MATHEMATICS OF COMPUTATION, VOLUME 26, NUMBER 120, OCTOBER 1972

# Optimal Multiplication Chains for Computing a Power of a Symbolic Polynomial

By W. Morven Gentleman

**Abstract.** This paper shows that in a certain model of symbolic manipulation of algebraic formulae, the simple method of computing a power of a symbolic polynomial by repeated multiplication by the original polynomial is, in essence, the optimal method.

**Introduction.** There is considerable literature [2] on optimal multiplication chains for computing a power of an integer. A rudimentary result is that, whereas simply repeatedly multiplying by the original integer can be arbitrarily bad, the well-known scheme of repeatedly squaring the integer, then taking appropriate products of the powers so formed, uses at worst twice the minimal number of multiplications required. Recently, several workers in the subject of symbolic manipulation of algebraic formulae have discovered that to compute a power of a symbolic polynomial, however, repeated squaring can be considerably more expensive than merely repeatedly multiplying by the original polynomial (already being 50% more expensive for just the 4th power of long polynomials). The purpose of this note is to establish the very strong result that, under appropriate conditions, repeated multiplication by the original polynomial is, in fact, the optimal multiplication chain for computing a power of a symbolic polynomial.

**Computational Model.** The computational model which we shall assume is that appropriate for sparse polynomials, namely that whatever the number of indeterminates in the polynomial, and whatever the degree of the polynomial in each indeterminate, there are only $n$ nonzero terms. That is, we assume $P_1$ is a multinomial of $n$ terms

$$P_1 = t_1 + t_2 + \cdots + t_n, \tag{1}$$

where the monomials $t_1, \cdots, t_n$ are distinct but can be of arbitrary degree in each of an arbitrary number of variables. We shall denote the number of nonzero terms in a polynomial $P$ as $L(P)$.

We assume further that each of the polynomials

$$P_j = (t_1 + t_2 + \cdots + t_n)^j \tag{2}$$

for all powers $j$ up to that desired contain, when expanded, exactly the number

Received March 23, 1971.

*AMS* 1969 *subject classifications*. Primary 1230, 68XX.

*Key words and phrases*. Symbolic algebraic manipulation, optimal multiplication chains, complexity.

Copyright © 1972, American Mathematical Society

of terms indicated by the $n$-term multinomial expansion, with no further collection of like terms possible. One can readily show, by induction or by combinatoric arguments, that, using the usual notation for binomial coefficients,

$$L(P_j) = \binom{n-1+j}{n-1}. \tag{3}$$

The cost of multiplying two sparse polynomials is the cost of multiplying each of the terms of the first by each of the terms of the second, then collecting like terms of the product so formed. The cost of the additions in actually collecting like terms is negligible, since the polynomials being sparse implies the number of like terms is small. Recognizing like terms, however, requires that the product monomials be ordered (say increasing in degree lexicographically by indeterminate) and since there is no simple way to generate the product monomials in order, some sorting is required. If the terms of the factor polynomials are so ordered, however, the product monomials corresponding to a specific term of one factor with each term of the other factor will also be ordered, so the necessary sorting is to merge these sequences.

To multiply a polynomial $P$ by a polynomial $Q$ thus costs $L(P)L(Q)$ monomial multiplications (coefficient multiplication plus exponent set addition) plus the cost of this merge, which is $L(P)L(Q) \log_2 \min (L(P), L(Q))$ comparisons. In the results that follow, we shall measure cost merely as $L(P)L(Q)$; consideration of the log term in the merge merely strengthens them.*

We note in passing that this computational model of polynomial manipulation is not the only one amenable to analysis. Another common model is that of dense polynomials: the polynomials are of degrees $d_1, d_2, \cdots, d_m$ in each of $m$ different variables, and each possible term is present. Sorting here is unnecessary as product monomials can be generated in order. Moreover, fast Fourier transform methods can be used to multiply the polynomials without forming and summing the individual product monomials [1]. Nevertheless, the superiority of repeated multiplication over repeated squaring has been shown for dense polynomials too, provided $m$ is not equal to one.

