The Coppersmith-Winograd Matrix Multiplication Algorithm

Matthew Anderson, Siddharth Barman

December 6, 2009

1 Introduction

Definition 1 (Matrix Multiplication)**.** *Given matrices $A \in \mathbb{F}^{m \times n}, B \in \mathbb{F}^{n \times p}$, compute $C \in F^{m \times p}$ such that $AB = C$.*

We only consider square matrices of dimension n (so $m = n = p$), though all arguments can be extended in some way to distinct dimensions (and even give better results in some cases).

Definition 2 ($M_k(n)$)**.** *The number of multiplication operations over field k sufficient to multiply two n by n matrices.*

Definition 3 (ω)**.** $\omega(k) \equiv \inf\{\tau \in \mathbb{R} | M_k(n) = O(n^\tau)\}$. *Informally $\omega(k)$ is the minimum exponent required to multiply two n by n matrices.*

For simplicity we fix $k = \mathbb{F}$, for some arbitrary field $\mathbb{F}$, so we will drop it from the notation and for the most part ignore it.

Trivially, $2 \leq \omega \leq 3$. The upper bound follows from the grade school algorithm for matrix multiplication and the lower bound follows because the output is of size of C is n^2.

Some progress has been made, though Coppersmith-Winograd represents the pinnacle thus far:

Year	$\omega <$	
< 1969	3	
1969	2.81	Strassen
1978	2.79	Pan
1979	2.78	Bini et al
1981	2.55	Schonhage
1982	2.50	Pan; Romani; CW
1987	2.48	Strassen
1987	2.38	CW

Figure 1: Historical improvements in bounding ω

2 Intuition

Before we get into the details of the CW, we'll try to provide some intuition for what is going on.

2.1 Multiplying Complex Numbers

Let $A, B \in \mathbb{C} = \mathbb{R}[i]/(i^2+1)$. Wlog $A = (a+ib)$ and $B = (c+id)$, then $AB = (ac-bd)+(ad+bc)i$. This process uses 4 multiplications. Can we do better? Certainly, define:

$$\begin{aligned} m_1 &= (a+b)(c-d) = (ac-bd) + bc - ad \\ m_2 &= bc \\ m_3 &= ad \\ AB &= (m_1 - m_2 + m_3) + (m_2 + m_3)i \end{aligned} \tag{1}$$

Thus, complex multiplication be accomplished with only 3 multiplications over $\mathbb{R}$.

2.2 Strassen's Algorithm

Strassen's 1969 algorithm, which gives $\omega < 2.81$ follows similarly. (For reference see [Str69], [Wik09], [BCS97] pages 10-14 or almost any book on algebraic algorithms). Let $A, B, C \in \mathbb{R}^{2\times 2}$.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix} \tag{2}$$

$$\begin{aligned} \mathbf{M}_1 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2}) \\ \mathbf{M}_2 &:= (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1} \\ \mathbf{M}_3 &:= \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2}) \\ \mathbf{M}_4 &:= \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1}) \\ \mathbf{M}_5 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2} \\ \mathbf{M}_6 &:= (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2}) \\ \mathbf{M}_7 &:= (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2}) \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{C}_{1,1} &= \mathbf{M}_1 + \mathbf{M}_4 - \mathbf{M}_5 + \mathbf{M}_7 \\ \mathbf{C}_{1,2} &= \mathbf{M}_3 + \mathbf{M}_5 \\ \mathbf{C}_{2,1} &= \mathbf{M}_2 + \mathbf{M}_4 \\ \mathbf{C}_{2,2} &= \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_6 \end{aligned}$$

Instead of using the 8 multiplications of the trivial approach, Strassen's algorithm only uses 7. Applying a divide and conquer strategy recursively (view $A_{i,j}$, $B_{i,j}$ and $C_{i,j}$ as matrices instead of scalars) allows matrix multiplication over $n = 2^N$ size matrices to be performed using only $7^N = 7^{\log_2 n} = n^{\log_2 7} = O(n^{2.81})$ multiplications.

3 Background

We need to introduce a more complex formalism to discuss CW.

Definition 4 (bilinear map)**.** *Let U, V, W be $\mathbb{F}$-spaces. $\phi : U \times V \to W$, is a* bilinear map *for all $u_1, u_2 \in U, v_1, v_2 \in V$ and $\lambda_1, \lambda_2, \mu_1, \mu_2 \in \mathbb{F}$, $\phi(\lambda_1 u_1 + \lambda_2 u_2, \mu_1 v_1 + \mu_2 v_2) = \sum_{i,j} \lambda_i \mu_j \phi(u_i, v_j)$.*

In particular, we get a linear map by holding the first entry of the bilinear map fixed, while letting the second entry vary. Similarly if we hold the second entry fixed we again get a linear map. For matrix multiplication $U = V = W = \mathbb{F}^{n\times n}$, and we write this bilinear map as $\phi = \langle n, n, n\rangle$.

Definition 5 (rank)**.** *Let ