DUALITY APPLIED TO THE COMPLEXITY OF MATRIX MULTIPLICATIONS AND OTHER BILINEAR FORMS[†]

J. Hopcroft
J. Musinski

Cornell University
Ithaca, New York 14850

Abstract

The paper considers the complexity of bilinear forms in a noncommutative ring. The dual of a computation is defined and applied to matrix multiplication and other bilinear forms. It is shown that the dual of an optimal computation gives an optimal computation for a dual problem. An nxm by mxp matrix product is shown to be the dual of an nxp by pxm or an mxn by nxp matrix product implying that each of the matrix products requires the same number of multiplications to compute. Finally an algorithm for computing a single bilinear form over a noncommutative ring with a minimum number of multiplications is derived by considering a dual problem.

Introduction

This paper is concerned with determining the minimum number of multiplications necessary to compute certain bilinear forms over a noncommutative ring. We define the dual of a set of expressions and the dual of a computation in such a manner that the dual of the computation of a set of expressions is a computation for the dual of the expressions. Furthermore, a computation and its dual both use the same number of multiplications. This implies that the minimum number of multiplications necessary to compute a set of expressions is the same as that to compute its dual.

The concept of duality is applied to matrix multiplication. The dual of a set of expressions representing the multiplication of two matrices is a set of expressions representing another matrix multiplication problem where the dimensions of the matrices have been permuted. Thus we are able to show that the minimum number of multiplications necessary to compute an nxm by mxp matrix product is the same as that required to compute an nxp by pxm or an mxn by nxp product. Optimal programs follow from previous results. Dual statements of several interesting theorems are presented. Finally it is shown that Strassen's algorithm for 2x2 by 2x2 matrix multiplication is unique to within a linear transformation.

Definition of a Computation

Let $\mathscr{C}$ be a commutative ring with a unit element and let ψ be a finite set of indeterminants. Let $\mathscr{R}$ be the noncommutative ring obtained by extending $\mathscr{C}$ by multi-

[†] This research was supported by the Office of Naval Research Grant N00014-67-A-0077-0021.

nomial expressions of the elements of ψ. Throughout this section and the next F will denote the set of bilinear forms

$$\left\{\sum_{j=1}^{m}\sum_{k=1}^{n} c_{ijk}\, a_j\, x_k \mid 1 \leq i \leq p,\ a_j,\ x_k \in \psi,\ c_{ijk} \in \mathscr{C}\right\}.$$

Similarly a and x will denote the column vectors $(a_1,a_2,\ldots,a_m)^T$ and $(x_1,x_2,\ldots,x_n)^T$.

We consider the notion of a computation (see Ostrowski [4]) as a sequence of instructions $f_i = g_i \circ h_i$ where $\circ$ stands for one of the binary operations of multiplication, addition or subtraction. Each f_i is a new variable and each g_i or h_i is either an element of $\mathscr{C} \cup \psi$ or a previously computed f_j. A multiplication of two elements of $\mathscr{R}$, neither of which is in $\mathscr{C}$ is assumed to take one unit of time. All other operations require no time to perform. The motivation for counting only multiplications between elements in $\mathscr{R}-\mathscr{C}$ is that in applications the elements of ψ may be large matrices (Strassen [5]) and thus the scheme is not only mathematically tractable but also reflects the actual computation time within a constant factor. It is well known that without division computations of bilinear forms can be reduced to computing linear combinations of products of pairs of linear forms. This motivates the following definition of a computation. Express the set of expressions F as $(a^TX)^T$† where X is an mxp matrix with elements of the form $\sum_{i=1}^{n} c_i\, x_i$, $c_i \in \mathscr{C}$. A <u>computation</u> of F is an expression of the form $M(Pa \cdot Rx)$ where M, P and R are matrices of dimensions pxq, qxm and qxn whose elements are from $\mathscr{C}$, the symbol $\cdot$ indicates element by element multiplication, and $M(Pa \cdot Rx) = (a^TX)^T$. Since the straight forward method of evaluating $M(Pa \cdot Rx)$ uses q multiplications between elements in $\mathscr{R}-\mathscr{C}$, the computation is said to have q multiplications.

<u>Duality</u>

This section defines the dual of a set of bilinear forms and the dual of a computation. It is then shown that the dual of any computation of F computes the dual of F.

Let b be the column vector $(b_1,b_2,\ldots,b_p)^T$, $b_i \in \psi$. The <u>left dual</u> of F is the system of equations given by $(b^TX^T)^T$. Let $M(Pa \cdot Rx) = (a^TX)^T$ be a computation of F. The <u>P-dual</u> of the computation is the computation $P^T(M^Tb \cdot Rx)$.

<u>Lemma 1</u>: The P-dual of any computation of a system of expressions F computes the left dual of F.

<u>Proof</u>: Let $M(Pa \cdot Rx) = (a^TX)^T$ be a computation of F. We must show that $P^T(M^Tb \cdot Rx)$ is a computation of $(b^TX^T)^T$. Let D be a diagonal matrix whose diagonal elements are the elements of the column vector Rx. Then $(M(Pa \cdot Rx))^T = (Pa)^T\, D\, M^T$. Since the elements

† By $(a^TX)^T$ we mean the matrix whose ij<u>th</u> element is the ji<u>th</u> element of a^TX. Since the elements are from a noncommutative ring rather than a field $(a^TX)^T \neq X^Ta$ in general.

of P commute with the elements of a, $(Pa)^T = a^T P^T$. Now $a^T P^T D M^T = a^T X$ for all a implies $P^T D M^T = X$ which in turn implies $b^T M D P = b^T X^T$ for all b. Thus $(M^T b)^T DP = b^T X^T$ implying $P^T(M^T b \cdot Rx) = (b^T X^T)^T$.

In a similar manner the system of expressions F can be expressed as Ax where A is a pxn matrix with elements of the form $\sum_{i=1}^{m} c_i a_i$, $c_i \in \mathscr{C}$. The right dual of F is the system of equations given by $A^T b$. If M(Pa·Rx) is a computation of F, then the R-dual of the computation is the computation $R^T(Pa \cdot M^T b)$. The R-dual of a computation of a system of expressions F computes the right dual of F.

