A CANTOR-ZASSENHAUS TYPE ALGORITHM FOR POLYNOMIAL FACTORING OVER FINITE FIELDS

HIMANSHU SHUKLA
MENTORS: DR. NITIN SAXENA & DR. RAJAT MITTAL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Abstract. Cantor-Zassenhaus [1] algorithm is a randomized algorithm to factor polynomials over finite fields. We give a Cantor-Zassenhaus type randomized algorithm to get pseudo factors of polynomials over finite fields and use Extended Riemann's Hypothesis (ERH) to get factors of the polynomial.

1. Intorduction

In Number Theory at times to get intuition about what is happening in $\mathbb{Z}$ or $\mathbb{F}_p$ where p is a prime, we move to the suitable Galois' extensions K of $\mathbb{Q}$ and look at the primes of $\mathbb{Z}$ in the number ring O_K of K. One of the most interesting Galois' extensions of $\mathbb{Q}$ are nth cyclotomic extensions of Q. A nth cyclotomic extension is the smallest Galois' extension of $\mathbb{Q}$ containing all the nth roots of unity. The irreducible cyclotomic polynomial has a lot of nice properties and we use the way they factor over $\mathbb{F}_p[X]$ and few simple properties of Ring theory to come up with a Cantor-Zassenhaus type algorithm. We first get a pseudo factor of the polynomial over the factor ring $R_r = \frac{\mathbb{F}_p[Y]}{(\Phi_r(Y))}$. and then from these pseudo factors we get factors of the polynomial using ERH.

In Section 2 we give the algorithm and in section 3 we give the intuition and a formal understanding of the algorithm and why does it give the factors of the polynomial at hand. In Section 4 we state two conjectures and in section 5 we talk about few of the emperical results and a informal justification behind the emperical results. In section 6 we conclude the report.

2. C-Z type Algorithm

The input to the algorithm is polynomial which is an output of distinct degree factorisation algorithm. We show the algorithm for $f(X)$ which completely splits in $\mathbb{F}_p[X]$ for other polynomials one can move to degree d extension of $\mathbb{F}_p$ and take the extensions of this d degree extension instead of $\mathbb{F}_p$. The pre-processing involved in getting the polynomial f is as follows.

- Given a polynomial $g(X) \in \mathbb{F}_p[X]$. get a polynomial $g'(X)$ such that $g'(X)$ is square free and contains the same irreducible factors as $g(X)$.
- Then run DDF (Distinct Degree Factorisation) algorithm over $g'(X)$ and get $f(X)$ which contains all the linear factors of $g'(X)$.

```
Result: This algorithm factors the given polynomial over finite field $\mathbb{F}_p[X]$
Input: polynomial $f(X)$ in $\mathbb{F}_p[X]$ with all irreducible factors of degree 1;
initialise $r = 2$;
while $r \leq log^2 p$ do
    $R_r := \frac{\mathbb{F}_p[Y]}{(\Phi_r(Y))}$;
    let $Y$ be the root of $\Phi_r(Y)$ in the ring $R_r$;
    let $q = p^{Ord_r(p)}$;
    Run the Euclidean GCD algorithm with respect to X to compute
     $(f(X+Y), X^{\frac{q-1}{2}} - 1)$;
    if The algorithm goes through then
        if $(f(X+Y), X^{\frac{q-1}{2}} - 1) = 1$ OR $(f(X+Y), X^{\frac{q-1}{2}} - 1) = f(X+Y)$ then
            r=r+1;
            continue;
        else
            Output: $f_1 = \frac{f(X+Y)}{(f(X+Y), X^{\frac{q-1}{2}} - 1)}$ and $f_2 = (f(X+Y), X^{\frac{q-1}{2}} - 1)$;
            And run the whole algorithm recursively on $f_1$ and $f_2$ if their degree is not 1
        end
    else
        We will have a zero divisor let it be $A(Y)$;
        Now factor the ring using $A(Y)$ as, $R_r := R_r^1 \oplus R_r^2$ where $R_r^1 = \frac{\mathbb{F}_p[X]}{(A(Y), \Phi_r(Y))}$ and
         $R_r^2 = \frac{\mathbb{F}_p[X]}{\frac{\Phi_r(y)}{(A(Y), \Phi_r(Y))}}$
        Run the while loop again with $R_r = R_r^1$ and $R_r^2$;
    end
end
```

Algorithm 1: Our Algorithm

3. Understanding the algorithm

First of all we note that the factor ring R_r is not always an integral domain. Also note that its is a finite ring. Hence it is a finite field iff $\Phi_r(Y)$ is irreducible over $\mathbb{F}_p$. For this we look at the following lemma.

