ON THE 2 × 2 MATRIX MULTIPLICATION
Nader H. Bshouty
Department of Computer Science
University of Calgary
Calgary, Alberta
CANADA

Abstract

In [SIAM J. Comput, June 1976] Probert proved that 15 additive operations are necessary and sufficient to multiply two 2 × 2 matrices over the binary field by a bilinear algorithm using 7 non-scalar multiplications. We prove this result for arbitrary field. We also use the algorithm of Winograd to classify all such algorithms.

1 Introduction

Let \( F \) be a field, \( x = (x_1, \ldots, x_n)^T \), \( y = (y_1, \ldots, y_m)^T \) be column vectors of indeterminates and \( G = \{G_1, \ldots, G_t\} \) be a set of \( n \times m \) matrices with entries from \( F \). A bilinear algorithm that computes the bilinear forms \( z^T G y = \{z^T G_1 y, \ldots, z^T G_t y\} \) with multiplicative complexity \( \mu \) is a triple of matrices \((A, B, C)\) such that

\[
\begin{pmatrix}
(z^T G_1 y) \\
\vdots \\
(z^T G_t y)
\end{pmatrix} = A^T (Bz \ast Cy)
\]  

where \( A, B \) and \( C \) are \( t \times \mu, \mu \times n \) and \( \mu \times m \) matrices with entries from the field \( F \) and \( \ast \) is the component-wise product of vectors. In other words, we think of a computation of \( z^T G y \) according to a bilinear algorithm as consisting of four disjoint stages.

(i) Compute \( Bz = (m_1, \ldots, m_\mu)^T \).
(ii) Compute \( Cy = (m_1', \ldots, m_\mu')^T \).
(iii) Compute \( M = Bz \ast Cy = (m_1 m_1', \ldots, m_\mu m_\mu')^T \).
(iv) Compute \( z^T G_1 y, \ldots, z^T G_t y = A^T M \).

Therefore the multiplicative complexity of the bilinear algorithm is \( \mu \) and the additive complexity is of additions needed in steps (i), (ii) and (iv). The minimal number of nonscalar multiplications \( \mu \) needed to compute \( z^T G y \) is denoted by \( \mu(G) \) or \( \mu(z^T G y) \) and the minimal number of additive operations needed to compute \( Bz \) is denoted by \( \delta(B) \). Therefore the additive complexity of the bilinear algorithm in (1) is \( \delta(A^T) + \delta(B) + \delta(C) \).

In this paper we prove the following

Theorem 1. 15 additive operations are necessary and sufficient to multiply two \( 2 \times 2 \) matrices by a bilinear algorithm using 7 nonscalar multiplications.

Theorem 2. Let

\[
X = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{pmatrix}, \quad Y = \begin{pmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{pmatrix}.
\]

Denote by \( \text{vec}(X) = (x_{1,1}, x_{2,1}, x_{1,2}, x_{2,2})^T \). Then

\[
\text{vec}(XY) = M^T (N \text{vec}(X) \ast K \text{vec}(Y)),
\]

is a bilinear algorithm that computes \( XY \) with multiplicative complexity 7 and additive complexity 15 if and only if

\[
(M, N, K) = \Pi(I_7^p A(G^T \otimes J^T), \ I_7^p B(H^T \otimes G^{-1}), \ I_7^p C(J^{-1} \otimes H^{-1}))
\]

where

1) \( \Pi \in \{\Pi_0, \Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5\} \) where for \( (D, E, F) \) we have

\[
\Pi_0(D, E, F) = (D, E, F), \quad \Pi_1(D, E, F) = (DW, FW, EW),
\]
\[ \Pi_2(D, E, F) = (E, D, FW) \]
\[ \Pi_3 = \Pi_1 \Pi_2, \Pi_4 = \Pi_2 \Pi_1, \Pi_5 = \Pi_1 \Pi_2 \Pi_1 \]
and
\[ W = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}. \]