Lemma 2: The R-dual of any computation of a system of expressions F computes the right dual of F.

Proof: The proof is analogous to that of Lemma 1.

Theorem 3: There is a computation for the system of expressions computed by M(Pa·Rx) with q multiplications if and only if there is a computation with q multiplications for each of the systems of expressions computed by $P^T(M^T b \cdot Rx)$, $R^T(Pa \cdot M^T b)$, $R^T(M^T b \cdot Pa)$, $P^T(Rx \cdot M^T b)$ and M(Rx·Pa).

Proof: The result follows from the fact that a computation, its R-dual and its P-dual each have the same number of multiplications.

Let M(Pa·Rx) be a computation of F and let c be a column vector such that M(Pa·Rx)= c. Let T, U, V be pxp, mxm, nxn matrices respectively with elements from $\mathscr{C}$. A transformation of a vector c of bilinear forms is the result of replacing each element of a and x by the corresponding elements of Ua, Vx in Tc. A transformation of the computation M(Pa·Rx) is the computation TM(PUa·RVx).

Lemma 4: The transformation of a computation of c is a computation of the transformation of c.

Corollary 5: If c' is a transformation of c, then c' can be computed in q multiplications if c can be computed in q multiplications. If T, U and V are nonsingular and c' can be computed in q multiplications then c can be computed in q multiplications.

Matrix Multiplication

Let A, B and C be mxn, nxp and mxp matrices whose elements are from ψ. We will show that there is a computation of AB with q multiplications if and only if there are computations for

$$A^T C,\ B^T A^T,\ BC^T,\ C^T A,\ CB^T$$

with q multiplications. In other words the number of multiplications needed to compute the product of an mxn matrix with an nxp matrix is the same as that required to compute an nxm by mxp, pxn by nxm, etc. If one uses the ordinary algorithms which require nmp multiplications then the result is not surprising. However, the result claims that no matter what method is used the minimum number of multiplications is the same.

Let a,b,c be column vectors whose elements are those of A, B and C respectively in row order

$$(\text{e.g. } a = (a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{mn})^T).$$

The ij^{th} element of AB is $\sum_{k=1}^{n} a_{ik}\, b_{kj}$. Therefore, there exist matrices M, P and R of dimensions mp x q, q x mn and q x np whose elements are from $\mathscr{C}$ such that $M(Pa\cdot Rb)$ is a computation for AB.

<u>Theorem 6</u>: The following statements are equivalent.

a) $M(Pa\cdot Rb)$ computes AB in row order using q multiplications.

b) $P^T(M^Tc\cdot Rb)$ " CB^T " " " " " "

c) $R^T(M^Tc\cdot Pa)$ " $(C^TA)^T$ " " " " " "

d) $M(Rb\cdot Pa)$ " $(B^TA^T)^T$ " " " " " "

e) $P^T(Rb\cdot M^Tc)$ " $(BC^T)^T$ " " " " " "

f) $R^T(Pa\cdot M^Tc)$ " A^TC " " " " " "

<u>Proof</u>: We will prove only that (a) => (b). Let D_B be the mn by mp matrix $\begin{bmatrix} B0 & & 0 \\ 0B & \ddots & 0 \\ 00 & \ddots & B \end{bmatrix}$.

$M(Pa\cdot Rb)$ computes AB in row order implies that $M(Pa\cdot Rb) = (a^TD_B)^T$ by definition of a computation. This in turn implies

$$P^T(M^Tc\cdot Rb) = (c^TD_B^T)^T \text{ by Lemma 1.}$$

Thus $P^T(M^Tc\cdot Rb)$ computes CB^T in row order.

<u>Corollary 7</u>: The minimum number of multiplications required to multiply mxn by nxp matrices without using commutativity is the same as to multiply nxm by mxp, nxp by pxm, pxm by mxn, pxn by nxm, or mxp by pxn.

Theorem 6 leads to new algorithms for multiplying various size matrices together. Some of the new algorithms are optimal, others are at least improvements over the best currently known. For example, in [3] it is shown that $\lceil (3pn + \max(n,p))/2 \rceil$ multiplications is sufficient for px2 by 2xn matrix multiplication. It follows that $\lceil (3pn + \max(n,p))/2 \rceil$ multiplications is sufficient for 2xp by pxn matrix multiplication. Since $\lceil 7n/2 \rceil$ multiplications are necessary and sufficient for 2x2 by 2xn matrix multiplication [3] it follows that $\lceil 7n/2 \rceil$ multiplications are necessary and sufficient for 2xn by nx2 matrix multiplication. Similarly since 15 multiplications are necessary and sufficient for 3x2 by 2x3 matrix multiplication, 15 multiplications are necessary and sufficient for 3x3 by 3x2 matrix multiplication.

The number of multiplications necessary to compute the product of two 3x3 matrices is an interesting open problem. If 21 or fewer multiplications are sufficient then the asymptotic growth rate of Strassen's method [5] could be improved. An examination of 3x2 by 2x3 and 3x3 by 3x2 matrix multiplication algorithms may shed some insight on the development of an algorithm for 3x3 by 3x3 matrix multiplication.

Let A, X, C and Y be 3x2, 2x3, 3x3 and 3x2 matrices whose elements are from ψ. Then

$$AX = \begin{bmatrix} m_1 + m_2 & -m_2 - m_3 + m_7 - m_8 & -m_1 - m_5 - m_{13} + m_{15} \\ -m_1 - m_4 + m_8 - m_9 & m_3 + m_4 & -m_3 - m_6 + m_{11} - m_{12} \\ -m_2 - m_6 + m_{13} - m_{14} & -m_4 - m_5 + m_{10} - m_{11} & m_5 + m_6 \end{bmatrix}$$

where

$m_1 = (a_{11}-a_{12})x_{11}$ $\quad m_6 = (-a_{31}+a_{32})x_{23}$ $\quad m_{11} = (a_{22}-a_{31}+a_{32})(+x_{12}+x_{13}+x_{23})$

$m_2 = a_{12}(x_{11}+x_{21})$ $\quad m_7 = (a_{11}+a_{21})(x_{11}+x_{12}+x_{21}+x_{22})$ $\quad m_{12} = (-a_{21}+a_{22}-a_{31}+a_{32})(+x_{12}+x_{13})$

