A COMPUTATIONAL ALGORITHM FOR POLYNOMIAL MULTIPLICATION

Nguyễn Đình Thúc

ABSTRACT

In field K, given two polynomials: $a(x) = \sum_{0\le i<n} a_i x^i$; $b(x) = \sum_{0\le i<n} b_i x^i$ and $m(x) = x^n - 1$. This paper presents a computational algorithm for polynomial multiplication:

$u(x) \equiv a(x)b(x) \bmod m(x) = \sum_{0\le i<n} u_i x^i$.

The coefficients $(u_i)_{0\le i<n}$ are determined based on convolution and using the Chinese remainder theorem.

I. INTRODUCTION

We have seen that the polynomial multiplication is equivalent to acyclic convolution. Therefore, the product of two polynomials can be effected via a convolution. We have also seen that the cyclic convolution (X) of two signals x and y that have the same length n satisfies the Convolution theorem [CP01]:

(1.1) x X y = DFT-1 (DFT(x)*DFT(y))

where DFT is the discrete Fourier transform and DFT-1 is the inverse transform. As the known complexity of DFT is O(nln(n)) [CP01], [Bai90], [Cra96].

In this paper, we present a computational algorithm for polynomial multiplication based on convolution and using the Chinese remainder theorem [DPS96]. The rest of this paper is structured as follows: Section II describes the extended Euclidian algorithm of polynomials. Section III describes the Chinese remainder theorem of polynomials and Section IV presents the computational algorithm for multiplication.

II. THE EXTENDED EUCLIDIAN ALGORITHM OF POLYNOMIALS

Definition 2.1. A field K is a communicative ring, in which all of non-zero element has a multiplicative inverse.

In the field K, the polynomial f of x is an expression:

(2.1) $f(x) = \sum_{0\le i<n} a_i x^i$, $n \in N$; $a_i \in K$, $\forall\, 0 \le i \le n$; $a_n \neq 0$;

n is called degree of polynomial f, symbol $\delta(f)$. an is called leading coefficient of f, $\alpha(f)$.

Set of all of polynomials f(x) in the field K, symbol K[x], is a communicative ring, and is called polynomial ring. Let $K^*[x] = \{f \in K[x] ; \delta(f) \ge 0\}$.

$F \in K[x]$, $\delta(f) \ge 1$, is called irreducible, if f is not product of two polynomials that have $\delta(f) \ge 1$.

Given $f \in K^*[x]$, $a \in K$ is called root of f if f(a) = 0.

We have:

(2.2) $\delta(fg) = \delta(f) + \delta(g)$; $\delta(f+g) \le \max((\delta(f), \delta(g))$, $\forall f,g \in K[x]$; $f \neq 0$, $g \neq 0$.

Theorem 2.2. Given $f, g \in K[x]$, $g \neq 0$, we have two unique polynomials:

(2.3) $q, r \in K[x]$: $f = gq + r$, $\delta(r) < \delta(g)$;

q is called quotient and r is called remainder of the division of f by g; and we symbol:

(2.4) q = f div g ; r = f mod g.

Definition 2.3. $g \in K^*[x]$, $f_1 \bmod g = f_2 \bmod g$ is equivalent relation, and correlative space is K[x]/g.

Definition 2.4. Given $f, g \in K^*[x]$. The greatest common divisor of – gcd(f,g), is $d \in K^*[x]$ that has the greatest degree so that: $d|f, d|g, \alpha(d) = 1$.

If $f, g \in K^*[x]$, gcd(f,g) = 1, then f and g is called coprime.

The following algorithm determines $s, t \in K[x]$ and $d \in K^*[x]$ so that: $sf + tg = d$.

Algorithm 2.5.

(1) Initiation

a. $s_2 \leftarrow 1$;

b. $s_1 \leftarrow 0$;

c. $t_2 \leftarrow 0$;

d. $t_1 \leftarrow 1$;

(2) While $(g \neq 0)$

a. $q \leftarrow f \text{ div } g$; $r \leftarrow f - gq$;

b. $s \leftarrow s_2 - qs_1$; $t \leftarrow t_2 - qt_1$;

c. $f \leftarrow g$; $g \leftarrow r$;

d. $s_2 \leftarrow s_1$; $s_1 \leftarrow s$; $t_2 \leftarrow t_1$; $t_1 \leftarrow t$;

(3) If $(g = 0)$

a. $d \leftarrow f$;

b. $s \leftarrow s_2$;

c. $t \leftarrow t_2$;

III. THE CHINESE REMAINDER THEOREM OF POLYNOMIALS

Definition 3.1. $f \in K^*[x]$; $g, h \in K[x]$. Congruence mod f is defined as the following:

(3.1) $g \equiv h \bmod f \Leftrightarrow g \bmod f = h \bmod f \Leftrightarrow (g - h) \bmod f = 0 \Leftrightarrow f|(g - h)$.

