AN ALGORITHM FOR POLYNOMIAL MULTIPLICATION WHICH DOES NOT DEPEND ON THE RING CONSTANTS

by

Michael Kaminski

Technical Report #418

June 1986
An algorithm for polynomial multiplication which
does not depend on the ring constants*

Michael Kamiński*

Faculty of Computer Science
Technion - Israel Institute of Technology
Haifa 32000
Israel

ABSTRACT

We present an algorithm for computing the coefficients of the product of two polynomials of degree $n$ over the ring of integers in $O(n \log n)$ multiplications. This algorithm relies on the Chinese Remainder Theorem, with cyclotomic polynomial presented as the moduli. The algorithm can be implemented over any ring and its implementation does not depend on the ring constants.

*The first version of this paper was written while the author was visiting Department of Computer Science of the University of Waterloo and presented at 25th annual IEEE symposium on Foundations of Computer Science.

**Supported in part by NSERC Grant No. A0254.
Introduction. Let \( R \) be a ring, and let \( f(\alpha) = \sum_{i=0}^{n-1} x_i \alpha^i \), \( g(\alpha) = \sum_{i=0}^{n-1} y_i \alpha^i \in R[\alpha] \) be polynomials over \( R \). Let \( h(\alpha) = f(\alpha)g(\alpha) = \sum_{i=0}^{2n-2} z_i \alpha^i \) be the product of \( f(\alpha) \) and \( g(\alpha) \):

\[
z_i = \sum_{m+k=i} x_m y_k, \quad i = 0, \ldots, 2n-2.
\]

We denote the column vectors of coefficients of \( f \), \( g \) and \( h \) by \( x \), \( y \) and \( z \), respectively: \( x = (x_0, \ldots, x_{n-1})^T \), \( y = (y_0, \ldots, y_{n-1})^T \), \( z = (z_0, \ldots, z_{2n-1})^T \).

It is known from [13] that if a set of bilinear forms of \((x, y)\) can be computed in \( m \) multiplications/divisions, then there exists a bilinear algorithm for computing the same set in \( m \) multiplications, i.e., each multiplication performed by this algorithm is of the form \( l \cdot l' \), where \( l(l') \) is a linear form of \( x(y) \). (This algorithm increases the number of additions and scalar multiplications at most by a factor of 9.) Therefore in what follows we restrict ourselves to bilinear algorithms.

If \( R = F \), where \( F \) is an infinite field, then any computation of \( \{z_i\}_{i=0}^{2n-2} \) (over \( R = F \) with \( \{x_i, y_i\}_{i=0}^{n-1} \) regarded as indeterminates) requires at least \( 2n-1 \) multiplications (cf. [5]).

In [14] Winograd proved that any algorithm for computing \( \{z_i\}_{i=0}^{2n-2} \) by means of \( 2n-1 \) multiplications has one of the following two forms:

a) Let \( t_0, \ldots, t_{2n-2} \) be distinct elements of \( F \), and \( W(t_0, \ldots, t_{2n-2}) = (w_{ij}) \) be \((2n-1) \times (2n-1)\) van der Monde matrix, i.e., \( w_{i+j+1} = t_i' \), \( i, j = 0, 1, \ldots, 2n-2 \). Let \( x' \) and \( y' \) be \( 2n-1 \) dimensional vectors defined by \( x' = (x_0, \ldots, x_{n-1}, 0, \ldots, 0)^T \), \( y' = (y_0, \ldots, y_{n-1}, 0, \ldots, 0)^T \). Compute \( Wx' = (l_0(x), \ldots, l_{2n-2}(x))^T \), \( Wy' = (l_0(y), \ldots, l_{2n-2}(y))^T \). \( l_0, \ldots, l_{2n-1} \) are linear forms, and no nonscalar multiplication is required.

Compute \( L(x, y) = (l_0(x)l_0(y), \ldots, l_{2n-2}(x)l_{2n-2}(y))^T \). This computation can be performed in \( 2n-1 \) nonscalar multiplications.