$m_3 = a_{21}x_{12}$ $\quad m_8 = (a_{11}-a_{12}+a_{21})(x_{11}+x_{21}+x_{22})$ $\quad m_{13} = (a_{12}+a_{31})(x_{11}-x_{23})$

$m_4 = a_{22}x_{22}$ $\quad m_9 = (a_{11}-a_{12}+a_{21}-a_{22})(x_{21}+x_{22})$ $\quad m_{14} = (-a_{12}-a_{32})(x_{21}+x_{23})$

$m_5 = a_{31}(x_{13}+x_{23})$ $\quad m_{10} = (a_{22}+a_{32})(+x_{12}+x_{13}+x_{22}+x_{23})$ $\quad m_{15} = (a_{11}+a_{31})(x_{11}+x_{13})$

If

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & -1 & 1 & -1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

then $M(Pa \cdot Rx)$ computes AX in row order with 15 multiplications. By Theorem 6 $P^T(M^Tc \cdot RVy)$ is an optimal algorithm for CY. Thus

$$BY = \begin{bmatrix} n_1 + n_7 + n_8 + n_9 + n_{15} & -n_1 + n_2 - n_8 - n_9 + n_{13} - n_{14} \\ n_3 + n_7 + n_8 + n_9 - n_{12} & n_4 - n_9 + n_{10} + n_{11} + n_{12} \\ n_5 - n_6 - n_{11} - n_{12} + n_{13} + n_{15} & n_6 + n_{10} + n_{11} + n_{12} - n_{14} \end{bmatrix}$$

where

$n_1 = (c_{11}-c_{13}-c_{21})y_{11}$ $\quad n_4 = (-c_{21}+c_{22}-c_{32})y_{22}$

$n_2 = (c_{11}-c_{12}-c_{31})(y_{11}+y_{12})$ $\quad n_5 = (-c_{13}-c_{32}+c_{33})(y_{31}+y_{32})$

$n_3 = (-c_{12}+c_{22}-c_{23})y_{21}$ $\quad n_6 = (-c_{23}-c_{31}+c_{33})y_{32}$

$$\begin{aligned} n_7 &= c_{12}(y_{11}+y_{12}+y_{21}+y_{22}) & n_{12} &= -c_{23}(y_{21}+y_{31}) \\ n_8 &= (-c_{12}+c_{21})(y_{11}+y_{12}+y_{22}) & n_{13} &= (-c_{13}+c_{31})(y_{11}-y_{32}) \\ n_9 &= -c_{21}(y_{12}+y_{22}) & n_{14} &= -c_{31}(y_{12}+y_{32}) \\ n_{10} &= c_{32}(y_{21}+y_{22}+y_{31}+y_{32}) & n_{15} &= c_{13}(y_{11}+y_{31}) \\ n_{11} &= (c_{23}-c_{32})(y_{21}+y_{31}+y_{32}) & & \end{aligned}$$

The algorithm for AX is the union of three optimal algorithms that compute

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, \begin{bmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{bmatrix}\begin{bmatrix} x_{11} & x_{13} \\ x_{21} & x_{23} \end{bmatrix} \text{and} \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}\begin{bmatrix} x_{12} & x_{13} \\ x_{22} & x_{23} \end{bmatrix}$$

respectively such that each diagonal component of AX is computed with exactly two multiplications. Furthermore, both algorithms computing a given diagonal component, compute it with the same two multiplications. Each of the three algorithms uses seven multiplications, but each pair of algorithms has two multiplications in common. Thus only 15 multiplications are used in computing AX.

The algorithm for CY is the dual of the algorithm for AX. Thus, there is a dual construction for it. This construction is described briefly below and followed by an example.

> Let W be the 3x2 matrix such that W = CY. Construct optimal algorithms that compute

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} = \begin{bmatrix} w_{11} - c_{13}y_{31} & w_{12} - c_{13}y_{32} \\ w_{21} - c_{23}y_{31} & w_{22} - c_{23}y_{32} \end{bmatrix}$$

$$\begin{bmatrix} c_{11} & c_{13} \\ c_{31} & c_{33} \end{bmatrix}\begin{bmatrix} y_{11} & y_{12} \\ y_{31} & y_{32} \end{bmatrix} = \begin{bmatrix} w_{11} - c_{12}y_{21} & w_{12} - c_{12}y_{22} \\ w_{31} - c_{32}y_{21} & w_{32} - c_{32}y_{22} \end{bmatrix}, \text{ and}$$

$$\begin{bmatrix} c_{22} & c_{23} \\ c_{32} & c_{33} \end{bmatrix}\begin{bmatrix} y_{21} & y_{22} \\ y_{31} & y_{32} \end{bmatrix} = \begin{bmatrix} w_{21} - c_{21}y_{11} & w_{22} - c_{21}y_{12} \\ w_{31} - c_{31}y_{11} & w_{32} - c_{31}y_{12} \end{bmatrix}$$

such that each c_{ii} appears in exactly two linear combinations which are left hand sides of multiplications in each of the two algorithms involving c_{ii}. Furthermore, if α and β are the right hand sides of the two multiplications in one algorithr, then α and β are the right hand sides in the other. Each pair of multiplications with the same right hand whose left hand sides contain c_{ii} are merged by the formula

$$\text{merge}\left((c_{ii} + \ell_1)\,\alpha,\ (c_{ii} + \ell_2)\alpha\right) = (c_{ii} + \ell_1 + \ell_2)\alpha,$$

where ℓ_1 and ℓ_2 are linear combinations of the components of C. Each of the three original algorithms contains seven multiplications and between each pair of algorithms, two pairs of multiplications are merged. Thus the composite algorithm uses 15 multiplications in computing CY.

The following example should clarify the above description.