2) \( v_1, v_2, v_3 \in (F^*)^7, F^* = F \setminus \{0\} \), for \( v = (\alpha_1, \ldots, \alpha_7) \in F^7 \) and permutation \( \phi \) on \{1, \ldots, 7\} the 7 \( \times \) 7 matrix \( J_{\phi}^\alpha \) is defined as
\[ J_{\phi}^\alpha[i,j] = \begin{cases} 
\alpha_i & j = \phi(i) \\
0 & \text{otherwise}
\end{cases}, \]

3)
\[ (G, J, H) = (I_{a_1,b_1,1}, I_{a_2,b_2,1}, I_{a_3,b_3,1}) \]
where
\[ U_1, U_2, U_3 \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \]
and
\[ (G, J, H) \in \{(I, I, I), (I, S_2, I), (S_1, S_1, I), (S_1, S_2, S_2), (S_2, I, S_2), (S_2, S_1, S_1)\} \]
\[ I = \begin{pmatrix} 1 & 1 \\
1 & 1 \end{pmatrix}, S_1 = \begin{pmatrix} -1 & 1 \\
1 & -1 \end{pmatrix}, S_2 = \begin{pmatrix} 1 & 1 \\
1 & -1 \end{pmatrix}. \]

And

4) \((A, B, C)\) is the bilinear algorithm of Winograd (See Section 2).

In Section 2 we prove Theorem 1 and in Section 3 we prove Theorem 2.

2. \( 2 \times 2 \) MATRICES

Let \( F \) be a field and
\[ X = \begin{pmatrix} x_{1,1} & x_{1,2} \\
x_{2,1} & x_{2,2} \end{pmatrix}, \quad Y = \begin{pmatrix} y_{1,1} & y_{1,2} \\
y_{2,1} & y_{2,2} \end{pmatrix}. \]
Denote by \( \text{vec}(X) = (x_{1,1}, x_{2,1}, x_{1,2}, x_{2,2})^T \). Then a bilinear algorithm that computes \( XY \) with 7 non-scalar multiplications is \((A, B, C)\) where:
\[ \text{vec}(XY) = A^T (B \text{vec}(X) \ast C \text{vec}(Y)), \quad (2) \]
\[ A, B \text{ and } C \text{ are } 7 \times 4 \text{ matrices}. \]

It is known that if \((A, B, C)\) is an algorithm for \( \text{vec}(XY) \) then are the dual algorithms [BM]
\[ (B, A, CW) \text{ and } (C, BW, A) \quad (3) \]
where
\[ W = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}. \]

If we prove the lower bounds \( \delta(A^T) \geq 4, \delta(B^T) \geq 4 \) and \( \delta(C^T) \geq 4 \). Then by the results in [KKB] we have \( \delta(B) = \delta(B^T) + 7 - 4 = \delta(B^T) + 3 \geq 7 \) and \( \delta(C) \geq 7 \) and therefore
\[ \delta(A^T) + \delta(B) + \delta(C) \geq 15. \quad (5) \]
This implies

\[ \text{Lemma 1}. \text{ If for every bilinear algorithm } (A, B, C) \text{ for } \text{vec}(XY) \text{ with multiplicative complexity } 7 \text{ we have } \delta(A) \geq 4, \delta(B) \geq 4 \text{ and } \delta(C) \geq 4 \text{ then the additive complexity of bilinear algorithms for } \text{vec}(XY) \text{ with multiplicative complexity 7 is } \delta(A^T) + \delta(B) + \delta(C) \geq 15. \quad \Box \]

We now give the algorithm of Winograd which computes the product of \( 2 \times 2 \) matrices with 7 non-scalar multiplications and 15 additive operations:
\[ \text{vec}(XY) = A^T \left( B \text{vec}(X) \ast C \text{vec}(Y) \right) \defeq \\
\left( \begin{array}{cccc}
0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0
\end{array} \right) \times \\
\left( \begin{array}{ccc}
-1 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1 \\
1 & -1 & 0 \\
0 & 1 & 0 \\
1 & -1 & 1 & -1 \\
0 & 0 & 0 & 1
\end{array} \right) \left( \begin{array}{ccc}
x_{1,1} \\
x_{2,1} \\
x_{1,2} \\
x_{2,2}
\end{array} \right) \times \\
\left( \begin{array}{ccc}
1 & 0 & -1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 0 & -1 & 1 \\
-1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array} \right) \left( \begin{array}{ccc}
y_{1,1} \\
y_{2,1} \\
y_{1,2} \\
y_{2,2}
\end{array} \right) \quad (6) \]
Now $M = B \mathrm{vec}(X)$ can be computed in 4 additive operations by