Lemma 1. *: Write $n = p^m * l$ such that $(p, l) = 1$ where p is a prime. Then the prime ideal pO_K where $K = \mathbb{Q}[\zeta]$ and ζ is nth primitive root of unity, splits with ramification index $e = \Phi(p^m)$ and inertial degree $f = Ord_n(p)$.*

Proof: This is theorem 26 in [3].

Lemma 2. *: Suppose $O_K = \mathbb{Z}[\theta]$ and let $f(X)$ be the irreducible polynomial of theta over $\mathbb{Z}[\mathbb{X}]$. If $f = \prod_{i=1}^{k} f_i^{e_i} mod(p)$, where p is prime. Then the ideal pO_K factorises as $\prod_{i=1}^{k} (f_i(\theta, p)^{e_i})$.*

Proof: This is theorem 5.5.1 in [4].

Theorem: *If $r < p$ then $\Phi_r(X) = \prod_{i=1}^{k} f_i(X)$, where $k = \frac{\phi(r)}{Ord_r(p)} f_i(X)$*

Proof: Proof of this theorem immediately follows from Lemma 1 and Lemma 2.
Hence one can write

$$R_r \cong \sum_{i=1}^{k} \frac{\mathbb{F}_p[X]}{(f_i(X))} \; R_r \cong \sum_{i=1}^{k} \mathbb{F}_q^i \cong \sum_{i=1}^{k} \mathbb{F}_q$$

where $q = p^{Ord_r(p)}$ and sum represents the direct sum. $\mathbb{F}_q^i$ is the finite extension of $\mathbb{F}_p$ of degree $Ord_r(p)$ containing $Ord_r(p)$, rth primitive roots of unity which satisfy the polynomial f_i and as all the finite fields of same order are isomorphic, hence $\mathbb{F}_q^i \cong \mathbb{F}_q$. where $\mathbb{F}_q$ contains the roots of polynomial $f_1(X)$.

In the next step of the algorithm we choose $Y \in R_r$, note that Y is a root of $\Phi_r(X)$ in R_r and run the Euclidean GCD algorithm to compute $gcd_X(f(X+Y, X^{\frac{q-1}{2}} - 1)$. Now there are two bad cases and one good case.

***Good Case*:** The gcd algorithm goes through and we get a non trivial factor of $f(X+Y)$. In this situation we output two polynomials f_1 and $f_2 \in R_r[X]$.

$$f_1 = \frac{f(X+Y)}{(f(X+Y), X^{\frac{q-1}{2}}}$$

$$f_2 = (f(X+Y), X^{\frac{q-1}{2}} - 1)$$

The factors which we get are pseudo factors as Y is not known. But using ERH we have polynomial time algorithm to compute factors of $f(X)$ from factors of $f(X+Y)$.

***Bad Case 1*:** If the gcd algorithm outputs a trivial gcd. In this case we change the value of r and continue.

***Bad Case 2*:** Suppose the gcd algorithm does not go through, this implies that we encounter a zero divisor say $A(Y)$. But then this means that $(A(Y), \Phi_r(Y))$ is non trivial and using this we factor the ring R_r as

$$R_r = R_r^1 \oplus R_r^2$$

where $R_r^1 = \frac{\mathbb{F}_p[Y]}{(G(Y))}$ and $R_r^2 = \frac{\mathbb{F}_p[Y]}{(\frac{\Phi_r(Y)}{G(Y)}}$ and $G(Y) = (A(Y), \Phi_r(Y))$.

Then we continue the same algorithm over both the components with the same r. So during the course of algorithm either one factors $f(X+Y)$ or will factor the ring R_r if we do not encounter a trivial gcd with respect to X. Else we change the value of r.

3.1. **Relation with roots of unity.** On a closer inspection of the algorithm we get the following, Y is a root of $\Phi_r(y)$ in R_r, and as $R_r \cong \sum_{i=1}^{k} \mathbb{F}_q$ we see that there exists and isomorphism such that Y under this isomorphism goes to $(\alpha_1, \alpha_2, \ldots, \alpha_k)$, where α_i is primitive rth root of unity and is one of the roots of polynomial f_1. All α_i's need not be distinct. Now we look at the gcd operations as they will look in this isomorphic ring of R_r. Under the isomorphism the following happens

$$f(X+Y) \mapsto f((X+\alpha_1, X+\alpha_2, \ldots, X+\alpha_k)) = (f(X+\alpha_1), f(X+\alpha_2), \ldots, f(X+\alpha_k))$$

$$X^{\frac{q-1}{2}} - 1 \mapsto (X^{\frac{q-1}{2}} - 1, X^{\frac{q-1}{2}} - 1, \ldots, X^{\frac{q-1}{2}} - 1)$$

