THE SCHÖNHAGE-STRASSEN ALGORITHM FOR MULTIPLYING POLYNOMIALS

FABIAN GUNDLACH

Lemma 1. *For $n \geqslant 1$, if f and g are polynomials of degree $< n$ with $g \neq 0$, and there are at most $1 \leqslant m \leqslant n$ monomials in $g(X)$, then we can compute $f \bmod g$ in time $\mathcal{O}(nm)$ on an $\mathcal{O}(\log n)$-bit RAM.*

Proof. Use schoolbook division. Each time we eliminate the leading coefficient, we only need to modify m coefficients. We eliminate the leading coefficient at most n times. ☐

We let ϕ_n be the n-th cyclotomic polynomial. Its degree is $\varphi(n) = \#(\mathbb{Z}/n\mathbb{Z})^\times$. Note that for any prime $r \geqslant 2$ and any $k \geqslant 1$, we have $\varphi(r^k) = (1 - 1/r)r^k$ and the cyclotomic polynomial $\phi_{r^k}(X) = \phi_r(X^{r^{k-1}}) = \sum_{i=0}^{r-1} X^{ir^{k-1}}$ has exactly r monomials.

We saw in class that Fourier transforms in $\prod_{i\in\mathbb{Z}/r^k\mathbb{Z}} R$ allow us to quickly compute products of elements of $R[X]/(X^{r^k} - 1)$. Between Fourier transforms, the algorithm involved multiplying two elements of $\prod_{i\in\mathbb{Z}/r^k\mathbb{Z}} R$, i.e. r^k multiplications in R. It turns out that if we want to multiply elements of the quotient $R[X]/\phi_{r^d}(X)$ of $R[X]/(X^{r^k} - 1)$, only the entries where i is relatively prime to r^k matter. (This makes sense because $\phi_{r^d}(X) = \prod_{i\in(\mathbb{Z}/r^k\mathbb{Z})^\times}(X - \zeta_{r^k}^i)$, so the ring $\mathbb{C}[X]/\phi_{r^d}(X)$ captures only the values of a polynomial at $\zeta_{r^k}^i$ with $i \in (\mathbb{Z}/r^k\mathbb{Z})^\times$.) We then only need to perform $\varphi(r^k)$ multiplications in R and this optimization will be crucial in the proof of Lemma 3 below.

Lemma 2. *Assume that R contains a root ζ_{r^k} of ϕ_{r^k}. For $k \geqslant 1$, given any (reduced) elements f and g of $R[X]/\phi_{r^k}(X)$, we can compute $r^k \cdot fg \in R[X]/\phi_{r^k}(X)$ in time $\mathcal{O}_r(r^k \cdot k)$ on an $\mathcal{O}(k)$-bit RAM. The arithmetic operations in R used are: $\mathcal{O}_r(r^k \cdot k)$ additions in R, $\mathcal{O}_r(r^k \cdot k)$ multiplications by powers of ζ_{r^k}, exactly $\varphi(r^k)$ further multiplications of two (arbitrary) elements of R.*

Proof. Write $f(X) = \sum_{i=0}^{\varphi(r^k)-1} a_i X^i$ and $g(X) = \sum_{i=0}^{\varphi(r^k)-1} b_i X^i$ and let $a_i = b_i = 0$ for $i = \varphi(r^k), \ldots, r^k - 1$. Let $a = (a_i)_{i\in\mathbb{Z}/r^k\mathbb{Z}}$ and $b = (b_i)_{i\in\mathbb{Z}/r^k\mathbb{Z}}$.

Use the Cooley-Tukey algorithm to compute the Fourier transforms $\widehat{a}$ and $\widehat{b}$ of a and b.

For $j \in \mathbb{Z}/r^k\mathbb{Z}$, compute $\widehat{c}_j = \begin{cases} \widehat{a}_j \cdot \widehat{b}_j, & j \in (\mathbb{Z}/r^k\mathbb{Z})^\times, \\ 0, & \text{otherwise.} \end{cases}$

Then, use the Cooley-Tukey algorithm to compute the Fourier transform c of $\widehat{c}$.