\[ l_1 = x_{1,1} - x_{2,1}; \quad l_2 = x_{2,2} - l_1; \]
\[ l_3 = x_{1,2} - l_2; \quad l_4 = x_{2,1} + x_{2,2} \]
\[ m_1 = l_2; \quad m_2 = x_{1,1}; \quad m_3 = x_{1,2}; \quad m_4 = l_1; \]
\[ m_5 = l_4; \quad m_6 = l_3; \quad m_7 = x_{2,2} \]

$M' = C \mathrm{vec}(Y)$ can be computed in 4 additive operations by

\[ l'_1 = y_{2,2} - y_{1,2}; \quad l'_2 = y_{1,1} - l'_1; \]
\[ l'_3 = y_{2,1} - l'_2; \quad l'_4 = y_{1,2} - y_{1,1} \]
\[ m'_1 = l'_2; \quad m'_2 = y_{1,1}; \quad m'_3 = y_{1,2}; \quad m'_4 = l'_1; \]
\[ m'_5 = l'_4; \quad m'_6 = y_{2,2}; \quad m'_7 = l'_3. \]

Define

\[ (s_1, s_2, \ldots, s_7) \overset{\text{def}}{=} M \ast M' = (m_1m'_1, \ldots, m_7m'_7). \]

Then $A(M \ast M')$ can be computed in 7 additions by

\[ z_{1,1} = x_{1,1}y_{1,1} + x_{1,2}y_{2,1} = s_2 + s_3 \]
\[ s_8 = s_1 + s_2 \]
\[ s_9 = s_3 + s_8 \]
\[ z_{2,1} = x_{2,1}y_{1,1} + x_{2,2}y_{2,1} = s_7 + s_9 \]
\[ z_{1,2} = x_{1,1}y_{1,2} + x_{1,2}y_{2,2} = s_5 + s_8 + s_8 \]
\[ z_{2,2} = x_{2,1}y_{1,2} + x_{2,2}y_{2,2} = s_5 + s_9 \]

this follows

\[ \text{Lemma 2.} \quad \text{The algorithm } (A, B, C) \text{ of Winograd for } \mathrm{vec}(XY) \text{ is of multiplicative complexity 7 and additive complexity 15.} \]

Winograd also proved [W] that the multiplicative complexity of $\mathrm{vec}(XY)$ cannot be less than 7.

We write

\[ (A, B, C) \overset{\text{def}}{=} (A', B', C'), \]

equivalent bilinear algorithms if $(A', B', C')$ can be obtained from $(A, B, C)$ by multiplying each of the $i$-th rows of $A$, $B$ and $C$ by $a_i$, $b_i$ and $c_i$, respectively, where $a_i b_i c_i = 1$, $i = 1, \ldots, \mu$ and by permuting the rows of $A$, $B$ and $C$ with the same permutation. Obviously, equivalent bilinear algorithms compute the same bilinear forms and has the same multiplicative complexity and additive complexity.

