the sylvester matrix is $2m$ where m is the $\deg_X(f)$. And each element of this matrix would be a polynomial in Y of degree at most n where $n = \deg_X(f)$. Therefore, the degree of the resultant polynomial is at most $2mn + 2$. And if 2^k is larger than $2mn + 2$, then

$$\text{Res}_X(f, \tilde{g}) = 0 \bmod Y^{2^k} \implies \text{Res}_X(f, \tilde{g}) = 0$$

and therefore, f and $\tilde{g}$ must infact share a common factor. Use the euclidian algorithm to extract out this factor.

All that is left to do are the corner cases. We shall deal with them in the next class.