We leave it as an exercise to show that $\sum_{i=0}^{r^k-1} c_i X^i \bmod \phi_{r^k}(X) = r^k \cdot f(X)g(X)$. ☐

Lemma 3. *Let $r = 2$ or 3. For $k \geqslant 1$, given any (reduced) elements f and g of $R[X]/\phi_{r^k}(X)$, we can compute $r^w \cdot fg$ in time $\mathcal{O}(r^k \cdot k \cdot \log(k+1))$ on an $\mathcal{O}(k)$-bit RAM, for some $w = w_{r,k} = \mathcal{O}(k)$.*

Proof. For small k, use schoolbook multiplication and reduce the result modulo ϕ_{r^k}.

For large k, proceed recursively as follows: Assume we can multiply for $k' < k$ in time $T(k')$.

Let

$$v = \begin{cases} k+2 & \text{if } r = 2, \\ k+1 & \text{if } r = 3, \end{cases}$$

and let $m = \lfloor v/2 \rfloor$ and $l = v - m = \lceil v/2 \rceil$. If k' is sufficiently large, then $k/2 \leqslant m \leqslant l < k$.

We will consider the following rings:

$$S = R[Y]/\phi_{r^l}(Y),$$
$$U = S[Z]/\phi_{r^m}(Z) = R[Y,Z]/(\phi_{r^l}(Y), \phi_{r^m}(Z)).$$

Note that S contains a root $[Y]$ of ϕ_{r^l}. Since $m \leqslant l$, it also contains a root $[Y^{r^{l-m}}]$ of ζ_{r^m}, so we can use Lemma 2 to multiply elements F, G of U (or rather, compute some power of r times FG). Any such multiplication involves $\mathcal{O}(r^m \cdot m)$ additions/multiplications by powers of $[Y^{r^{l-m}}]$ in S. Each of them takes time $\mathcal{O}(\varphi(r^l)) \leqslant \mathcal{O}(r^l)$, so the total is $\mathcal{O}(r^{m+l} \cdot m) \leqslant \mathcal{O}(r^k \cdot k)$. Moreover, we need to do exactly $\varphi(r^m)$ multiplications of two elements of S. Since $l < k$, we can recursively apply the multiplication algorithm described in this proof. Each such multiplication in S takes time $T(l)$, so the total is $\varphi(r^m) \cdot T(l)$. All in all, we can multiply two elements of U in time $\mathcal{O}(r^k \cdot k) + \varphi(r^m) \cdot T(l)$.

Now, we describe how to reduce the multiplication in $R[X]/\phi_{r^k}(X)$ to a single multiplication in U:

Note that $\varphi(r^l)$ is divisible by 2 and that $\varphi(r^m)\varphi(r^l) = (1-1/r)^2 r^{m+l} = (1-1/r)^2 r^s = 2\varphi(r^k)$.

We can therefore (in time $\mathcal{O}(r^k)$) write

$$f(X) = \sum_{i=0}^{\varphi(r^m)-1} p_i(X) \cdot X^{i \cdot \varphi(r^l)/2}$$

with polynomials $p_i(X)$ of degree $< \varphi(r^l)/2$ and write

$$g(X) = \sum_{i=0}^{\varphi(r^m)-1} q_i(X) \cdot X^{i \cdot \varphi(r^l)/2}$$

with polynomials $q_i(X)$ of degree $< \varphi(r^l)/2$.

Consider the elements

$$F(Z) = \sum_{i=0}^{\varphi(r^m)-1} [p_i(Y)] Z^i$$

and

$$G(Z) = \sum_{i=0}^{\varphi(r^m)-1} [q_i(Y)] Z^i$$

of the ring U. As described above, compute $r^w \cdot FG$ for some integer $w \geqslant 0$.

Write

$$r^w \cdot F(Z) \cdot G(Z) = \sum_{i=0}^{\varphi(r^m)-1} [e_i(Y)] Z^i$$

with polynomials $e_i(Y)$ of degree $< \varphi(r^l)$.