To prove Theorem 1 we begin with the following

\[ \text{Lemma 3.} \quad \text{If } (A, B, C) \text{ is a bilinear algorithm for } \mathrm{vec}(XY) \text{ with multiplicative complexity 7 then the rows of } A, B \text{ and } C \text{ are pairwise independent.} \]

\[ \text{Proof.} \quad \text{By (3) it is enough to prove that for every bilinear algorithm } (A, B, C) \text{ for } \mathrm{vec}(XY) \text{ the rows of } B \text{ are pairwise independent. Assume that (w.l.o.g)} \]

\[ \begin{pmatrix} v_1 \\ \beta v_1 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix} \overset{\text{def}}{=} B \]

\[ v_1 \mathrm{vec}(X) = a_1 x_{1,1} + a_2 x_{2,1} + a_3 x_{1,2} + a_4 x_{2,2} \]

and $a_1 \neq 0$. Substituting $x_{1,1} = -(a_2 x_{2,1} + a_3 x_{1,2} + a_4 x_{2,2})/a_1$ in the algorithm we obtain a new algorithm that computes for $u = -(a_2/a_1)x_{2,1} - (a_3/a_1)x_{1,2} - (a_4/a_1)x_{2,2}$ the multiplication

\[ \begin{pmatrix} \beta u \\ x_{1,2} \\ x_{2,1} \\ x_{2,2} \end{pmatrix} \overset{\text{def}}{=} B' \]

with 5 nonscalar multiplications. Since by [FW, Lemma] the multiplicative complexity of (7) is 6, we have a contradiction.

Let

\[ \mathrm{vec}(XY) = A^T (B \mathrm{vec}(X) \ast C \mathrm{vec}(Y)). \]

Since $XY = G(G^{-1}XH)(H^{-1}Y(J^T)^{-1})J^T$ for nonsingular $2 \times 2$ matrices $G$, $H$ and $J$ and since $\mathrm{vec}(RSL) = (L^T \otimes R) \mathrm{vec}(S)$ we have

\[ \mathrm{vec}(XY) = (J \otimes G) \mathrm{vec}(G^{-1}XH)(H^{-1}Y(J^T)^{-1}) \]

By (8) we have

\[ \mathrm{vec}(G^{-1}XH)(H^{-1}Y(J^T)^{-1}) = A^T (B \mathrm{vec}(G^{-1}XH) \ast C \mathrm{vec}(H^{-1}Y(J^T)^{-1})) \]

\[ = A^T (B(H^T \otimes G^{-1}) \mathrm{vec}(X) \ast C(J^{-1} \otimes H^{-1}) \mathrm{vec}(Y)) \]

and with (8) we obtain

\[ \mathrm{vec}(XY) = ((J \otimes G)A^T) (B(H^T \otimes G^{-1}) \mathrm{vec}(X) \ast C(J^{-1} \otimes H^{-1}) \mathrm{vec}(Y)). \]

Since $XY = (Y^T X^T)^T$ we have

\[ \mathrm{vec}(XY) = \mathrm{vec}(Y^T X^T)^T = W \mathrm{vec}(Y^T X^T)^T = \]

\[ \mathrm{vec}(XY) = \mathrm{vec}(Y^T X^T)^T = W \mathrm{vec}(Y^T X^T)^T = \]
\[(AW)^T (CW \text{ vec}(X) \ast BW \text{ vec}(Y))\]
where \(W\) is the permutation matrix in (4). This with (3) and (10) follows

**Lemma 4.** If \((A, B, C)\) is a bilinear algorithm for \(\text{vec}(XY)\) then \((AW, CW, BW)\), \((B, A, CW)\) and \((A(J^T \otimes G^T), B(H^T \otimes G^{-1}), C(J^{-1} \otimes H^{-1}))\) are algorithms for \(\text{vec}(XY)\) with the same multiplicative complexity where \(W\) is the matrix in (4) and \(J, G\) and \(H\) are any nonsingular \(2 \times 2\) matrices.

Let
\[
I(A, B, C) = (A, B, C),
\]
\[
\Pi_1(A, B, C) = (AW, CW, BW),
\]
\[
\Pi_2(A, B, C) = (B, A, CW),
\]
\[
\Lambda_{J,G,H}(A, B, C) = (A(J^T \otimes G^T), B(H^T \otimes G^{-1}), C(J^{-1} \otimes H^{-1})).
\]

In [Gr2] de Groote proved that every bilinear algorithm \((M', N', K')\) for \(\text{vec}(XY)\) is equivalent to an algorithm that can be obtained from a sequence of the operations \(\Pi_1, \Pi_2, \Lambda_{J,G,H}\) on \((A, B, C)\) that defined in (6). Now to simplify the result of de Groote we give the following

**Lemma 5.** Every bilinear algorithm \((M', N', K')\) for \(\text{vec}(XY)\) is equivalent to a bilinear algorithm \((M, N, K)\) such that
\[
(M, N, K) = \Pi \Lambda_{J,G,H}(A, B, C)
\]
for some \(\Pi \in \{\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5, \Pi_6, \Pi_7, \Pi_8\}\) where \((A, B, C)\) is the algorithm of Winograd.

