CKKS Lecture Notes
Part 5: Polynomial arithmetic

FEI

October 2026

1 Residue number system

Let $Q_n = q_0 \dots q_n$ be the modulus chain. Let

$$f \in \mathbb{Z}_{Q_\ell}[X]/(X^N + 1) = \mathbb{Z}_{q_0 \dots q_\ell}[X]/(X^N + 1).$$

By the Chinese remainder theorem (CRT), the following holds:

Fact

$$\mathbb{Z}_{Q_\ell}[X]/(X^N + 1) \cong \mathbb{Z}_{q_0}[X]/(X^N + 1) \times \cdots \times \mathbb{Z}_{q_\ell}[X]/(X^N + 1).$$

Therefore, we can represent f in terms of its "limbs" $f_0 \in \mathbb{Z}_{q_0}[X]/(X^N + 1), \dots, f_\ell \in \mathbb{Z}_{q_\ell}[X]/(X^N + 1)$ such that

$$\begin{aligned} f_0 &\equiv f \pmod{q_0} \\ &\vdots \\ f_\ell &\equiv f \pmod{q_\ell}. \end{aligned}$$

Because this is an isomorphism, we can do operations separately on each limb. Let $g \in \mathbb{Z}_{Q_\ell}[X]/(X^N + 1)$ have limbs $g_0, \dots, g_\ell$. The limbs of $f + g$ are

$$f_0 + g_0, \dots, f_\ell + g_\ell.$$

The limbs of fg are

$$f_0 g_0, \dots, f_\ell g_\ell.$$

We can also rescale. Remember that rescaling f from modulus Q_ℓ to $Q_{\ell-1} = Q_\ell / q_\ell$ takes two steps:

- Round each coefficient of f to the nearest multiple of q_ℓ. Equivalently, compute $f - \mathsf{rep}(f_\ell)$.
- Divide by q_ℓ to get a result in $\mathbb{Z}_{Q_{\ell-1}}[X]/(X^N+1)$.

We can do these steps on the limbs:

- Compute $f' \in \mathbb{Z}_{Q_{\ell-1}}[X]/(X^N+1)$ with limbs

$$(f_0 - \mathsf{rep}(f_\ell), \ldots, f_{\ell-1} - \mathsf{rep}(f_\ell)).$$

- Multiply by the inverse of q_ℓ:

$$(q_\ell^{-1}(f_0 - \mathsf{rep}(f_\ell)), \ldots, q_\ell^{-1}(f_{\ell-1} - \mathsf{rep}(f_\ell))).$$

Finally, we can lift or lower the modulus. If we add a new prime p to the modulus, the limbs for $q_0, \ldots, q_\ell$ stay the same. The new limb for p is $\mathsf{rep}(f) \pmod p$. To remove a prime from the modulus, we can just delete its limb.

By composing these, we can do any CKKS operation.

2 Number theoretic transform

In this section, we will look at how to express individual limbs so that we can efficiently operate on them.

Exercise

Let $f \in \mathbb{Z}_q[X]/(X^N+1)$ for some prime q. For which $z \in \mathbb{Z}_q$ does it make sense to evaluate $f(z)$? For which q does such z exist?

Solution

Since $X^N + 1 \equiv 0$ in $\mathbb{Z}_q[X]/(X^N+1)$, it only makes sense to evaluate $f(z)$ if $z^N + 1 \equiv 0 \pmod q \iff z^N \equiv -1 \pmod q$.

Such z exists when $q \equiv 1 \pmod{2N}$. In fact, if such z satisfies $z^N \equiv -1 \pmod q$, then

$$z^1, z^3, \ldots, z^{2N-1}$$

is a complete list of roots of $X^N + 1 \equiv 0 \pmod q$.

Call a prime q *NTT-friendly* if $q \equiv 1 \pmod{2N}$. We choose the modulus chain $q_0 \ldots q_n$ such that each q_i is NTT-friendly. Since q_i is NTT-friendly, we can use the Chinese Remainder Theorem to split the ring even further.

Fact

Let $q \equiv 1 \pmod{2N}$ be prime. Then,

$$\mathbb{Z}_q[X]/(X^N + 1) \cong \mathbb{Z}_q^N.$$

Proof

Let $z \in \mathbb{Z}_q$ such that $z^N + 1 \equiv 0 \pmod{q}$. Since $z^1, z^3, \ldots, z^{2N-1}$ are the roots of $X^N + 1 \equiv 0 \pmod{q}$,

$$X^N + 1 \equiv (X - z^1)(X - z^3) \ldots (X - z^{2N-1}) \pmod{q}.$$

By the Chinese Remainder Theorem,

$$\mathbb{Z}_q[X]/(X^N + 1) \cong \mathbb{Z}_q[X]/(X - z^1) \times \cdots \times \mathbb{Z}_q[X]/(X - z^{2N-1}) \cong \mathbb{Z}_q^N.$$

This isomorphism is given by

$$\Phi(f) = (f(z^1), f(z^3), \ldots, f(z^{2N-1})).$$

Therefore, the limbs of a polynomial can be stored in two different forms:

- **Coefficient form.** This stores the coefficients of f.
- **NTT form.** This stores the evaluations of f at $z^1, z^3, \ldots, z^{2N-1}$.