Finally, compute \( z = W^{-1}L(x, y) \). This computation does not require nonscalar multiplications.

b) Let \( t_1, \ldots, t_{2n-2} \) be distinct elements of \( F \), and \( W'(t_1, \ldots, t_{2n-2}) = (w'_{ij}) \) be \((2n-2) \times (2n-2)\) van der Monde matrix defined by \( w'_{i+j} = t_i'^{-1}, i, j = 1, \ldots, 2n-2 \). Let \( x'' \) and \( y'' \) be \( 2n-2 \) dimensional vectors defined by \( x'' = (x_0, \ldots, x_{n-1}, 0, \ldots, 0)^T \), \( y'' = (y_0, \ldots, y_{n-1}, 0, \ldots, 0)^T \). Compute \( W'x'' = (l'_1(x), \ldots, l'_{2n-2}(x)) \), \( W'y'' = (l'_1(y), \ldots, l'_{2n-2}(y)) \).
An algorithm for polynomial multiplication over \( R \) - \( A^R \) consists of a sequence of instructions
\(~I_1, \ldots, I_l\), where each instruction is of the forms:

1. \( I_m = \alpha_m \leftarrow \alpha_i \pm \alpha_j \quad i,j < m, \quad \alpha_i \text{ and } \alpha_j \text{ are variables.} \)
2. \( I_m = \alpha_m \leftarrow \alpha_i \ast \alpha_j \pm R \ast R \quad i,j < m, \quad \alpha_i \text{ and } \alpha_j \text{ are variables.} \)
3. \( I_m = \alpha_m \leftarrow \mathcal{C}_m \ast \alpha_i \quad i < m, \quad \mathcal{C}_m \in Z, \quad \alpha_i \text{ is a variable.} \)

(Recall that we are restricted to bilinear algorithms.)

An algorithm for polynomial multiplication over \( R \) - \( A^R \) consists of a sequence of instructions
\(~I_1^R, \ldots, I_l^R\), where \( I_m^R \) is obtained from \( I_m^R(m = 1, \ldots, l) \) as follows:

If \( I_m \) is of the form (i) then \( I_m^R = \alpha_m \leftarrow \alpha_i \pm R \alpha_j \), where \( \pm R \) denotes the addition (subtraction) in \( R \).

If \( I_m \) is of the form (ii), then \( I_m^R = \alpha_m \leftarrow \alpha_i \ast R \alpha_j \), where \( \ast R \) denotes the multiplication in \( R \).

If \( I_m \) is of the form (iii), then \( I_m^R = \alpha_m \leftarrow \mathcal{C}_m \ast \alpha_i \), where \( \mathcal{C}_m \) is a variable.
multiplicative-identity, we treat \([c_m]_R \cdot R \alpha_i\) as the sum of \(c_m\) copies of \(\alpha_i\), i.e., we introduce \(c_m - 1\) additions for computing \(\alpha_m\) (instead of one multiplication by a constant \(c_m\)).

3) \(R\) is not an integral domain or \(R\) is noncommutative ring. In this case a polynomial of degree \(m\) cannot be determined by its values at \(m\) points.

The algorithms a) and b) are "noncommutative", i.e., they do not require \(x_i\) and \(y_i\) to commute. However, the correctness of these algorithms is based on the assumption that the indeterminates commute with the field constants. But there exist extensions of \(R\) whose elements, generally speaking, do not commute with elements of \(R\). Consider, for example, \(\text{END}_R(\mathbb{R}^+)\) - the ring of endomorphisms of the additive group of \(R\) treated as a module over \(R\). One can embed \(R\) into \(\text{END}_R(\mathbb{R}^+)\) identifying \(s \in R\) with (say) \(l_s\) - the left multiplication by \(s\). It is known, that the elements of \(\text{END}_R(\mathbb{R}^+)\) do not commute with the images of the elements of \(\mathbb{R}\), even if \(R\) is a field.

The main result of this paper is an algorithm for multiplying polynomials of degree \(n\) over the ring of integers in \(O(n \log n)\) multiplications. (Thus the total number of operations is \(O(n^{1+\varepsilon})\) for any \(\varepsilon > 0\).) This algorithm is based on the Chinese Remainder Theorem, with the cyclotomic polynomials presented as the moduli. It can be implemented over any ring, and its implementation does not depend on the ring constants, i.e., it is non-commutative in the "strong" sense, cf. Theorem 3 below. The property of an algorithm to be independent on the ring constants can be used in a parallel computation by a vector machine over different domains.