**Multiplication Chain Cost Comparison.** To compare multiplication chains, we shall assume that somehow $P_r$ has been obtained, and $P_{r+s}$ is desired, $s \leqq r$. We shall also assume $P_s$ is available free, so the first method is to produce $P_{r+s}$ by multiplying $P_r$ and $P_s$, at cost

$$\text{Cost I} = L(P_r)L(P_s) = \binom{n-1+r}{n-1}\binom{n-1+s}{n-1}. \tag{4}$$

The second method consists of ignoring $P_s$, and multiplying $P_r$ by $P_1$ to produce $P_{r+1}$, then $P_{r+1}$ by $P_1$ to produce $P_{r+2}$, etc., until $P_{r+s}$ is produced. The cost of this repeated multiplication is

* The referee has pointed out that if $P$ and $Q$ are the same, i.e., if we square a polynomial, a special polynomial multiplication routine can be used that avoids half the apparent multiplications and comparisons. The analysis in this paper ignores this possibility, but allowing for it only has the effect of increasing the size necessary for $r$ to be sufficiently large in the theorem and corollary.

$$\text{Cost II} = \sum_{j=r}^{r+s-1} L(P_1)L(P_j) = n \sum_{j=r}^{r+s-1} \binom{n-1+j}{n-1}$$

$$(5) \qquad = n\binom{n+r+s-1}{n} - n\binom{n+r-1}{n}$$

$$= (r+s)\binom{n-1+r+s}{n-1} - r\binom{n-1+r}{n-1}$$

where we have used the identity

$$(6) \qquad \sum_{j=0}^{k} \binom{n-1+j}{n-1} = \binom{n+k}{n}.$$

If Cost I is bigger than Cost II, then using $P_s$ is more expensive than repeated multiplication, even though $P_s$ is available free. We will show that this is exactly what happens.

THEOREM. (a) *If* $n > 3$, *then for any sufficiently large* $r$, *Cost* I *exceeds Cost* II *for all* $2 \leqq s \leqq r$.

(b) *If* $n = 3$, *then for any sufficiently large* $r$, *Cost* I *exceeds Cost* II *for all* $3 \leqq s \leqq r$, *but Cost* II *exceeds Cost* I *for* $s = 2$ *and all* $r$.

(c) *If* $n = 2$, *then Cost* II *exceeds Cost* I *for* $2 \leqq s \leqq r$ *and all* $r$.

*Proof.* For fixed $n$, the difference between Cost I and Cost II can be viewed as a polynomial in $s$ with coefficients that are polynomials in $r$. If $r$ is fixed sufficiently large, the signs of these coefficient polynomials depend only upon the leading term of each in $r$. By examining these leading terms, we can use Decartes rule of signs to count the number of positive roots of the polynomial in $s$. Evaluating it at a few points then sufficiently isolates the roots to prove (a) and the first part of (b):

$$\text{Cost I} - \text{Cost II}$$

$$= \binom{n-1+r}{n-1}\binom{n-1+s}{n-1} - (r+s)\binom{n-1+r+s}{n-1} + r\binom{n-1+r}{n-1}$$

$$= \frac{(n-1+r)\cdots(1+r)}{(n-1)!}\binom{n-1+s}{n-1}$$

$$- \frac{(n-1+r+s)\cdots(r+s)}{(n-1)!} + \frac{(n-1+r)\cdots(r)}{(n-1)!}$$

$$(7) \qquad = \frac{r^{n-1}}{(n-1)!}\binom{n-1+s}{n-1} + O(r^{n-2}s^{n-1})$$

$$- \frac{s^n}{(n-1)!} - \frac{r^n}{(n-1)!} - \frac{n(n-1+2s)r^{n-1}}{2(n-1)!} + O(r^{n-2}s^{n-1})$$

$$+ \frac{r^n}{(n-1)!} + \frac{n(n-1)r^{n-1}}{2(n-1)!} + O(r^{n-2}s^{n-1})$$

$$= -\frac{s^n}{(n-1)!} + \binom{n-1+s}{n-1}\frac{r^{n-1}}{(n-1)!} - \frac{nsr^{n-1}}{(n-1)!} + O(r^{n-2}s^{n-1}).$$