In either form, f is stored as a list of N elements of $\mathbb{Z}_q$.

Exercise

According to the ring learning with errors assumption, given $a, b \in \mathbb{Z}_q[X]/(X^N + 1)$ such that $as + b \equiv e \pmod{q}$ for some unknown error e with small coefficients, it is hard to solve for s.

In evaluation form, we have

$$\begin{aligned} a(z^1)s(z^1) + b(z^1) &\equiv e(z^1) \pmod{q} \\ a(z^3)s(z^3) + b(z^3) &\equiv e(z^3) \pmod{q} \\ &\vdots \\ a(z^{2N-1})s(z^{2N-1}) + b(z^{2N-1}) &\equiv e(z^{2N-1}) \pmod{q}. \end{aligned}$$

Why can't we solve $as + b \equiv e \pmod{q}$ by solving each of the above equations individually?

Answer

The error e is only guaranteed to have small coefficients, so $e(z^k)$ is not necessarily small.

3 Operations in coefficient and NTT form

Addition and multiplication act independently on the limbs.

- We can add limbs in coefficient form by adding the coefficients. Similarly, we can add limbs in NTT form by adding the evaluations.
- We can only multiply efficiently in NTT form. We can multiply limbs in NTT form by multiplying the evaluations.

We can remove primes from the modulus of a polynomial in either form by removing limbs. We can only take rep of a polynomial in coefficient form, so rescaling and lifting require coefficient form.

4 Converting between forms

We can convert between the forms in $O(N\log(N))$ time.

Problem: Negacyclic NTT

Let $q \equiv 1 \pmod{2N}$ be prime. Let $z \in \mathbb{Z}_q$ such that $z^N + 1 \equiv 0 \pmod{q}$.

Given $f(X) = r_0 + r_1 X + \cdots + r_{N-1}X^{N-1} \in \mathbb{Z}_q[X]/(X^N+1)$, compute

$$(f(z^1), \ldots, f(z^{2N-1})).$$

This is equivalent to decomposing f into $f_1 \in \mathbb{Z}_q[X]/(X-z)$, $f_3 \in \mathbb{Z}_q[X]/(X-z^3), \ldots, f_{2N-1} \in \mathbb{Z}_q[X]/(X-z^{2N-1})$. We have:

$$\begin{aligned}
&\mathbb{Z}_q[X]/(X^N+1) \\
&\cong \mathbb{Z}_q[X]/(X^{N/2}-z^{N/2}) \times \mathbb{Z}_q[X]/(X^{N/2}+z^{N/2}) \\
&\cong \mathbb{Z}_q[X]/(X^{N/2}-z^{N/2}) \times \mathbb{Z}_q[X]/(X^{N/2}-z^{3N/2}) \\
&\cong \left(\mathbb{Z}_q[X]/(X^{N/4}-z^{N/4}) \times \mathbb{Z}_q[X]/(X^{N/4}-z^{5N/4})\right) \\
&\quad \times \left(\mathbb{Z}_q[X]/(X^{N/4}-z^{3N/4}) \times \mathbb{Z}_q[X]/(X^{N/4}-z^{7N/4})\right) \\
&\vdots \\
&\cong \mathbb{Z}_q[X]/(X-z^1) \times \cdots \times \mathbb{Z}_q[X]/(X-z^{2N-1}).
\end{aligned}$$

So, to compute the negacyclic NTT of f, it suffices to compute the following:

Subproblem

Let $q \equiv 1 \pmod{2N}$ be prime. Let $z \in \mathbb{Z}_q$ such that $z^N + 1 \equiv 0 \pmod{q}$. Let $k \leq \log_2(N)$ and let $m \in \mathbb{Z}$ be odd.

Let $f(X) = r_0 + r_1 X + \cdots + r_{2^k-1} X^{2^k-1} \in \mathbb{Z}_q[X]/(X^{2^k} - z^{2^k m})$. Then, f is mapped to some element of

$$\mathbb{Z}_q[X]/(X^{2^{k-1}} - z^{2^{k-1} m}) \times \mathbb{Z}_q[X]/(X^{2^{k-1}} + z^{2^{k-1} m})$$

under the isomorphism given by CRT. Determine this element.