The best algorithm known from the literature for multiplying polynomials over the ring of integers is the "divide and conquer algorithm" of Karatsuba and Ofman ([8]). It requires \(O(n^{1+\varepsilon})\) multiplications, and is based on the recursive application of the algorithm which computes the product of the first degree polynomials in three multiplications (algorithm b).

An application of the Chinese Remainder Theorem to polynomial multiplication. In this section we present general ideas lying behind our algorithm (cf. [5]).

Let \(v(\alpha), u(\alpha) \in \mathbb{Z}[\alpha]\). If \(u(\alpha)\) is a monic polynomial, then \(\text{res}(v(\alpha), u(\alpha))\) denotes the residue of \(v(\alpha)\) modulo \(u(\alpha)\), i.e., \(v(\alpha) = u(\alpha)r(\alpha) + \text{res}(v(\alpha), u(\alpha))\), for some \(r(\alpha) \in \mathbb{Z}[\alpha]\), and
\[ \text{deg } \text{res}(\alpha, u(\alpha)) < \text{deg } u(\alpha). \] Note that the coefficients of \( \text{res}(v(\alpha), u(\alpha)) \) are linear functions of the coefficients of \( v(\alpha) \) and can be computed over \( \mathbb{Z} \) given the coefficients of \( u(\alpha) \). Our algorithm is based on the Chinese Remainder Theorem stated below.

**The Chinese Remainder Theorem.** Let \( u_1(\alpha), \ldots, u_m(\alpha) \in \mathbb{Z}[\alpha] \) be monic polynomials generating pairwise comaximal ideals, i.e., for each \( i, j \leq m \), there exist polynomials \( p_{ij}(\alpha) \) and \( q_{ij}(\alpha) \) such that

\[
p_{ij}(\alpha) u_i(\alpha) + q_{ij}(\alpha) u_j(\alpha) = 1.
\]

Then for every \( v_1(\alpha), \ldots, v_m(\alpha) \in \mathbb{Z}[\alpha] \) such that \( \text{deg } v_i(\alpha) < \text{deg } u_i(\alpha), i = 1, \ldots, m \) there exists a unique polynomial \( v(\alpha) \in \mathbb{Z}[\alpha] \) of degree less than \( \sum_{i=1}^m \text{deg } u_i(\alpha) \) such that \( \text{res}(v(\alpha), u_i(\alpha)) = \text{res}(v(\alpha), u_j(\alpha)) \), \( i = 1, \ldots, m \). The coefficients of \( v(\alpha) \) are linear functions of the coefficients of \( v_1(\alpha), \ldots, v_m(\alpha) \) over \( \mathbb{Z} \), and can be computed from the coefficients of \( v_1(\alpha), \ldots, v_m(\alpha) \) over \( \mathbb{Z} \). (We suppose that \( u_1(\alpha), \ldots, u_m(\alpha) \) are given fixed polynomials.)

See [9 pp. 63-64] for the proof.

Let \( M(n) \) denote the number of multiplications needed to multiply two polynomials of degree \( n \) over the ring of integers. Let \( u_1(\alpha), \ldots, u_m(\alpha) \in \mathbb{Z}[\alpha] \) be fixed monic polynomials generating pairwise comaximal ideals, and, satisfying \( \sum_{i=1}^m \text{deg } u_i(\alpha) > 2n \). If \( f(\alpha), g(\alpha) \in \mathbb{Z}[\alpha] \) are polynomials of degree \( n \) with indeterminate coefficients then \( f(\alpha)g(\alpha) \) can be computed by means of the following algorithm:

Compute \( f_i(\alpha) = \text{res}(f(\alpha), u_i(\alpha)), g_i(\alpha) = \text{res}(g(\alpha), u_i(\alpha)) \), for \( i = 1, \ldots, m \). This computation does not require nonscalar multiplications and integer divisions.

Compute \( h_1(\alpha) = f_1(\alpha) g_1(\alpha) \), for \( i = 1, \ldots, m \). The number of multiplications required is \( \sum_{i=1}^m M(\text{deg } u_i(\alpha) - 1) \).