The notation $O(r^{n-2}s^{n-1})$ is used here to indicate a polynomial of degree not more than $n-2$ in $r$ nor $n-1$ in $s$. Since the binomial coefficient $\binom{n-1+s}{n-1}$ is a polynomial of degree $n - 1$ in $s$ with all coefficients positive, the leading coefficients in $r$ for each power of $s$ are given in (7). The coefficient of $s^n$ is negative, the coefficients of $s^{n-1}, \cdots, s^2$ are positive, the coefficient of $s^1$ is $(r^{n-1}/(n-1)!)\{\sum_{j=1}^{n-1} 1/j - n\}$ which is always negative, and the coefficient of $s^0$ is again positive. If $n \geqq 3$, there are thus three sign changes, and by Descartes' rule of signs, either one or three positive roots. By construction, one of these roots is at $s = 1$.

Since the leading coefficient of $s$ in (7) is negative, for fixed $n$ and $r$ and all sufficiently large $s$, Cost I $-$ Cost II is negative. But if $s = r$, we have from (7) that Cost I $-$ Cost II is $r^{2n-2}/(n-1)!(n-1)! + O(r^{2n-3})$, so that if the fixed $r$ is sufficiently large, Cost I $-$ Cost II is positive. This shows there are indeed three roots, but one is larger than $r$. It remains to locate the last root. Clearly, showing Cost I $-$ Cost II is positive at $s = 2$ suffices to show that this last root is less than 2, and hence proves (a). But Cost I $-$ Cost II at $s = 2$ is $\binom{n-1+r}{n-1}n\{(n-3)/2 - (n-1)/(1+r)\}$ from (4) and (5), so for $n > 3$ and $r$ sufficiently large, Cost I $-$ Cost II is positive. However, if $n = 3$, we note Cost I $-$ Cost II at $s = 2$ is negative for all $r$, proving the second part of (b). To locate the root for $n = 3$, we evaluate Cost I $-$ Cost II for $n = 3$, $s = 3$ getting $\frac{1}{2}(r^2 - 15r - 40)$ which is positive for sufficiently large $r$, so by the argument used previously, the last root for $n = 3$ is less than $s = 3$, proving the first part of (b). The explicit expansion of Cost I $-$ Cost II for $n = 2$ is $(1 - s)(1 + r + s)$ which is negative for all $2 \leqq s \leqq r$ and all $r$, proving (c).

**Comment.** It is clearly of interest to know what 'sufficiently large' means in parts (a) and (b) of this theorem. From the explicit form of Cost I $-$ Cost II for $s = 2$, and using induction on $s$, it can readily be proved that the smallest $r$, such that Cost I is not less than Cost II for all $2 \leqq s \leqq r$, is 5 if $n$ is 4, is 3 if $n$ is 5 or 6, and is 1 if $n$ is 7 or greater. Unfortunately, no proof has been found that the theorem applies for all $r$ greater than these, but extensive tabulation suggests this is so. Similarly, it can be shown that the smallest $r$, such that Cost I is not less than Cost II for all $3 \leqq s \leqq r$, is 18 if $n$ is 3, and more interestingly that, if $n$ is 3 and $s$ is even, the sequence $P_{r+2}, P_{r+4}, \cdots, P_{r+s}$ is cheaper than Cost I if $r$ is 8 or 9 but not smaller. Again, there is no proof available that this holds for all larger $r$, but tabulation indicates that it does.