**Proof.** It can be easily verify that \(W^2 = I\) and \((J \otimes G) W = W (G \otimes J)\) and therefore
\[
\Pi_1^2 = I, \quad \Pi_2^2 = I,
\]
\[
\Lambda_{J_1, G_1, H_1} \Lambda_{J_2, G_2, H_2} = \Lambda_{J_1, J_2, G_1 G_2, H_1 H_2},
\]
\[
\Lambda_{J, G, H} \Pi_1 = \Pi_1 \Lambda_{J, G, H (H^{-1})^T},
\]
\[
\Lambda_{J, G, H} \Pi_2 = \Pi_2 \Lambda_{H (G^{-1})^T, J},
\]
\[
\Pi_1 \Pi_2 \Pi_1 = \Pi_2 \Pi_1 \Pi_2.
\]

With the result of de Groote the result follows.

Since \(\{\Pi_1, \Pi_2, \Pi_3, \Pi_4, \Pi_5, \Pi_6, \Pi_7, \Pi_8\}\) are operations that only permute the columns of \(A, B\) and \(C\) we have that the bilinear algorithms \((A, B, C)\) and \(\Pi (A, B, C)\) have the same additive complexity.

Therefore to prove Theorem 1, by lemma 1 and lemma 5, it is sufficient to show that for every nonsingular matrices \(G, J\) and \(H\) we have
\[
\delta(A(G^T \otimes J^T)) \geq 4, \quad \delta(B(H^T \otimes G^{-1})) \geq 4,
\]
\[
\delta(C(J^{-1} \otimes H^{-1})) \geq 4
\]
or
\[
\delta(K(G^T \otimes J^T)) \geq 4
\]
for every \(K \in \{A, B, C\}\).

For a permutation \(\phi\) we define the matrix
\[
I^{(\alpha_1, \ldots, \alpha_8)}_{\phi}
\]
as follows:
\[
I^{(\alpha_1, \ldots, \alpha_8)}_{\phi} = \begin{cases} \alpha_i & j = \phi(i); \\ 0 & \text{otherwise} \end{cases}
\]
Then it can be easily shown that
\[
B = I^{(-1,1,1,1,-1,1,1,-1)}_{\phi} \in A(G_1 \otimes J_1),
\]
\[
C = I^{(-1,1,1,1,-1,1,1,-1)}_{\phi} \in A(J_2 \otimes J_2)
\]
for every \(K \in \{A, B, C\}\).

Since the addition complexity does not change when we multiply the matrix from the left handside by \(I^{(\alpha_1, \ldots, \alpha_8)}_{\phi}\), we obtain that if for every nonsingular matrix \(G\) and \(J\) we have \(\delta(A(G^T \otimes J^T)) \geq 4\) then: for every nonsingular matrices \(G\) and \(J\) we have
\[
4 \leq \delta(A(G^T \otimes J^T)) = \delta(I^{(-1,1,1,1,-1,1,1,-1)}_{\phi} \in A(G_1 \otimes J_1)(G^T \otimes J^T)) = \delta(B(G^T \otimes J^T))
\]
and so for \(C\).

We proved

**Lemma 6.** If for every nonsingular \(2 \times 2\) matrices \(G\) and \(J\) and for the matrix
\[
\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{pmatrix}
\]
\[
= A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}
\]
(12)
we have \( \delta(A(G^T \otimes J^T)) \geq 4 \) then every bilinear algorithm for \( \text{vec}(XY) \) with multiplicative complexity 7 has additive complexity 15. \( \square \)

Denote by \( \{e_1, e_2, e_3, e_4\} \) the rows of the \( 4 \times 4 \) identity matrices.