Compute (using the Chinese Remainder Theorem) \( h(\alpha) \) of degree less than \( \sum_{i=0}^m \text{deg } u_i(\alpha) \) such that \( \text{res}(h(\alpha), u_i(\alpha)) = \text{res}(h_i(\alpha), u_i(\alpha)) \), \( i = 1, \ldots, m \). This computation does not require nonscalar multiplications.

By the Chinese Remainder Theorem \( f(\alpha)g(\alpha) = h(\alpha) \). It follows that
then there exist monic polynomials \( W_j(\alpha) \) generating pairwise comaximal ideals. 

Obviously, \( deg W_j(\alpha) \leq k \). Then

\[
\frac{kd_k}{2} > 2n
\]

implies that \( \sum_{i=1}^{d_k} \deg u_i(\alpha) \geq \frac{kd_k}{2} > 2n \), and therefore it follows from (2) that

\[
M(n) \leq \sum_{i=1}^{d_k} M(\deg u_i(\alpha) - 1) \leq d_k M(k).
\]

Lemma 1. Let \( u_1(\alpha), \ldots, u_m(\alpha) \) be monic polynomials of degree not exceeding \( k \) generating pairwise comaximal ideals. Then there exist monic polynomials \( w_1(\alpha), \ldots, w_m(\alpha) \) generating pairwise comaximal ideals, such that \( k/2 < \deg w_i(\alpha) \leq k \), \( i = 1, \ldots, m \).

Proof. Define \( w_i(\alpha) = u_i(\alpha)^{[k/\deg u_i(\alpha)]} \), \( i = 1, \ldots, m \), where \( [z] \) denotes the greatest integer not exceeding \( z \). These polynomials generate comaximal ideals (cf. [9] p. 64), and \( \deg w_i(\alpha) = [k/\deg u_i(\alpha)] \deg u_i(\alpha), i = 1, \ldots, m \). Hence \( \deg w_i(\alpha) = \deg u_i(\alpha) \) if \( \deg u_i(\alpha) > \frac{k}{2} \), and \( \deg w_i(\alpha) > k - \deg u_i(\alpha) \) if \( \deg u_i(\alpha) \leq \frac{k}{2} \). In both cases \( \deg w_i(\alpha) > \frac{k}{2} \).

Obviously, \( \deg w_i(\alpha) \leq k \). \( \square \)
The dependence of $M(n)$ on $d_1, d_2, \ldots, d_k, \ldots$ is established in Theorem 1.

**Theorem 1.** If for any $k d_k \geq k^b \lg^b k$ (for some constants $a > 0$ and $b$), then

$$M(n) = O \left( \frac{1}{n l \lg \frac{1}{l \lg n}} \right).$$

To prove Theorem 1 we need some preliminary.

**Lemma 2.** Let $d_k$ satisfy the condition of Theorem 1. There exists a constant $\delta$ (that depends on $a$ and $b$), such that for any $n$, if $k = (\delta n / \lg n)^{1+a}$, then $\frac{kd_k}{2} > 2n$.

**Proof.** Let $\delta$ be such that

$$\frac{\delta}{4(1+a)} b \left( 1 - \frac{\lg n}{\lg n} \right) > 1.$$  

Such a $\delta$ exists since $\lim_{n \to \infty} \frac{\lg n}{\lg n} = 0$.

Then

$$\frac{kd_k}{2} = \frac{1}{2} \left( \frac{\delta n / \lg n}{1+a} \right)^{1+a} \frac{1}{\delta n / \lg n} = \frac{1}{2} \left( \frac{\delta n / \lg n}{1+a} \right)^{1+a} \frac{\delta n / \lg n}{1+a} = \frac{1}{2} \cdot \frac{\delta n}{\lg n} \cdot \frac{1}{(1+a)^a} \lg^b \left( \frac{\delta n / \lg n}{1+a} \right) \cdot \frac{\delta n}{\lg n} \geq \left( \frac{\delta}{4(1+a)^b} \right)^{1+1} \left( 1 - \frac{\lg n}{\lg n} \right) > 2n.$$

**Proof of Theorem 1.** Decreasing $k$ obtained in Lemma 2 and the number of polynomials, if necessary, we may assume that

$$2n < \sum_{i=1}^{m(\delta d_k)} \deg w_i(\alpha) < 2n + n^{1-e_1}$$

for some $0 < e_1 < 1 - \frac{1}{1+a}$ (Lemma 2), where $\{w_1, \ldots, w_m\} \subseteq D_k$.

