Final Coding Project Report for CS 487

Gabrielle Lu
x244lu@uwaterloo.ca

1 Introduction

Given three univariate polynomials $f(X), g(X), h(X)$ over a ring R with $h(X)$ having invertible leading coefficient, computing $f(g(X)) \bmod h(X)$ is the problem of MODULAR COMPOSITION. It is the backbone of many algorithms, including the fast methods for polynomial factorization.

1.1 Retrospection

1.1.1 Naive Algorithm

A simple natural algorithm to compute $f(g(X)) \bmod h(X)$ has two steps: first compute $f(g(X))$, then reduce modulo $h(X)$. This takes $O(n^2)$ operations because $f(g(X))$ has n^2 terms. However, we want a modular composition algorithm that takes about $O(n)$ operations.

1.1.2 Algorithm of Brent and Kung

The Brent and Kung algorithm[1] is the first modular polynomial composition algorithm to provide subquadratic complexity. It achieves an operation count of $O(n^{(\omega+1)/2})$, where ω is the exponent of matrix multiplication.

1.1.3 Algorithm of Kedlaya and Umans

In 2011, Kedlaya and Uman[2] gave an algorithm that works over any finite field, and has running times optimal up to lower order terms. They did this by giving a new algorithm for MULTIVARIATE MULTIPOINT EVALUATION with running times optimal up to lower order terms.

1.2 Outline

This report illustrates the algorithm of Kedlaya-Umans, and perform an implementation in Maple.

2 Preliminaries

2.1 Notations

In this report, we set R a commutative ring. Denote $R[X]$ by the ring of polynomials with coefficients in R. $\mathbb{F}_p$ denotes the finite field $\mathbb{Z}/p\mathbb{Z}$ and $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$ denotes the finite ring where $(\mathbb{Z}/r\mathbb{Z})[Z]$ is the ring of univariate polynomial with coefficients in $\mathbb{Z}/r\mathbb{Z}$ and $E(Z)$ is a monic polynomial.

2.2 Problem statements

2.2.1 Modular Composition of Polynomials

Given two univariate polynomials $f(X), g(X) \in R[X]$ together with the modulus $h(X) \in R[X]$ (with invertible leading coefficient), to compute $f(g(X)) \bmod h(X)$.

2.2.2 Multivariate Multipoint Evaluation of Polynomials

Given m-variate polynomial $f(X_0, \cdots, X_{m-1})$ over R of degree at most $d-1$ in each variable, and given $\alpha_i \in R^m$ for $i = 0, \cdots, N-1$, compute $f(\alpha_i)$ for $i = 0, \cdots, N-1$.

2.3 Useful facts

Definition 1. The map $\psi_{h,l}$ from $R[X]$ to $R[Y_0, \cdots, Y_{l-1}]$ is defined as follows. Given X^a, write a in base h, $a = \sum_{j=0}^{l-1} a_j h^j$, then the map $\psi_{h,l}$ is:

$$\psi_{h,l}(X^a) := Y_0^{a_0} Y_1^{a_1} \cdots Y_{l-1}^{a_{l-1}}$$

Example 1. Let $f(X) = X + 2X^2 + 3X^3 + 4X^4$. Then $\psi_{3,2}(f) = Y_0 + 2Y_0^2 + 3Y_1 + 4Y_0 Y_1$.

Lemma 1. For all integers $N \geq 2$, the product of the primes less than or equal to $16 \log N$ is greater than N.

3 Technical Section

3.1 The reduction

This section gives an algorithm that reduce the problem of MODULAR COMPOSITION to MULTIVARIATE MULTIPOINT EVALUATION, and a proof that this algorithm returns the desired result.

3.1.1 Algorithm

Algorithm 1

Input: $f(X)$ with degree at most $d-1$
$g(X)$ with degree at most $N-1$
$h(X)$ with degree at most $N-1$ and invertible leading coefficient

Output: $f(g(X)) \bmod h(X)$

---

0. Pick a $d_0 \in [2, d-1]$, and compute $m' = l$, $N' = N l d_0$ where $l = \lceil \log_{d_0} d \rceil$.
1. Compute $f' = \psi_{d_0,l}(f)$.
2. Compute $g_i(X) := g(X)^{d_0^i} \bmod h(X)$ for $i = 0, \cdots, l-1$.
3. Select N' distinct elements of R, $\beta_0, \cdots, \beta_{N'-1}$, whose differences are units in R. Compute $\alpha_{i,k} := g_i(\beta_k)$ for all $i = 0, \cdots, l-1$ and $k = 0, \cdots, N'-1$.