Example

$$\begin{bmatrix} c_{11}c_{12} \\ c_{21}c_{22} \end{bmatrix} \begin{bmatrix} y_{11}y_{12} \\ y_{21}y_{22} \end{bmatrix} = \begin{bmatrix} n_1-n_4+n_6-n_7 & n_1+n_2 \\ -n_1+n_4+n_5+n_7 & -n_1+n_3+n_5+n_7 \end{bmatrix}$$

$$\begin{bmatrix} c_{11}c_{13} \\ c_{31}c_{33} \end{bmatrix} \begin{bmatrix} y_{11}y_{12} \\ y_{31}y_{32} \end{bmatrix} = \begin{bmatrix} n_8+n_9 & n_8-n_{11}+n_{13}-n_{14} \\ -n_8+n_{10}+n_{12}+n_{14} & -n_8+n_{11}+n_{12}+n_{14} \end{bmatrix}$$

$$\begin{bmatrix} c_{22}c_{23} \\ c_{32}c_{33} \end{bmatrix} \begin{bmatrix} y_{21}y_{22} \\ y_{31}y_{32} \end{bmatrix} = \begin{bmatrix} n_{15}+n_{16} & n_{16}+n_{18}+n_{20}+n_{21} \\ -n_{15}+n_{17}+n_{19}+n_{21} & n_{17}-n_{18} \end{bmatrix}$$

where

$n_1=c_{12}(y_{12}+y_{22})$

$n_2=(c_{11}-c_{12})y_{12}$

$n_3=(c_{21}-c_{22})(y_{21}-y_{22})$

$n_4=c_{21}(y_{11}-y_{12}+y_{21}-y_{22})$

$n_5=(c_{12}+c_{22})y_{21}$

$n_6=(c_{11}+c_{21})y_{11}$

$n_7=(c_{12}+c_{21})(y_{12}-y_{21}+y_{22})$

$n_8=c_{13}(y_{11}+y_{31})$

$n_9=(c_{11}-c_{13})y_{11}$

$n_{10}=(c_{31}-c_{33})(-y_{31}+y_{32})$

$n_{11}=c_{31}(-y_{11}+y_{12}-y_{31}+y_{32})$

$n_{12}=(c_{13}+c_{33})y_{32}$

$n_{13}=(c_{11}+c_{31})y_{12}$

$n_{14}=(c_{13}+c_{31})(y_{11}+y_{31}-y_{32})$

$n_{15}=c_{23}(y_{21}+y_{31})$

$n_{16}=(c_{22}-c_{23})y_{21}$

$n_{17}=(-c_{32}+c_{33})y_{32}$

$n_{18}=c_{32}(-y_{22}-y_{32})$

$n_{19}=(-c_{23}-c_{33})(-y_{31}+y_{32})$

$n_{20}=(-c_{22}-c_{32})(y_{21}-y_{22})$

$n_{21}=(c_{23}+c_{32})(y_{21}+y_{32})$

Then

$$CY = \begin{bmatrix} m_1-m_4+m_6-m_7+m_8 & m_1+m_2+m_8-m_{10}-m_{12} \\ -m_1+m_4+m_5+m_7+m_{13} & -m_1+m_3+m_5+m_7+m_{14}+m_{15} \\ -m_8+m_9+m_{11}+m_{12}-m_{13}+m_{15} & -m_8+m_{10}+m_{11}+m_{12}-m_{14} \end{bmatrix}$$

where

$m_1=n_1$

$m_2=\text{merge}(n_2,n_{13})=(c_{11}-c_{12}+c_{31})y_{12}$

$m_3=\text{merge}(n_3,n_{20})=(c_{21}-c_{22}-c_{32})(y_{21}-y_{22})$

$m_4=n_4$

$m_5=\text{merge}(n_5,n_{16})=(c_{12}+c_{22}-c_{23})y_{21}$

$m_6=\text{merge}(n_6,n_9)=(c_{11}-c_{13}+c_{21})y_{11}$

$m_7=n_7$

$m_8=n_8$

$m_9=\text{merge}(n_{10},n_{19})=(-c_{23}+c_{31}-c_{33})(-y_{31}+y_{32})$

$m_{10}=n_{11}$

$m_{11}=\text{merge}(n_{12},n_{17})=(c_{13}-c_{32}+c_{33})y_{32}$

$m_{12}=n_{14}$

$m_{13}=n_{15}$

$m_{14}=n_{18}$

$m_{15}=n_{21}$

It is hoped that the techniques used above to construct algorithms for 3x2 by 2x3 and 3x3 by 3x2 matrix multiplication can be applied toward developing an optimal algorithm for 3x3 by 3x3 matrix multiplication. To date no algorithm for the latter using less than 24 multiplications has been found. However, there is no indication that

24 multiplications is the minimum.

Let D be a 3x3 matrix with elements from ψ. Then CD can be computed with 24 multiplications by partitioning the problem into a 3x2 by 2x3 and a 3x1 by 1x3 matrix multiplication problem or by partitioning the problem into a 3x3 by 3x2 and a 3x2 by 3x1 problem. These two partitions result in dual computations for 3x3 by 3x3 matrix multiplication. A third computation, also with 24 multiplications, can be obtained by using both of the above combining techniques. In this case find optimal algorithms that compute

$$\begin{bmatrix} c_{11}c_{12} \\ c_{21}c_{22} \end{bmatrix} \begin{bmatrix} d_{11}d_{12} \\ d_{21}d_{22} \end{bmatrix}, \begin{bmatrix} c_{11}c_{13} \\ c_{31}c_{33} \end{bmatrix} \begin{bmatrix} d_{11}d_{13} \\ d_{31}d_{33} \end{bmatrix} \text{and} \begin{bmatrix} c_{22}c_{23} \\ c_{32}c_{33} \end{bmatrix} \begin{bmatrix} d_{22}d_{23} \\ d_{32}d_{33} \end{bmatrix}$$

so that there are three pairs of multiplications used in computing the diagonal elements of CD such that the two multiplications in each pair are either the same multiplication or can be merged into a single multiplication. Thus 18 multiplications are used. To these are added the six multiplications $c_{13}d_{32}$, $c_{12}(d_{23}-d_{11})$, $(c_{23}+c_{13})d_{31}$, $(c_{21}+c_{31})d_{13}$, $c_{31}d_{12}$, and $c_{32}(d_{21}-d_{33})$. The computation of CD minus the above six multiplications is illustrated below.