In the variable Y, both sides have degree $< \varphi(r^l)$, so we in fact have an equality

$$r^w \cdot \left(\sum_{i=0}^{\varphi(r^m)-1} p_i(Y) Z^i \right) \cdot \left(\sum_{i=0}^{\varphi(r^m)-1} q_i(Y) Z^i \right) = \sum_{i=0}^{\varphi(r^m)-1} e_i(Y) Z^i$$

in the ring $R[Z]/\phi_{r^m}(Z)$, not just in U. Next, note that $\phi_{r^k}(X) = \phi_{r^m}(X^{r^{k-m}}) = \phi_{r^m}(X^{\varphi(r^l)/2})$. Hence, plugging in $Y = X$ and $Z = X^{\varphi(r^l)/2}$, we obtain

$$r^w f(X) g(X) = r^w \cdot \left(\sum_{i=0}^{\varphi(r^m)-1} p_i(X) X^{i\varphi(r^l)/2} \right) \cdot \left(\sum_{i=0}^{\varphi(r^m)-1} q_i(X) X^{i\varphi(r^l)/2} \right) = \sum_{i=0}^{\varphi(r^m)-1} e_i(X) X^{i\varphi(r^l)/2}.$$

Since each polynomial $e_i(X)$ has degree $< \varphi(r^l)$, this addition can be performed in time $\mathcal{O}(\varphi(r^m)\varphi(r^l)) = \mathcal{O}(r^k)$.

To summarize: All steps in the above algorithm take time $\mathcal{O}(r^k)$, except the ones in the application of Lemma 2, which take time $\mathcal{O}(r^k \cdot k) + \varphi(r^m) \cdot T(l)$.

The entire algorithm therefore has running time

$$T(k) \leqslant \mathcal{O}(r^k \cdot k) + \varphi(r^m) \cdot T(l).$$

We apply induction to show that $T(k) \leqslant C \cdot r^k \cdot (k-3) \cdot \log_2(k-3)$ for sufficiently large k (and some constant C). Note that $l - 3 \leqslant \frac{v+1}{2} - 3 \leqslant \frac{k+3}{2} - 3 = \frac{1}{2}(k-3)$. Hence,

$$\begin{aligned} T(k) &\leqslant \mathcal{O}(r^k \cdot k) + C \cdot \varphi(r^m) \cdot r^l \cdot (l-3) \cdot \log_2(l-3) \\ &\leqslant \mathcal{O}(r^k \cdot k) + C \cdot \varphi(r^m) \cdot r^l \cdot \frac{1}{2}(k-3) \cdot \log_2 \frac{1}{2}(k-3) \\ &= \mathcal{O}(r^k \cdot k) + C \cdot 2r^k \cdot \frac{1}{2}(k-3) \cdot \log_2 \frac{1}{2}(k-3) \\ &= \mathcal{O}(r^k \cdot k) + C \cdot r^k \cdot (k-3) \cdot (\log_2(k-3) - 1). \end{aligned}$$

For sufficiently large k, we have $\mathcal{O}(r^k \cdot k) \leqslant \mathcal{O}(r^k \cdot (k-3))$. As long as C is larger than the constant on the right-hand side, it follows that indeed

$$T(k) \leqslant C \cdot r^k \cdot (k-3) \cdot \log_2(k-3). \qquad \square$$

It's not difficult to show by induction that the exponent w of r satisfies $w = \mathcal{O}(k)$.

Corollary 1. *For large n, we can compute the product of two polynomials $f, g \in R[X]$ of degree $< n$ in time $\mathcal{O}(n \log n \log \log n)$.*

Proof. Choose the smallest numbers $k_2, k_3 \geqslant 0$ such that $\varphi(2^{k_2}) > 2n$ and $\varphi(3^{k_3}) > 2n$. Then, $2^{k_2} = \mathcal{O}(n)$ and $3^{k_3} = \mathcal{O}(n)$. As shown above, we can compute $2^{w_2} \cdot f(X)g(X) \mod \phi_{2^{k_2}}(X)$ and $3^{w_3} \cdot f(X)g(X) \mod \phi_{3^{k_3}}(X)$ in time $\mathcal{O}(r^{k_r} \cdot k_r \cdot \log k_r) = \mathcal{O}(n \log n \log \log n)$. Since they already reduced modulo the cyclotomic polynomials, we can in fact compute the polynomials $2^{w_2} \cdot f(X)g(X)$ and $3^{w_3} \cdot f(X)g(X)$. Writing 1 as a linear combination of 2^{w_2} and 3^{w_3}, we can compute the product $f(X)g(X)$. $\square$

See sections 3 and 4 of [Ber08] for an overview of the history behind this algorithm.

References

[Ber08] Daniel J. Bernstein. "Fast multiplication and its applications". In: *Algorithmic number theory: lattices, number fields, curves and cryptography.* Vol. 44. Math. Sci. Res. Inst. Publ. Cambridge Univ. Press, Cambridge, 2008, pp. 325–384.