Theorem 3.2. $f \in K^*[x]$, Congruence mod f satisfies:

(a) $g \equiv g \bmod f$;

(b) $g \equiv h \bmod f \Rightarrow h \equiv g \bmod f$;

(c) $g \equiv \bmod f$, $h \equiv \bmod f \Rightarrow g \equiv I \bmod f$;

(d) $g \equiv g1 \bmod f$, $h \equiv h1 \bmod f \Rightarrow (g+g1) \equiv (h+h1) \bmod f$, $(g*g1) \equiv (h*h1) \bmod f$.

Theorem 3.3 (The Chinese remainder theorem [DPS96])

(3.2) $k \geq 2$, $b1,\ldots, bk \in K[x]$; $n1,\ldots, nk \in K^*[x]$: $\gcd(n1,nj) = 1$, $\forall\, 1 \leq i \neq j \leq k$.

For each i, $1 \leq i \leq k$, let:

(3.3) $c_i = \prod_{1 \leq i \neq j \leq k} n_j = n \text{ div } n_i$;

$n = \prod_{1 \leq j \leq k} nj \in K[x]$.

Then the system of congruence:

(3.4) $u \equiv bi \bmod ni$; $1 \leq i \leq k$;

has unique solution [mod n] $u \in K[x]$.

IV. THE CONVOLUTION AND THE CHINESE REMAINDER THEORY

Definition 4.1. Given two arrays a, b in a field K:

(4.1) $a = \{a_i; 0 \leq i < n\}$, $b = \{b_i; 0 \leq i < n\}$; $a_i, b_i \in K$;

The cyclic convolution is an array u in K:

(4.2) $u = \{u_i; 0 \leq i < n\}$;

$u_i = \sum_{0 \leq i < n} u_i x^i$, $0 \leq i < n$.

Let

(4.3) $a(x) = \sum_{0 \leq i < n} a_i x^i$,

$b(x) = \sum_{0 \leq i < n} b_i x^i$, $u(x) = \sum_{0 \leq i < n} u_i x^i$, $m(x) = x^n - 1$;

and (4.2) gives:

(4.4) $u \equiv a(x)b(x) \bmod m(x)$

Our purpose is to determine the cyclic convolution $(u_i)_{0 \leq i < n}$. We will use the Chinese remainder theory to solve this problem.

Suppose that, in a field K, we have polynomials $(m_j)_{0 \leq j < k}$ so that m_i and m_j are coprime, $\forall\, i \neq j$, and:

(4.5) $m(x) = x^n - 1 = \prod_{1 \le i \ne j \le k} m_j(x)$.

Let

(4.6) $c_j = n \text{ div } m_j$,

$B_j = ab \text{ div } m_j$, $0 \le j < k$.

For each j, $0 \le j < k$, the algorithm 2.5 gives:

(4.7) $d_j \in K[x]: d_j c_j \equiv 1 \bmod m_j$,

$0 \le j < k$.

We set:

(4.8) $u = \sum_{0 \le j < k} B_j c_j d_j \bmod m$;

And we have:

(4.9) $u \equiv B_j \bmod m_j$, $0 \le j < k$.

We have also:

(4.10) $(ab \bmod m) \equiv B_j \bmod m_j$,

$0 \le j < k$.

Due to unique of solution of the Chinese remainder theory, we have: $u \equiv ab \bmod m$.

V. CONCLUSION

We have seen that the polynomial multiplication is equivalent to acyclic convolution. Therefore, the product of two polynomials can be effected via a convolution. In this paper, we present a computational algorithm using the Chinese remainder theory to determine the cyclic convolution. This method can be used to develop fast algorithms for large-integer arithmetic that are important in cryptography.

REFERENCES

[1] [Bai90] D. Bailey. FFTs in external or hierarchical memory. J. Supercomp., 4:23-35, 1990.

[2] [Cra96] R. Crandall. Topics in Advanced Scientific Computation. TELOS/ Springer-Verlag, 1996.

[3] [CP01] R. Crandall and C. Pomerance. Primer Numbers – A Computational Perspective. Springer-Verlag, 2001.

[4] [DPS96] C. Ding , D. Pei and A. Salomaa. Chinese Remainder Theorem. World Scientific, 1996.