We first prove

**Lemma 7.** The matrix \( H = A(G^T \otimes J^T) \) cannot have 4 rows from \( \{a_1 e_1, a_2 e_2, a_3 e_3, a_4 e_4 \mid a_i \in F\} \).

**Proof.** Since by lemma 3 the rows of \( H \) are pairwise independent it is enough to prove that \( H \) cannot have in its rows the vectors \( a_1 e_1, a_2 e_2, a_3 e_3 \) and \( a_4 e_4 \). This is equivalent to say that there exists no distinct rows \( \beta_1 v_{i_1}, \beta_2 v_{i_2}, \beta_3 v_{i_3}, \beta_4 v_{i_4} \) in \( A \) such that

\[
\begin{pmatrix}
\beta_1 v_{i_1} \\
\beta_2 v_{i_2} \\
\beta_3 v_{i_3} \\
\beta_4 v_{i_4}
\end{pmatrix} (G^T \otimes J^T) = I,
\]

which implies that

\[
V = \begin{pmatrix}
\beta_1 v_{i_1} \\
\beta_2 v_{i_2} \\
\beta_3 v_{i_3} \\
\beta_4 v_{i_4}
\end{pmatrix} = (G^T)^{-1} \otimes (J^T)^{-1} \otimes S \otimes R =
\begin{pmatrix}
s_{1,1} R & s_{1,2} R \\
s_{2,1} R & s_{2,2} R
\end{pmatrix}
\]

If \( s_{1,2} \neq 0 \) and \( s_{2,2} \neq 0 \) then since \( R \) is nonsingular \( \begin{pmatrix} s_{1,2} R \\ s_{2,2} R \end{pmatrix} \) has two nonzero pair of dependent rows. Observing the two last columns of \( A \) we conclude that this case cannot happen and therefore \( s_{1,2} = 0 \) or \( s_{2,2} = 0 \). In the same way we can prove that \( s_{1,1} = 0 \) or \( s_{2,1} = 0 \). Now if \( s_{1,2} = 0 \) then since \( S \) is nonsingular we have \( s_{1,1} \neq 0 \) and therefore \( s_{2,1} = 0 \) which implies that

\[
\begin{pmatrix}
\beta_1 v_{i_1} \\
\beta_2 v_{i_2} \\
\beta_3 v_{i_3} \\
\beta_4 v_{i_4}
\end{pmatrix} = \begin{pmatrix}s_{1,1} R & 0 \\
0 & s_{2,2} R
\end{pmatrix}.
\]

Observing the rows of \( A \) we must have \( \{i_1, i_2, i_3, i_4\} = \{3, 5, 6, 7\} \) and in this case the rows 3,5,6 and 7 cannot be of the form in (13). \( \square \)

Now we give the last lemma in this section which complete the proof of Theorem 1.

**Lemma 8.** We have \( \delta(A(G^T \otimes J^T)) \geq 4 \) for every nonsingular matrices \( G \) and \( J \).

**Proof.** By lemma 4, \( H = A(G^T \otimes J^T) \) cannot contain more than 3 rows from \( \{a_1 e_1, a_2 e_2, a_3 e_3, a_4 e_4 \} \) and by lemma 3 the other 4 rows are distinct. Therefore each one of the other 4 rows requires at least one additive operations which implies that \( \delta(H) \geq 4 \). \( \square \)

### 3. Classification

In this section we classify all the algorithms for \( \text{vec}(XY) \) with multiplicative complexity 7 and additive complexity 15. I.e we shall find

\[
\mathcal{R} = \{(M, N, K) \mid (M, N, K) \text{ computes vec}(XY) \}
\]

with \( \mu = 7 \) and \( \delta = 15 \).