$m_1=(c_{11}-c_{12})d_{11}$

$m_2=c_{12}(d_{21}+d_{11})$

$m_3=c_{21}d_{12}$

$m_4=c_{22}d_{22}$

$m_5=(c_{11}+c_{21})(d_{11}+d_{12}+d_{21}+d_{22})$

$m_6=(c_{11}-c_{12}+c_{21}-c_{22})(d_{21}+d_{22})$

$m_7=(c_{11}-c_{12}+c_{21})(d_{11}+d_{21}+d_{22})$

$m_8=(c_{11}-c_{13})d_{11}$

$m_9=c_{13}(d_{31}+d_{11})$

$m_{10}=c_{31}(d_{13}+d_{33})$

$m_{11}=(-c_{31}+c_{33})d_{33}$

$m_{12}=(c_{11}+c_{31})(d_{11}+d_{33})$

$m_{13}=(-c_{13}-c_{33})(d_{31}+d_{33})$

$m_{14}=(c_{13}+c_{31})(d_{11}-d_{33})$

$m_{15}=c_{32}(d_{23}+d_{33})$

$m_{16}=(-c_{32}+c_{33})d_{33}$

$m_{17}=c_{23}d_{32}$

$m_{18}=c_{22}d_{22}$

$m_{19}=(c_{23}+c_{33})(-d_{22}-d_{23}-d_{32}-d_{33})$

$m_{20}=(-c_{22}+c_{23}-c_{32}+c_{33})(-d_{22}-d_{23})$

$m_{21}=(c_{23}-c_{32}+c_{33})(-d_{22}-d_{23}-d_{33})$

$y_{11} = \text{merge}\ (m_1,m_8) + m_2 + m_9$

$y_{12} = -m_2 - m_3 + m_5 - m_7$

$y_{13} = -m_{10} + m_{12} - m_{14} - \text{merge}(m_1,m_8)$

$y_{21} = -m_4 - m_6 + m_7 - m_9 - \text{merge}(m_1,m_8)$

$y_{22} = m_3 + m_4 + m_{17}$

$y_{23} = -m_{10} - m_{18} + m_{20} - m_{21} - \text{merge}(m_{11},m_{16})$

$y_{31} = -m_9 - m_{13} + m_{14} - \text{merge}(m_{11},m_{16})$

$y_{32} = -m_{15} - m_{17} - m_{19} + m_{21}$

$y_{33} = \text{merge}\ (m_{11},m_{16}) + m_{10} + m_{15}$

In addition to helping find optimal (or better) algorithms for matrix multiplication, Theorem 6 or its more general form, Theorem 3, can be applied to previously published theorems to yield new results. For example, the following appear in [3]. For sake of simplicity the theorems are expressed for $\mathscr{C}$ being the integers. Some of the theorems are more general.

A set of vectors $v_1,v_2,\ldots,v_p$ with elements from $\mathscr{R}$ are <u>nondependent</u> such that $\sum_{i=1}^{p} c_i v_i$ is a vector with elements from $\mathscr{C}$, each c_i an element of $\mathscr{C}$ implies each $c_i=0$. Since an expression can be considered to be a one dimensional vector, the notion of nondependence applies also to expressions.

Lemma 8: (Winograd) Let A be an mxn matrix whose elements are from $\mathscr{R}$ and let $x = (x_1,x_2,\ldots,x_n)^T$ where $x_1 \in \psi$. If A has p nondependent columns, then any computation of Ax requires at least p multiplications.

Lemma 9: Let $\mathscr{C}$ be a field and let $F = \{f_1,\ldots,f_k,\ldots,f_p\}$ be a set of expressions, where $f_1,\ldots,f_k$ are nondependent and each can be expressed as a single product. If F can be computed with q multiplications, then there exists an algorithm for F with q multiplications in which k of the multiplications are $f_1,\ldots,f_k$.

Lemma 10: Let A and X be 2x2 and 2xn matrices respectively whose elements are from ψ. If an algorithm for computing AX has k multiplications of forms $a_{11}\alpha, (a_{12}+a_{21})\beta$, and $(a_{11}+a_{12}+a_{21})\gamma$, then the algorithm requires at least 3n + k multiplications.

Corollary 11: Let T be the group of transformations generated by the set of transformations which:

(1) interchange the two rows of A, two columns of X, or the two columns of A and the two rows of X.

(2) either add (subtract) row i of A to row j of A, column i of X to column j of A, or add (subtract) column i of A to column j of A and simultaneously subtract (add) row j of X to row i of X. By applying transformations from T we also have similar theorems for

(a) $(a_{11}+a_{21})\alpha$, $(a_{12}+a_{21}+a_{22})\beta$, $(a_{11}+a_{12}+a_{22})\gamma$

(b) $(a_{11}+a_{12})\alpha$, $(a_{12}+a_{21}+a_{22})\beta$, $(a_{11}+a_{21}+a_{22})\gamma$

(c) $(a_{11}+a_{12}+a_{21}+a_{22})\alpha$, $(a_{12}+a_{21})\beta$, $(a_{11}+a_{22})\gamma$

(d) $(a_{21})\alpha$, $(a_{11}+a_{22})\beta$, $(a_{11}+a_{21}+a_{22})\beta$

(e) $(a_{21}+a_{22})\alpha$, $(a_{11}+a_{12}+a_{22})\beta$, $(a_{11}+a_{12}+a_{21})\gamma$

(f) $a_{12}\alpha$, $(a_{11}+a_{22})\beta$, $(a_{11}+a_{12}+a_{22})\beta\gamma$

(g) $(a_{12}+a_{22})\alpha$, $(a_{11}+a_{21}+a_{22})\beta$, $(a_{11}+a_{12}+a_{21})\gamma$

(h) $a_{22}\alpha, (a_{12}+a_{21})\beta$, $(a_{12}+a_{21}+a_{22})\gamma$

Lemma 12: Let A and X be 2x2 and 2xn matrices respectively whose elements are from ψ. Any algorithm for computing AX which has k multiplications of types $a_{11}\alpha$, $a_{12}\beta$, and $(a_{11}+a_{12})\gamma$ has at least 3n+k/2 multiplications.