Solution

We have

$$\begin{aligned} f &\equiv (r_0 + r_1 X + \cdots + r_{2^{k-1}-1} X^{2^{k-1}-1}) \\ &\quad + X^{2^{k-1}} (r_{2^{k-1}} + r_{2^{k-1}+1} X + \cdots + r_{2^k-1} X^{2^{k-1}-1}) \\ &\equiv (r_0 + r_1 X + \cdots + r_{2^{k-1}-1} X^{2^{k-1}-1}) \\ &\quad + z^{2^{k-1} m} (r_{2^{k-1}} + r_{2^{k-1}+1} X + \cdots + r_{2^k-1} X^{2^{k-1}-1}) \\ &\equiv (r_0 + z^{2^{k-1} m} r_{2^{k-1}}) + \cdots + (r_{2^{k-1}-1} + z^{2^{k-1} m} r_{2^k-1}) X^{2^{k-1}-1} \\ &\quad \pmod{X^{2^{k-1}} - z^{2^{k-1} m}}. \end{aligned}$$

Similarly,

$$\begin{aligned} f &\equiv (r_0 - z^{2^{k-1} m} r_{2^{k-1}}) + \cdots + (r_{2^{k-1}-1} - z^{2^{k-1} m} r_{2^k-1}) X^{2^{k-1}-1} \\ &\quad \pmod{X^{2^{k-1}} + z^{2^{k-1} m}}. \end{aligned}$$

We can solve the subproblem in $O(2^k)$ time. We can apply the isomorphism

$$\Phi : \mathbb{Z}_q[X]/(X^N + 1) \to \mathbb{Z}_q[X]/(X - z^1) \times \cdots \times \mathbb{Z}_q[X]/(X - z^{2N-1})$$

by repeatedly solving the subproblem:

- **Step 1.** Solve the subproblem for $k = \log_2(N), m = 1$. This produces an element
$$(f_0, f_1) \in \mathbb{Z}_q[X]/(X^{N/2} - z^{N/2}) \times \mathbb{Z}_q[X]/(X^{N/2} - z^{3N/2}).$$
- **Step 2.** Solve the subproblem for $k = \log_2(N) - 1, m = 1$ on input f_0. This produces an element
$$(f_{00}, f_{01}) \in \mathbb{Z}_q[X]/(X^{N/4} - z^{N/4}) \times \mathbb{Z}_q[X]/(X^{N/4} - z^{5N/4}).$$

Solve the subproblem for $k = \log_2(N) - 1, m = 3$ on input f_1. This produces an element

$$(f_{10}, f_{11}) \in \mathbb{Z}_q[X]/(X^{N/4} - z^{3N/4}) \times \mathbb{Z}_q[X]/(X^{N/4} - z^{7N/4}).$$

- **Steps 3-$\log_2(N)$.** Repeat this process.

Each step takes $O(N)$ time, so the whole algorithm takes $O(N \log N)$ time. Converting from NTT form to coefficient form works similarly. We can essentially reverse each step.

5 Montgomery arithmetic

Now, we will look at how to do arithmetic modulo q efficiently.

Basic method

Represent $x \in \mathbb{Z}_q$ as its representative in $[0, q)$. Do each operation (addition or multiplication) by doing it in $\mathbb{Z}$, then reducing the result modulo q.

Let $x, y \in [0, q)$. Then, $x + y \in [0, 2q)$. To reduce the result modulo q:

- If $x + y \geq q$, subtract q.
- If $x + y < q$, do nothing.

This is efficient.

However, $xy \in [0, q^2)$. To reduce xy modulo q, we compute

$$xy - q \left\lfloor \frac{xy}{q} \right\rfloor.$$

This is less efficient because we need to divide.

We can solve this problem by storing x and y in Montgomery form. Choose some fixed $M > q$ to be a power of 2. The Montgomery form $\mathsf{Mont}(x)$ is an integer in $[0, q)$ such that

$$\mathsf{Mont}(x) \equiv Mx \pmod{q}.$$

Addition in Montgomery form works the same way as the basic method. Multiplication in Montgomery form is more efficient than in the basic method.

Montgomery multiplication

We have

$$\mathsf{Mont}(x)\mathsf{Mont}(y) \equiv M^2 xy \pmod{q}.$$

We would like to remove the extra M factor and reduce modulo q. We do this with the following steps:

- Add a small multiple of q to $\mathsf{Mont}(x)\mathsf{Mont}(y)$ such that the result is divisible by M. (This does not change the evaluation modulo q.)
- Divide by M. (Because M is a power of 2, this is efficient).

Compute

$$r \equiv -q^{-1}\mathsf{Mont}(x)\mathsf{Mont}(y) \pmod{M}.$$

Again, the fact that M is a power of 2 makes this efficient. Compute

$$rq + \mathsf{Mont}(x)\mathsf{Mont}(y).$$

This is a multiple of M. So, we can divide by M to get a result w. We have

$$w = \frac{rq + \mathsf{Mont}(x)\mathsf{Mont}(y)}{M} \equiv Mxy \pmod{q}.$$

Also,

$$w = \frac{rq + \mathsf{Mont}(x)\mathsf{Mont}(y)}{M} < 2q.$$

So, we can get the Montgomery form of xy by reducing w modulo q, which takes at most one subtraction.