4. Compute $f'(\alpha_{0,k}, \cdots, \alpha_{l-1,k})$ for $k = 0, \cdots, N' - 1$.

5. Interpolate to recover $f'(g_0(X), \cdots, g_{l-1}(X))$ by using the values computed in step 4, it returns a univariate polynomial of degree less than N'.

6. Output the result modulo $h(X)$.

3.1.2 Proof

Correction follows from the observation that

$$\begin{aligned} & f'(g_0(X), \cdots, g_{l-1}(X)) \qquad \bmod h(X) \\ \equiv\ & f'(g(X)^{d_0^0}, \cdots, g(X)^{d_0^{l-1}}) \\ \equiv\ & f(g(X)) \end{aligned}$$

3.2 Fast multivariate multipoint evaluation in prime fields

This section gives an algorithm to perform multivariate multipoint evaluation in prime field, and a proof that this algorithm returns the desired result.

3.2.1 Algorithm

Algorithm 2
Input: $f(X_0, \cdots X_{m-1}) \in \mathbb{F}_p[X_0, \cdots X_{m-1}]$ (p prime) with degree at most $d-1$ in each variable
$\alpha_0, \cdots, \alpha_{N-1} \in \mathbb{F}_p^m$
Output: $f(\alpha_i)$ for $i = 0, \cdots, N-1$

---

1. Compute the reduction $\bar{f}$ of f modulo $X_j^p - X_j$ for $j = 0, \cdots, m-1$.

2. Compute $\bar{f}(\alpha)$ for all $\alpha \in \mathbb{F}_p^m$.

   (a) When $m = 1$, perform univariate multipoint evaluation.

   (b) When $m > 1$, write $\bar{f}(X_0, X_1, \cdots, X_{m-1})$ as $\sum_{i=0}^{p-1} X_0^i f_i(X_1, \cdots, X_{m-1})$. For each f_i, recursively compute its evaluations at all of $\mathbb{F}_p^{m-1}$.

   (c) For each $\beta \in \mathbb{F}_p^{m-1}$, evaluate the univariate polynomial $\sum_{i=0}^{p-1} X_0^i f_i(\beta)$ at all of $\mathbb{F}_p$.

3. Loop up and return $f(\alpha_i)$ for $i = 0, \cdots, N-1$

3.2.2 Proof

Step 1 adds the information that the polynomial will be evaluated in $\mathbb{F}_p$.
Step 2 evaluate $\bar{f}$ at all the possible points in the finite field, and step 3 look up the desired values. Therefore, when N is much more larger than p^m, this algorithm is more efficient than evaluating f at each α_i one by one.

3.3 Fast multivariate multipoint evaluation in Rings of the form $\mathbb{Z}/r\mathbb{Z}$

This section gives an algorithm to perform multivariate multipoint evaluation in the Rings of the form $\mathbb{Z}/r\mathbb{Z}$, and a proof that this algorithm returns the desired result.

3.3.1 Algorithm

Algorithm 3

Input: $f(X_0, \cdots X_{m-1}) \in (\mathbb{Z}/r\mathbb{Z})[X_0, \cdots X_{m-1}]$ with degree at most $d-1$ in each variable

$\alpha_0, \cdots, \alpha_{N-1} \in (\mathbb{Z}/r\mathbb{Z})^m$

t is the number of rounds.