By lemma 1 and 5 we have

\[
\mathcal{R} = \bigcup_{R \in \mathcal{R}} \mathcal{R}'
\]

where

\[
\mathcal{R}' = \{(M', N', K') \mid (M', N', K') \text{ is equivalent to } (M, N, K)\}
\]

\[
= \{(A(G^T \otimes J^T), B(H^T \otimes G^{-1}), C(J^{-1} \otimes H^{-1})) \}
\]

\[
\delta(M) = \delta(N) = \delta(L) = 4
\]

Then we reduce the algorithms to nonequivalent algorithms as follows:

\[
\mathcal{R}' = \bigcup_{(\alpha_1, \ldots, \alpha_7) \in F^*} \{(\beta_1, \ldots, \beta_7) \in F^*^7 \}
\]

\[
= \{(\gamma_1, \ldots, \gamma_7) \in F^*^7 \}
\]

\[
\phi \in S_7
\]

of

\[
\{(f^{(\alpha_1, \ldots, \alpha_7)}_M, f^{(\beta_1, \ldots, \beta_7)}_N, f^{(\gamma_1, \ldots, \gamma_7)}_K) \}
\]

\[
(M, N, K) \in \mathcal{R}
\]

where \( F^* = F \setminus \{0\} \), \( S_7 \) is the permutation group over \( \{1, \ldots, 7\} \) and

\[
\mathcal{R} = \{(M, N, K) \mid (A(G^T \otimes J^T), B(H^T \otimes G^{-1}), C(J^{-1} \otimes H^{-1})) \}
\]

where

\[
\delta(M) = \delta(N) = \delta(K) = 7
\]
Now we need the following lemma

**Lemma 9.** If \( \delta(A(R \otimes S)) = 4 \) then \( A(R \otimes S) \) has 3 distinct rows from \( M = \{ \alpha_1e_1, \alpha_2e_2, \alpha_3e_3, \alpha_4e_4 \} \).

**Proof.** As in the proof of lemma 8, if \( A \) has less than 4 rows from \( M \) then it has at least 5 pairwise independent rows not from \( M \) which implies that the additive complexity is at least 5. \( \square \)

Assume that \( A(R \otimes S) \) contains the rows \( \alpha_1e_1, \alpha_2e_2, \alpha_3e_4 \). Then there exist \( v_1, v_2, v_3 \) rows in \( A \) such that
\[
\begin{pmatrix}
v_1 \\
v_2 \\
v_3
\end{pmatrix}
(R \otimes S) = \begin{pmatrix}
\alpha_1e_1 \\
\alpha_2e_2 \\
\alpha_3e_4
\end{pmatrix}.
\]

Let \( I_{\beta, \gamma} = \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \) then
\[
\begin{pmatrix}
v_1 \\
v_2 \\
v_3
\end{pmatrix}
(RI_{\frac{1}{2}, \frac{1}{2}} \otimes SI_{\frac{1}{2}, \frac{1}{2}}) = \begin{pmatrix}
e_1 \\
e_2 \\
e_4
\end{pmatrix}.
\]

Notice that this transformation can be done also for \( \{ e_1, e_2, e_3 \} \) and \( \{ e_2, e_3, e_4 \} \).

Assume that \( A(R \otimes S) \) has \( e_1, e_2, e_3 \) in its rows. Then there exist \( u_1, u_2, u_3, u_4 \) such that three of them are distinct rows of \( A \) and
\[
\begin{pmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4
\end{pmatrix}
(R \otimes S) = I.
\]

This follows that
\[
\begin{pmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4
\end{pmatrix} = (R^{-1} \otimes S^{-1}) D \otimes E = \begin{pmatrix} d_{1,1}E & d_{1,2}E \\
d_{2,1}E & d_{2,2}E \end{pmatrix}.
\]