Let $M(n) = n (s(n) \lg n) \frac{1}{l \lg \frac{1}{l \lg n}}$, where $s(n)$ is a non-decreasing function. It is sufficient to prove that $s(n)$ is bounded.
(2) implies that

\[ M(n) = n (s(n)\log(n)) \leq \sum_{i=1}^{m} M(\deg w_i(\alpha) - 1) \leq \]

\[ \sum_{i=1}^{m} \deg w_i(\alpha) (s(\deg w_i(\alpha)) \log \deg w_i(\alpha)) \frac{1}{\log(1+a)} \leq \]

\[ \leq \left( \sum_{i=1}^{m} \deg w_i(\alpha) \right) (s(k)\log k) \frac{1}{\log(1+a)} \leq (2n + n^{-1}) (s(k)\log k) \frac{1}{\log(1+a)}. \]

The last inequality follows from (6).

Hence

\[ (s(n)\log(n)) \frac{1}{\log(1+a)} \leq (2n + n^{-1}) (s(k)\log k) \frac{1}{\log(1+a)}, \]

and therefore

\[ s(n) \leq (2n^{2}+n^{-1}) \frac{\log(k)}{\log(n)} s(k) \leq \]

\[ \leq (2n^{2}+n^{-1}) \frac{\log([\delta n/\log(n)])}{\log(n)} s(n^{2}), \]

for some \( 0 < \epsilon < 1 \). (In the last expression we substituted \((\delta n/\log(n))^{-1}\) for \( k \). See Lemma 2.) From the last inequality we obtain that

\[ s(n) \leq (2n^{2}+n^{-1}) \frac{\log(k)}{\log(n)} \left[ 1 + \frac{\delta}{\log(n)} - \frac{\log(n)}{\log(n)} \right] s(n^{2}) \leq \]

\[ \leq 1 + a + \epsilon n^{-1} \frac{1}{1 + a} \left\{ 1 + \frac{1}{\log(bn)} \right\} s(n^{2}) \text{, say}. \]

Therefore

\[ s(n) \leq \left[ 1 + \frac{1}{\log(n)} \right] s(n^{2}) \]

for sufficiently large \( n \)'s.

Hence

\[ s(n) \leq \gamma \prod_{i=0}^{m} \left[ 1 + \frac{1}{\sqrt[\epsilon]{\log(n)}} \right]. \]

for some constant \( \gamma \). Taking logarithms of both sides we obtain that
\[ \lg s(n) \leq \lg \gamma + \sum_{i=0}^{\frac{-\lg s/\lg e}{2}} \lg \left( 1 + \frac{1}{\sqrt{e} \lg n} \right) \leq \lg \gamma + \sum_{i=0}^{\frac{-\lg s/\lg e}{2}} \frac{2}{\sqrt{e} \lg n} = \]
\[ \lg \gamma + \frac{2}{\sqrt{\lg n}} \sum_{i=0}^{\frac{-\lg s/\lg e}{2}} \frac{1}{\sqrt{e} \lg n} = \frac{1}{\sqrt{e} \lg n} - 1 \]

The last expression is bounded, thus \( s(n) \) is bounded. □

In the next section we derive an \( O(n \lg n) \) upper bound using sets \( D_k \) which consist of powers of cyclotomic polynomials.

**An upper bound.** To establish sets of monic polynomials generating pairwise comaximal ideals which can be used in the algorithm given in the previous section we need the following proposition.

**Proposition 1.** a) Let \( i \) and \( j \) be positive integers. The cyclotomic polynomials \( \Phi_i(\alpha) \) and \( \Phi_j(\alpha) \) generate comaximal ideals if and only if \( \frac{i}{j} \) is not a power of a prime number. b) For an \( i \geq 1 \), \( \Phi_i(\alpha) \) and \( \alpha \) generate comaximal ideals.