Output: $f(\alpha_i)$ for $i = 0, \cdots, N-1$

---

1. Construct the polynomial $\tilde{f}(X_0, \cdots X_{m-1}) \in \mathbb{Z}[X_0, \cdots X_{m-1}]$ from f by replacing each coefficient with its lift in $\{0, \cdots, r-1\}$. For $i = 0, \cdots, N-1$, construct the m-tuple $\tilde{\alpha_i} \in \mathbb{Z}^m$ from α_i by replacing each coordinate with its lift in $\{0, \cdots, r-1\}$.
2. Compute the primes $p_1, \cdots, p_k$ less than or equal to $l = 16\log(d^m(r-1)^{md})$.
3. For $h = 1, \cdots, k$, compute the reduction $f_h \in \mathbb{F}_{p_h}[X_0, \cdots X_{m-1}]$ of $\tilde{f}$ modulo p_h. For $h = 1, \cdots, k$ and $i = 0, \cdots, N-1$, compute the reduction $\alpha_{h,i} \in \mathbb{F}_{p_h}^m$ of $\tilde{\alpha_i}$ modulo p_h.
4. If $t = 1$, then for $h = 1, \cdots, k$, apply **Algorithm 2** to compute $f_h(\alpha_{h,i})$ for $i = 1, \cdots, N-1$; otherwise run **Algorithm 3**$(f_h, \alpha_{h,0}, \cdots, \alpha_{h,N-1}, p_h, t-1)$ to compute $f_h(\alpha_{h,i})$ for $i = 1, \cdots, N-1$.
5. For $i = 1, \cdots, N-1$, use the Chinese Remainder Theorem to compute the unique integer in $\{0, \cdots, (p_1p_2\cdots p_k)-1\}$ congruent to $f_h(\alpha_{h,i})$ modulo p_h for $h = 1, \cdots, k$ and return its reduction modulo r.

3.3.2 Proof

First, step 1 adds the information that the polynomial will be evaluated in $\mathbb{Z}/r\mathbb{Z}$. Therefore, $\tilde{f}(\tilde{\alpha_i}) \leq d^m(r-1)^{md}$. And by **Lemma 1**, $d^m(r-1)^{md} < p_1p_2\cdots p_k$. Then, $\tilde{f}(\tilde{\alpha_i}) < p_1p_2\cdots p_k$. Then $\tilde{f}(\tilde{\alpha_i}) \bmod p_1p_2\cdots p_k = \tilde{f}(\tilde{\alpha_i})$.

Furthermore, this algorithm uses the fact from the Chinese Remainder Theorem that

$$\mathbb{Z}/(p_1p_2\cdots p_k) = \mathbb{Z}/(p_1) \times \mathbb{Z}/(p_2) \times \cdots \times \mathbb{Z}/(p_k)$$

Then, $\tilde{f}(\tilde{\alpha_i}) \bmod p_1p_2\cdots p_k \leftrightarrow (\tilde{f}(\tilde{\alpha_i}) \bmod p_1, \cdots, \tilde{f}(\tilde{\alpha_i}) \bmod p_k)$.

We know $f_h \equiv \tilde{f} \bmod p_h$, $\alpha_{h,i} \equiv \tilde{\alpha_i} \bmod p_h$. Then $f_h(\alpha_{h,i}) \equiv \tilde{f}(\tilde{\alpha_i}) \bmod p_h$.

Then $\tilde{f}(\tilde{\alpha_i}) \bmod p_1p_2\cdots p_k \leftrightarrow (f_h(\alpha_{1,i}), \cdots, f_h(\alpha_{k,i}))$.

Therefore, the unique integer that congruent to $f_h(\alpha_{h,i}) \bmod p_h$ for $h = 1, \cdots, k$ is equal to $\tilde{f}(\tilde{\alpha_i})$, which is equal to $f(\alpha_i)$ in $\mathbb{Z}/r\mathbb{Z}$.

The number of rounds does not affect the result of this algorithm, more rounds means we apply Chinese Remainder Theorem more times. For small r, one round is enough.

3.4 Fast multivariate multipoint evaluation in extension rings

This section gives an algorithm to perform multivariate multipoint evaluation in the extension ring $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))[X_0, \cdots, X_{m-1}]$, and and a proof that this algorithm returns the desired result.

3.4.1 Algorithm

Algorithm 4

Input: R is a finite ring given as $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$ for some monic polynomial $E(Z)$ of degree e
$f(X_0, \cdots X_{m-1}) \in R[X_0, \cdots X_{m-1}]$ with degree at most $d-1$ in each variable
$\alpha_0, \cdots, \alpha_{N-1} \in R^m$
t is the number of rounds.

Output: $f(\alpha_i)$ for $i = 0, \cdots, N-1$

---

0. Compute $M = d^m(e(r-1))^{(d-1)m+1}$ and $r' = M^{(e-1)dm+1}$.

1. Construct the polynomial $\tilde{f}(X_0, \cdots X_{m-1}) \in \mathbb{Z}[Z][X_0, \cdots X_{m-1}]$ from f by replacing each coefficient with its lift, which is a polynomial of degree at most $e-1$ with coefficients in $\{0, \cdots, r-1\}$. For $i = 0, \cdots, N-1$, construct the m-tuple $\tilde{\alpha_i} \in \mathbb{Z}[Z]^m$ from α_i by replacing each coordinate with its lift, in which is a polynomial of degree at most $e-1$ with coefficients $\{0, \cdots, r-1\}$.