Now if the gcd is a trivial gcd, then this means that $f(X+\alpha_i) \forall i \in [k]$ is a factor of $X^{\frac{q-1}{2}} - 1$ or $(f(X+\alpha_i), X^{\frac{q-1}{2}} - 1) = 1$, which means that the roots of $f(X+\alpha_i)$ are all quadratic residues in $\mathbb{F}_q$ or quadratic non-residues in $\mathbb{F}_q$ respectively $\forall i \in [k]$. Otherwise if

$$(f(X+\alpha_i), X^{\frac{q-1}{2}} - 1) = g_i(X) \in \mathbb{F}_q[X]$$

is non-trivial $\forall i \in [k]$ then we factor $(f(X+\alpha_1), f(X+\alpha_2), \ldots, f(X+\alpha_k))$ as

$$(f(X+\alpha_1, X+\alpha_2, \ldots, X+\alpha_k)) = (g_1(X), g_2(X), \ldots, g_k(X)) * (\tfrac{f(X+\alpha_1)}{g_1(X)}, \tfrac{f(X+\alpha_2)}{g_2(X)}, \ldots, \tfrac{f(X+\alpha_k)}{g_k(X)}).$$

Now in case the algorithms is stuck this means that we are not able to compute the gcd and hence we encounter an element of the form (β_i), $i \in [k]$ such that at least one of β_i is 0. Note that each of $\beta_i \in \mathbb{F}_q$.

Now if the algorithm does not factors f and terminates then this means that all the roots of $f(X+\alpha_i) \forall i \in [k] \forall r \in [log^2 p]$ are either quadratic residues or non-residues. $\mathbb{F}_q$. But conjecturally this cannot happen as the emperical results in [2] show. Hence this algorithm should factor the polynomial $f(X)$.

4. Conjecture

We state the following conjectures:

Conjecture 1: Let $r \in [log^p]$ and ζ_r be the rth primitive root of unity. Also let α, $\beta \in \mathbb{F}_p$ such that $\chi_p(\alpha) = \chi_p(\beta)$, where $\chi_p(.)$ is the quadratic character in $\mathbb{F}_q$. Then $\exists$ $r_o \in [log^2 p]$ such that $\chi_p(\alpha + \zeta_{r_o}) \neq \chi_p(\beta + \zeta_{r_o})$.

Conjecture 2: Let $r \in [log^p]$ and α, $\beta \in \mathbb{F}_p$ and $\alpha \neq \beta$. Then $< \{(\alpha + r) * (\beta + r) | r \in [log^2 p]\} >= \mathbb{F}_p$. Where $< \{\} >$ represents subgroup generated by the set $\{.\}$.

5. Emperical results

We wrote the code for Conjecture 2 for the primes till 10^6 and for 1000 random pairs (α, β). The empirical results show that it is correct with random (a, b). Also We checked this conjecture exhaustively for primes till 10^4. Further emperically checked the following statement:

Let α, $\beta \in \mathbb{F}_p$ and $\chi_p(\alpha) \neq \chi_p(\beta)$, then $< \{\Phi_r(\alpha) * \Phi_r(\beta) | r \in [log^2 p]\} >= \mathbb{F}_p$.

But this statement turns out to be false but in very rare cases in fact there may not be quadratic residue in the set $\{\Phi_r(\alpha) * \Phi_r(\beta) | r \in [log^2 p]\}$. One such case occurs when $p = 1009, r \in [99]$ and $(a, b) = (58, 87)$.

This statement turns out to be false because of the fact that degree of $\Phi_r(X)$ can be $\phi(r)$ which may be large and if α and β have very low order in $\mathbb{F}_p$. then the higher powers will be truncated, hence the set $\{\Phi_r(\alpha) * \Phi_r(\beta) | r \in [log^2 p]\}$ is much smaller than $log^(p)$.

6. Conclusion:

The algorithm which we present is helpful is giving greater insights into the problem of solving the problem of polynomial factoring over finite fields as the emperical results show. The future work on this will be to deterministically prove the conjecture 1, that will in essence solve the problem of polynomial factoring over finite fields.

References

[1] David G. Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36(154):pp. 587–592, 1981.

[2] Kundan Kumar. Deterministic polynomial factorisation over a finite fields. 2014.

[3] Daniel A. Marcus. *Number Fields*. Springer-Verlag, 1977.

[4] M. Ram Murty and Jody Esmonde. *Problems in Algebric Number Theory*. Springer-Verlag, 2005.