Corollary 13: By transformations we have similar theorems for $a_{21}\alpha$, $a_{22}\beta$, $(a_{21}+a_{22})\gamma$ and for $(a_{11}+a_{21})\alpha$, $(a_{12}+a_{22})\beta$, $(a_{11}+a_{12}+a_{21}+a_{22})\gamma$.

Applying Theorems 3 and 6 to each of the above yields several new theorems. However, only one new theorem for each will be presented. The others are similar.

Lemma 14: (Fiduccia) Let A be an nxm matrix whose elements are from $\mathscr{R}$ and let x be an arbitrary vector. If A has p nondependent rows, then any algorithm computing Ax requires at least p multiplications.

Lemma 15: Let $\mathscr{C}$ be a field and F be the set of expressions

$$\left\{\sum_{j=1}^{m} c_{ij}a_jB_{ij} \,\middle|\, 1 \le i \le p;\ a_j \in \psi,\ c_{ij} \in \mathscr{C};\ B_{ij} = \sum_{k=1}^{n} d_{ijk}x_k,\ d_{ijk} \in \mathscr{C},\ x_k \in \psi\right\} \text{where}$$

$B_{1j} = \ldots = B_{pj}$, $1 \le j \le t$. Let B be the pxt matrix whose ijth element is $c_{ij}B_{ij}$.

If F can be computed with q multiplications and B has t nondependent columns, then F can be computed with q multiplications in which $a_1, \ldots, a_t$ appear in exactly one multiplication each and that multiplication has the form $(a_j + \ell_j)B_j$ where

$$\ell_j = \sum_{i=t+1}^{m} \ell_{ij} a_i,\ \ell_{ij} \in \mathscr{C},\ 1 \leq j \leq t.$$

<u>Lemma 16</u>: Let A and X be 2xn by nx2 matrices respectively whose elements are from ψ. If an algorithm for computing AX = Y has k multiplications that are used only in computing y_{11} or only in computing y_{12} and y_{21} or only in computing y_{11}, y_{12}, and y_{21}, then the algorithm requires 3n+k multiplications.

<u>Corollary 17</u>: By applying transformations in T we have similar theorems for

(a) y_{11} and y_{21}; y_{12}, y_{21} and y_{22}; y_{11}, y_{12} and y_{22}

(b) y_{11} and y_{12}; y_{12}, y_{21} and y_{22}; y_{11}, y_{21} and y_{22}

(c) y_{11}, y_{12}, y_{21} and y_{22}; y_{12} and y_{21}; y_{11} and y_{22}

(d) y_{21}; y_{11} and y_{22}; y_{11}, y_{21} and y_{22}

(e) y_{21} and y_{22}; y_{11}, y_{12} and y_{22}; y_{11}, y_{12} and y_{21}

(f) y_{12}; y_{11} and y_{22}; y_{11}, y_{12} and y_{22}

(g) y_{12} and y_{22}; y_{11}, y_{21} and y_{22}; y_{11}, y_{12} and y_{21}

(h) y_{22}; y_{12} and y_{21}; y_{12}, y_{21} and y_{22}

<u>Lemma 18</u>: Let A and X be 2xn by nx2 matrices whose elements are from ψ. Any algorithm for computing AX = Y which has k multiplications used only in computing y_{11}, y_{12} or both has at least 3n+k/2 multiplications.

<u>Corollary 19</u>: By transformations we also have theorems for

(a) y_{21}; y_{22}; y_{21}; and y_{22};

(b) y_{11} and y_{21}; y_{12} and y_{22}; y_{11}, y_{12},y_{21} and y_{22}.

Instead of using Theorems 3 and 6 to prove the above we can construct "dual" proofs. As an example, we will present a proof for Lemma 18, the dual of Lemma 12.

<u>Proof of Lemma 18</u>:

We first state some results without proofs.

(1) Let $a_1, \ldots, a_p$ be n-vectors whose elements are of the form $\sum_{i=1}^{n} c_i x_i, c_i \in \mathscr{C}$ and $x_i \in \psi$. Then $a_1, \ldots, a_p$ are nondependent if and only if they are linearly independent.

(2) Let C and D be mxn matrices whose elements are from $\mathscr{R}$. If C and D have k_1 and k_2 nondependent columns respectively then C+D has at most k_1+k_2 nondependent columns.

(3) Let C be a 2xn matrix whose elements are from $\mathscr{R}$. If C has k nondependent columns, then row 1 or row 2 of C has at least k/2 nondependent elements.

Let $m_1,\ldots,m_k$ be the k multiplications which are assumed to be used only in computing y_{11}, y_{12} or both. Then $y_{11} = M_1 + F_1$ and $y_{12} = M_2 + F_2$ where M_1 and M_2 are sums of the $m_1,\ldots,m_k$ and $F_1 = y_{11} - M_1$ and $F_2 = y_{12} - M_2$. Without loss of generality we can assume

$$\begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = Gx \quad \text{and} \quad \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = Hx \quad \text{where } x = [x_{11},x_{12},\ldots,x_{n1},x_{n2}]^T$$

and G and H are 2x2n matrices whose elements are of the form $\sum_{i=1}^{2} \sum_{j=1}^{n} c_{ij}a_{ij}$, $c_{ij} \in \mathscr{C}$. Let G' and H' be the matrices resulting when we set $a_{21} = \ldots = a_{2n} = 0$. G' has at most k nondependent columns by Lemma 8. Since $\begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} = \left(G' + H' \right) x$, G' + H' must have 2n nondependent columns. Hence by (2) H' has at least 2n-k nondependent columns and by (3) row i, i is 1 or 2, has at least n-k/2 nondependent elements. Therefore, by (1) H has n-k/2 elements in row i of the form

$\sum_{j=1}^{n} c_j a_{1j} + \sum_{k=1}^{n} d_k a_{2k}$, c_j, $d_k \in \mathscr{C}$, such that the $\sum_{j=1}^{n} c_j a_{1j}$ parts of each element are linearly independent.

Assume $\mathscr{C}$ is a field. We can remove n-k/2 multiplications from Q by

(1) removing $m_1,\ldots,m_k$

(2) equating an appropriate choice of n-k/2 elements in row i of H to zero and solving for n-k/2 a_{1j}'s.