2. Compute the reduction $\bar{f} \in (\mathbb{Z}/r'\mathbb{Z})[X_0, \cdots X_{m-1}]$ of $\tilde{f}$ modulo r' and $Z - M$. For $i = 0, \cdots, N-1$, compute the reduction $\bar{\alpha_i} \in (\mathbb{Z}/r'\mathbb{Z})^m$ of $\tilde{\alpha_i}$ modulo r' and $Z - M$.

3. Run **Algorithm 3**$(\bar{f}, \bar{\alpha_0}, \bar{\alpha_1}, \cdots, \bar{\alpha_{N-1}}, r', t)$ to compute $\beta_i = \bar{f}(\bar{\alpha_i})$ for $i = 0, \cdots, N-1$.

4. For $i = 1, \cdots, N-1$, compute the unique polynomial $Q_i[Z] \in \mathbb{Z}[Z]$ of degree at most $(e-1)dm$ with coefficients in $\{0, \cdots, M-1\}$ for which $Q_i(M)$ has remainder β_i modulo r', and return the reduction of Q_i modulo r and $E(Z)$.

3.4.2 Proof

Step 1 adds the information that the polynomial will be evaluated in $(\mathbb{Z}/r\mathbb{Z})[Z]/(E(Z))$. Therefore, $\tilde{f}(\tilde{\alpha_i})$ has degree at most $(e-1)dm$. What's more, when $Z = 1$, the value of each coefficient of $\tilde{f}$ and each coordinate of $\tilde{\alpha_i}$ is at most $e(r-1)$. Therefore, $\tilde{f}(\tilde{\alpha_i})(1) \leq d^m(e(r-1))^{(d-1)m+1} = M-1$, which means that each coefficient of $\tilde{f}(\tilde{\alpha_i})$ is less than M.
We know that $Q_i[Z]$ is also a polynomial of degree at most $(e-1)dm$ and coefficients in $\{0, \cdots, M-1\}$. And $Q_i[M] = \beta_i = \bar{f}(\bar{\alpha_i}) = \tilde{f}(\tilde{\alpha_i})(M) \bmod r'$. Then $f(\alpha_i) = (Q_i[Z] \bmod r') \bmod E(Z)$.

3.5 Helper Functions

This section gives the purpose of each helper function in the implementation of the algorithms.

3.5.1 Modular_Composition

$Select\_N\_Distinct\_Elements(N, r, E)$: select N distinct elements from the ring $(Z/rZ)[Z]/(E[Z])$, if there are not enough elements, return False.
$Kronecker\_SubStitution\_Inverse(f, h, l)$: return $\psi_{h,l}(f)$ with unknowns $\{Y_0, Y_1, \cdots, Y_l\}$.

3.5.2 MultiModular_Prime_Field

$Coeffs\_from\_zero\_to\_p\_minus\_one(f, p, x)$: return the coefficients of f from x^0 to x^{p-1}.
$Kth\_element\_in\_each\_list(L, k)$: L is a list of list, $L = [L0, L1, \cdots]$, returns $[L0[k], L1[k], \cdots]$.
$MultidimentionFFT(f, m, p, X)$: perform the step 2 of **Algorithm 2**, return a list of evaluations.

3.5.3 MultiModular_$\mathbb{Z}$_$r\mathbb{Z}$

$Primes\_Less\_than\_N(N)$ returns a list of primes less than or equal to N.
$compute\_d(f, X)$ returns d where f has degree at most $d - 1$ in each variable.

3.5.4 MultiModular_Extension_Ring

$alpha\_mod\_poly(\alpha, quo, Z, m)$ returns a new list α' from replacing each coordinate of each point in α with its value modulo quo. α is a list of evaluation points.

References

[1] R. P. Brent and H. T. Kung. *Fast algorithms for manipulating formal power series.* Journal of the Association for Computing Machinery, Vol. 25, No. 4, pp. 581–595, 1978.

[2] Kedlaya and Umans *Fast polynomial factorization and modular composition.* Journal on Computing, Vol. 40, No. 6, pp. 1767–1802, 2011.