Since three of \( u_1, u_2, u_3, u_4 \) are rows of \( A \) we have: the rows of \( [d_{1,1}E][d_{2,2}E] \) or of \( [d_{2,1}E][d_{2,2}E] \) contained in the rows of \( A \). Observing the rows of \( A \) and since \( D \otimes E \) and \( E \) are nonsingular we have \( d_{1,1}E \) or \( d_{1,2}E \) must be one of the matrices in
\[
P = \{ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \}
\]

and therefore \( E \in \{ \lambda P | P \in P \} \) where \( \lambda \in F \). Now we must have \( d_{i,j} \in \{ 0, \lambda^{-1} \} \) and since \( D \) is nonsingular we have \( D \in \lambda^{-1} P \). Since \( D = R^{-1} \) and \( E = S^{-1} \), and with (17) we have

**Lemma 10.** If \( \delta(A(R \otimes S)) = 4 \) then
\[
R, S \in \bigcup_{\alpha, \beta \in F^*} Q_{I_{\alpha, \beta}}.
\]

where
\[
Q = \{(1, 1)(1, 1)(1, -1)(1, 1), (1, -1)(1, 1)\}.
\]

We now reduce the problem to finite number of cases by using lemma 10 in the following way: Since \( \delta(A(G^T \otimes J^T)) = 4 \) we have
\[
G, J \in \bigcup_{\alpha, \beta \in F^*} I_{\alpha, \beta} Q.
\]

Now by (11) we have for some \( v \in \{1, -1\} \) and \( \phi \in S^* \)
\[
4 = \delta(B(H^T \otimes G^{-1})) = \delta(I_{\phi} A(G_1 \otimes J_1)(H^T \otimes G^{-1})) =
\]
and by lemma 10 \( G_1H^T \in \bigcup_{\alpha, \beta \in F^*} Q I_{\alpha, \beta} \) which implies that
\[
H \in \bigcup_{\alpha, \beta \in F^*} I_{\alpha, \beta} Q.
\]

Now by (18) and (19) we have
\[
G = I_{\alpha_1, \beta_1} G', J = I_{\alpha_2, \beta_2} J', H = I_{\alpha_3, \beta_3} H'
\]
where \( G', J', H' \in Q \). Also since
\[
Q = S \cup \{(1, 1)S \}
\]

where
\[
S = \{I, S_1, S_2\} \equiv \{(1, 1)(-1, 1)(1, -1)\}
\]

we can write
\[
G = I_{\alpha_1, \beta_1} U_1 \overline{G}, J = I_{\alpha_2, \beta_2} U_2 \overline{J}, H = I_{\alpha_3, \beta_3} U_3 \overline{H}
\]
where \( U_1, U_2, U_3 \in \{I, (1, 1)\} \) and \( \overline{G}, \overline{J}, \overline{H} \in S \).
Now since
\[ A(G^T \otimes J^T) = A(G^T \otimes J^T)(U_1 I_{\alpha_1, \beta_1} \otimes U_2 I_{\alpha_2, \beta_2}) \]
and \((U_1 I_{\alpha_1, \beta_1} \otimes U_2 I_{\alpha_2, \beta_2})\) does not change the additive complexity of \(A(G^T \otimes J^T)\) we have
\[
\tilde{\mathcal{R}} = \bigcup_{U_1, U_2, U_3 \in \left\{ I, \left( \begin{array}{cc} 1 & 1 \\ 1 & 1 \end{array} \right) \right\}} (\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3) \in (F^*)^6
\]
of
\[
\lambda U_1 I_{\alpha_1, \beta_1} U_2 I_{\alpha_2, \beta_2} U_3 I_{\alpha_3, \beta_3} \tilde{\mathcal{R}} \quad (20).
\]
where
\[
\tilde{\mathcal{R}} = \{(M, N, K) = (A(G^T \otimes J^T), B(H^T \otimes G^{-1}), C(J^{-1} \otimes H^{-1})) | G, J, H \in S, \delta(M) = \delta(N) = \delta(K) = 4\}
\]
It can be easily verify that
\[
\tilde{\mathcal{R}} = \{(I, I, I), (I, S_1, I), (S_1, S_1, I), (S_1, S_2, S_2), (S_2, I, S_2), (S_2, S_1, S_1)\} \quad (21)
\]
and then by (14), (15), (16), (20) and (21), Theorem 2 follows.

Reference