The new algorithm computes y_{21} and y_{22} which requires 2n multiplications. Hence Q must have had at least 3n+k/2 multiplications. If $\mathscr{C}$ is not a field 3n+k/2 is still a lower bound.

Theorems 3 and 6 and the preceding lemmas lead to the following lemmas for 2x3 by 3xn matrix multiplications (and hence 2xn by nx3, 3x2 by 2xn, 3xn by nx2, nx2 by 2x3, and nx3 by 3x2 matrix multiplications).

Lemma 20: Let A and X be 2x3 and 3xn matrices respectively whose elements are from ψ. Any algorithm for computing AX which has k multiplications of forms $a_{11}\alpha$, $a_{12}\beta$, $(a_{11}+a_{12})\gamma$, $a_{13}\delta$, $(a_{11}+a_{13})\xi$, $(a_{12}+a_{13})\theta$, and $(a_{11}+a_{12}+a_{13})\phi$ has at least 4n+2k/3 multiplications.

Proof: Similar to Lemma 12.

Corollary 21: Extend the definition of T in Corollary 11 in the obvious way to 2x3 by 3xn matrix multiplication. Then by transformations in T we have similar results for

(a) $a_{21}\alpha$, $a_{22}\beta$, $(a_{21}+a_{22})\gamma$, $a_{23}\delta$, $(a_{21}+a_{23})\xi$, $(a_{22}+a_{23})\theta$, $(a_{21}+a_{22}+a_{23})\phi$

(b) $(a_{11}+a_{21})\alpha$, $(a_{12}+a_{22})\beta$, $(a_{11}+a_{12}+a_{21}+a_{22})\gamma$, $(a_{13}+a_{23})\delta$, $(a_{11}+a_{13}+a_{21}+a_{23})\xi$, $(a_{11}+a_{13}+a_{22}+a_{23})\theta$, $(a_{11}+a_{12}+a_{13}+a_{21}+a_{22}+a_{23})\phi$.

Corollary 22: If n=3, and Q is an optimal algorithm for computing AX, then $k \leq 4$.

Corollary 23: Let n=3 and let Q be an optimal algorithm for computing AX. Let S_A be the set of all multiplications in Q that begin with

$$a_{11},\ a_{12},\ a_{11+12},\ a_{13},\ a_{11}+a_{13},\ a_{12}+a_{13},\ a_{11}+a_{12}+a_{13},\ a_{21},\ a_{22},\ a_{21}+a_{22},\ a_{23},$$
$$a_{21}+a_{23},\ a_{22}+a_{23},\ a_{21}+a_{22}+a_{23},\ a_{11}+a_{21},\ a_{12}+a_{22},\ a_{11}+a_{12}+a_{21}+a_{22},\ a_{13}+a_{23},$$
$$a_{11}+a_{13}+a_{21}+a_{23},\ a_{12}+a_{13}+a_{22}+a_{23},\ a_{11}+a_{12}+a_{13}+a_{21}+a_{22}+a_{23}.$$

Then

(i) at most 12 multiplications of Q are in S_A.

(ii) no two multiplications of Q that are in S_A have the same left hand side.

Proof

(i) Follows from corollary 22.

(ii) Suppose some multiplication of Q is in S_A. By applying transformations from T we can assume without loss of generality that multiplication has the form $a_{11}\alpha$. Set $a_{11} = 0$. This removes at least one multiplication from Q. Q now computes

$$\text{(I)}\qquad \begin{bmatrix} 0 & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$$

By Lemma 15 we can assume that setting $a_{21} = 0$ causes three multiplications to disappear. The resulting computation is a 2x2 by 2x3 matrix multiplication which requires 11 multiplications. Thus if setting $a_{11} = 0$ removed more than one multiplication Q must originally have had 16 multiplications and hence was not optimal. Therefore Q had only one multiplication of form $a_{11}\alpha$.

For the remainder of this discussion on matrix multiplication let $\mathscr{C} = Z_2$, the integers modulo 2. We will conclude this section by showing that Strassen's algorithm for 2x2 by 2x2 matrix multiplication is unique to within a transformation of T (as defined in Corollary 11). That is, every optimal algorithm for 2x2 by 2x2 matrix multiplication can be obtained from any given optimal algorithm for 2x2 by 2x2 matrix multiplication by applying a transformation of T to the latter. Let A and X be 2x2 matrices whose elements are from ψ. Let $a = [a_{11},a_{12},a_{21},a_{22}]^T$ and $x = [x_{11},x_{12},x_{21},x_{22}]^T$. Let M, P, R be 4x7, 7x4, and 7x4 matrices respectively whose elements are from $\mathscr{C}$ such that $M(Pa\cdot Rx)$ computes AX in row order. $M(Pa.Rx)$ uses 7 multiplications and hence is an optimal algorithm.

Lemma 24: For fixed P and R, M is unique.

Proof: Assume $M(Pa\cdot Rx) = M'(Pa\cdot Rx)$ where M' is a 4x7 matrix whose elements are from $\mathscr{C}$ and $M \neq M'$. Then there exists an equation $m_1 + \ldots + m_k = 0$, $k \geq 1$, where m_i is an entry of the column vector $Pa\cdot Rx$. Thus m_1 can be replaced by $(m_2 + m_3 + \ldots + m_k)$, implying that AX can be computed with 6 multiplications. In [3] it is shown that 7 multiplications are required. Therefore, M is unique.

Theorem 25: Any optimal algorithm Q for 2x2 by 2x2 matrix multiplication is unique to within a transformation of T.