COROLLARY. *For each $n > 3$, there is some power, $r(n)$, such that whatever the optimal multiplication chain is for computing $P_{r(n)}$, the optimal multiplication chain for computing $P_j$, $j > r(n)$, is to compute $P_{r(n)}$, then repeatedly multiply by $P_1$. For $n = 3$, there is an odd power ro and an even power re, such that, whatever the optimal multiplication chains are for computing $P_{ro}$ and $P_{re}$, the optimal multiplication chain for computing $P_{2j+1}$, $2j + 1 > ro$, is to compute $P_{ro}$ and then repeatedly multiply by $P_2$, and, similarly, the optimal multiplication chain for computing $P_{2j}$ is to compute $P_{re}$ and then repeatedly multiply by $P_2$.*

*Proof.* Follows directly from the theorem once we confirm that, for $n = 3$, the optimal multiplication chain cannot terminate as $P_r, P_{r+2}, \cdots, P_{r+2k}, P_{r+2k+1}$, because the chain $P_r, P_{r+1}, P_{r+3}, \cdots, P_{r+2k+1}$ is cheaper.

**Magnitude Comparison.** While the optimality of repeated multiplication has been shown, it might still be suspected that the costs are negligibly different. This is not so, as can be seen for long polynomials by comparing the cost of producing $P_{2r}$ by squaring $P_r$ (assuming $P_r$ is available free)

$$\text{Cost III} = \binom{n-1+r}{r}^2$$

$$(8) \qquad = \frac{(n^r + \tfrac{1}{2}r(r-1)n^{r-1} + O(n^{r-2}))^2}{(r!)^2}$$

$$= \frac{n^{2r}}{(r!)^2}\left(1 + \frac{r(r-1)}{n}\right) + O(n^{2r-2}),$$

with the full cost of producing $P_{2r}$ by repeated multiplication starting from $P_1$,

$$\text{Cost IV} = 2r\binom{n-1+2r}{2r} - n$$

$$(9) \qquad = 2r\,\frac{(n^{2r} + \tfrac{1}{2}(2r)(2r-1)n^{2r-1} + O(n^{2r-2}))}{(2r)!}$$

$$= \frac{2r}{(2r)!}\,n^{2r}\left(1 + \frac{r(2r-1)}{n}\right) + O(n^{2r-2}).$$

The ratio of these costs is

$$(10) \qquad \frac{\text{Cost III}}{\text{Cost IV}} = \frac{1}{2r}\binom{2r}{r}\left(1 - \frac{r^2}{n}\right) + O(n^{-2})$$

which shows that repeated squaring is essentially more expensive than repeated multiplication by a binomial factor. Computing the 4th power of long polynomials is already 1.5 times more expensive, computing the 8th power 8.75 times more expensive, and computing the 16th power 804 times more expensive. An effect of this magnitude cannot be ignored.

**Conclusion.** The importance of these results is not that they show the cheapest way to compute powers of a polynomial (suitably crafty substitution into the multinomial expansion is cheaper, although multiplication chains would probably be used in practice). Rather, they show the extent to which our intuition has been violated in attempting to apply to polynomials results that are true for integer operations. Moreover, they provide new insight into the general (and poorly understood) problem of symbolic substitution of algebraic arguments into an algebraic form.

**Acknowledgments.** This problem arose in connection with the Altran symbolic manipulator at Bell Telephone Laboratories and the author would like to thank W. S. Brown, A. J. Goldstein, and other members of the Altran group for interesting discussions on the subject. He would also like to thank P. C. Fischer for a useful suggestion in the proof of the theorem. This research was supported by a University of Waterloo Research Grant.

Department of Applied Analysis and Computer Science
University of Waterloo
Waterloo, Ontario, Canada

1. W. M. Gentleman & G. Sande, "Fast Fourier transforms—for fun and profit," *Proceedings of the* 1966 *Fall Joint Computer Conference, AFIPS*, Spartan Books, Washington, 1966, pp. 563–578.

2. D. Knuth, *The Art of Computer Programming*: Vol. II. *Seminumerical Algorithms*, Addison-Wesley, Reading, Mass., 1968.