We remind the reader that \( \Phi_k(\alpha) \) is the minimal (monic) polynomial of the \( k \)th primitive root of 1 \( \zeta_k = e^{\frac{2\pi i}{k}} \). \( \Phi_k(\alpha) = \frac{\alpha^k - 1}{\prod_{d|k, d \neq k} \Phi_d(\alpha)} \) has integral coefficients, and \( \deg \Phi_k(\alpha) = \phi(k) \), where \( \phi \) is the Euler function.

**Proof.** a) It is known from [2] that the resultant of \( \Phi_i(\alpha) \) and \( \Phi_j(\alpha) \) is equal to 1. Hence the equation \( p(\alpha) \Phi_i(\alpha) + q(\alpha) \Phi_j(\alpha) \) has a solution (cf. [9], p. 136) and the result follows. b) Since \( \Phi_k(0) = \pm 1 \), \( \Phi_k(\alpha) = \pm 1 + \alpha p(\alpha) \) for some polynomial \( p(\alpha) \). Therefore 1 belongs to the ideal generated by \( \alpha \) and \( \Phi_k(\alpha) \) □

We construct a sequence of sets of monic polynomials generating pairwise comaximal ideals as follows:

Define \( S'_k(S''_k) = \{ \Phi_i(\alpha) | i \text{ is a square-free integer having an odd (even) number of prime factors and satisfying } \phi(i) \leq k \} \). It easily follows from Proposition 1 that elements of \( S'_k(S''_k) \) generate
pairwise comaximal ideals. Let $S_k$ be that of $S'_k$, $S''_k$, who contains more elements. The number of
elements of $S_k$ is $O(k)$ since the asymptotic density of square-free integers not exceeding $k$ is $\frac{6}{\pi^2}k$, cf.
[12], p. 243. (Actually, it is not hard to show that the number of elements of $S'_k$ and $S''_k$ is, asymptot-
ically, $\frac{3}{\pi^2}k$.) Let $D_k$ be any maximal set of polynomials of degree not exceeding $k$ generating pairwise
comaximal ideals that contains $S_k$. Then $d_k = O(k)$. Now, having a sequence $D_1$, $D_2$, we can per-
form the algorithm given in the previous section.

Theorem 2. $M(n) = O(n\log n)$.

Proof. We apply (2) recursively with an appropriate $D_k$'s. Since $d_k = O(k)$, the result follows from
Theorem 1 □.

This completes the description of the algorithm.

Remark. Let $C(k)$ denote $\max \{d \mid \text{there exist } d \text{ monic polynomials of degree not exceeding } k \gen-
erating pairwise comaximal ideals\}$. Obviously, polynomials generating comaximal ideals must be
coprime modulo 2, therefore $C(k) \leq \frac{1}{k} \sum_{i|k} \mu(i)2^{k/i} < \frac{2^{k}}{k}$, cf. [11] p. 115. Here $\mu$ denotes the Môbius
function.

The above consideration together with the construction given after Proposition 1 implies that

$$ck \leq C(k) \leq 2^{k/k}$$

for some constant $c$.

Improving $O(k)$ lower bound on $C(k)$ can improve $O(n\log n)$ upper bound on $M(n)$, cf. Theorem
1. However $O(k)$ lower bound cannot be improved if we restrict ourselves to cyclotomic polynomi-
als:

Let $D_k$ be any maximal set of cyclotomic polynomials of the degree not exceeding $k$ generating
pairwise comaximal ideals. Obviously, $d_k$ is less than the number of positive integers $l$ satisfying
$\phi(l) \leq k$. The last number is $O(k)$, cf. [3].

Finally, we prove the "universal property" of bilinear algorithms for polynomial multiplication
over the ring of integers.
Theorem 3. Let $A$ be a bilinear algorithm for polynomial multiplication over $\mathbb{Z}$. Then for any ring $R$, $A^R$ is a bilinear algorithm for polynomial multiplication over $R$. (See Introduction for the definition of $A^R$.)

Proof. Treating multiplying by a positive integer $m$ as a sum of $m$ copies, we may assume that $A$ contains only instructions of the type (i) or (ii) of the set (1) of the Introduction. Hence the correctness of $A$ can be proved by using only the commutativity and the associativity of $+$ and the distributivity of $\ast$ over $+$. This proof relativized to $R$ gives the correctness of $A^R$. Obviously, $A^R$ is bilinear. $\square$
References