Proof: Divide the multiplications of Q into two disjoint sets S_A and S_B, where the multiplications in S_A have left hand sides which can be mapped onto a_{11} by a transformation in T and the multiplications in S_B have left hand sides which can be mapped onto $a_{11} + a_{22}$ by a transformation in T. In [3] it is shown that an optimal algorithm must have six multiplications from S_A and one from S_B. Since any element of S_B can be mapped to any other element of S_B by a transformation in T we can assume without loss of generality that Q has a multiplication of the form $(a_{11}+a_{22})\alpha$. Lemmas 10 and 12 tell us that the remaining multiplications have forms

$$(a_{12}+a_{22})\alpha_2,\ (a_{11}+a_{21})\alpha_3,\ a_{22}\alpha_4,\ (a_{11}+a_{12})\alpha_5,\ (a_{21}+a_{22})\alpha_6,\ a_{11}\alpha_7.$$

Since the transformations of T preserve AX, any transformation that sends $a_{11}+a_{22}$ into itself will send the set of remaining left hand sides into itself. Thus we can assume without loss of generality that

$$Pa = \begin{bmatrix} (a_{11}+a_{22}) \\ (a_{12}+a_{22}) \\ (a_{11}+a_{21}) \\ a_{22} \\ (a_{11}+a_{12}) \\ (a_{21}+a_{22}) \\ a_{11} \end{bmatrix}$$

By the same reasoning and using duals of Lemmas 10 and 12, we can conclude that the right hand sides of the multiplications of Q must be a transformation of

$$\{x_{11}+x_{22},\ x_{21}+x_{22},\ x_{11}+x_{12},\ x_{22},\ x_{11}+x_{21},\ x_{12}+x_{22},\ x_{11}\}.$$

Since for any two sets of possible right hand sides there exists a transformation in T that sends one to the other without changing the set of left hand sides corresponding to the former, we can assume without loss of generality that

$$WRx = \begin{bmatrix} (x_{11}+x_{22}) \\ (x_{21}+x_{22}) \\ (x_{11}+x_{12}) \\ x_{22} \\ (x_{11}+x_{21}) \\ (x_{12}+x_{22}) \\ x_{11} \end{bmatrix} \qquad \text{where W is a 7x7 permutation matrix}$$

We need only show that R is unique.

Somehow we must form the product $a_{12}x_{21}$. Hence one of the four multiplications $(a_{12}+a_{22})(x_{21}+x_{22})$, $(a_{12}+a_{22})(x_{11}+x_{21})$, $(a_{11}+a_{12})(x_{21}+x_{22})$, and $(a_{11}+a_{12})(x_{11}+x_{21})$ must be present. Assume $(a_{12}+a_{22})(x_{11}+x_{21})$ is in Q. Then $(a_{11}+a_{12})(x_{11}+x_{22})$ must also be in Q since this is the only way to cancel the product $a_{12}x_{11}$ from $(a_{12}+a_{22})(x_{11}+x_{21})$ and to introduce the term $a_{12}x_{22}$. However, we cannot obtain $a_{12}x_{21}$ and $a_{12}x_{22}$ in separate expressions. Thus $(a_{12}+a_{22})(x_{11}+x_{21})$ is not in Q. Similar arguments eliminate

$(a_{11}+a_{12})(x_{21}+x_{22})$ and $(a_{11}+a_{12})(x_{11}+x_{21})$, leaving $(a_{12}+a_{22})(x_{21}+x_{22})$.

Considering products involving a_{12}, a_{21}, x_{12}, x_{21} we find that $(a_{12}+a_{22})(x_{21}+x_{22})$, $(a_{11}+a_{21})(x_{11}+x_{12})$, $a_{22}(x_{11}+x_{21})$, $(a_{11}+a_{12})x_{22}$, $(a_{21}+a_{22})x_{11}$, $a_{11}(x_{12}+x_{22})$ are in Q. This leaves $(a_{11}+a_{22})$ to match with $(x_{11}+x_{22})$.

Since the left and right hand sides can match up only one way R is unique. Thus by Lemma 24 M is unique and thus, the algorithm is unique to within a transformation of T.

General Expressions

Let $a_1,\dots,a_m$, $x_1,\dots,x_n$, d, be in $\mathscr{C}$ and let $c_{11},c_{12},\dots,c_{mn}$ be in ψ. Let $a = [a_1,\dots,a_m]^T$, $x = [x_1,\dots,x_m]^T$ and $d = [d]$. In this section we develop an effective procedure which will yield an optimal algorithm for computing a single expression

$$\sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}a_ix_j.$$

Vari [6] accomplishes the above provided that $\sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}a_ix_i = 0$ if and only if $a_i = x_j = 0$ for all i,j. Vari has subsequently removed this condition. Using Theorem 3, we give a second proof.

Theorem 26: There exists an effective procedure which yields an optimal algorithm for computing the expression

$$\sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}a_ix_j.$$

Proof: Theorem 3 tells us that an optimal algorithm for computing $\sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}a_ix_j$ is the P-dual of an optimal algorithm for computing the set of expressions

$$S = \{ \sum_{j=1}^{n} c_{ij}dx_j \mid i=1,\dots,m\}.$$

The minimum number of multiplications needed to compute this set of expressions equals the maximum number of nondependent expressions in the set $\{\sum_{j=1}^{n} c_{ij}x_j \mid i = 1,\dots,m\}$. Clearly, then we can find matrices M, P, R of appropriate dimensions such that $M(Pd\cdot Rx)$ computes S with the minimum number of multiplications. Then $P^T(M^Ta\cdot Rx)$ computes

$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}a_ix_j$ with the minimum number of multiplications.

References

1. Fiduccia, C. M., "Fast Matrix Multiplication", Proceedings of Third Annual ACM Symposium on Theory of Computing, pp. 45-49, 1971, Shaker Heights, Ohio

2. Gastinel, N., "Sur le calcul des produits de matrices", Numer. Math., 17, pp. 222-229, 1971

3. Hopcroft, J. E. and Kerr, L. R., "On minimizing the number of multiplications necessary for matrix multiplication", SIAM J. Appl. Math., Vol. 20, No. 1, pp. 30-35, 1971

4. Ostrowski, A. M., "On two problems in abstract algebra connected with Horner's rule", in Studies in Mathematics and Mechanics, pp. 40-48, Academic Press, New York

5. Strassen, N., "Gaussian elimination is not optimal", Numer. Math., 13, pp. 354-356, 1969

6. Vari, T. M., "On the number of multiplications required to compute quadratic functions", TR 72-120, Computer Science Department, Cornell University, Ithaca, New York, 1972

7. Winograd, S., "On the number of multiplications required to compute certain functions", Proc. Nat. Acad. Sci. USA, 58, pp. 1840-1842, 1967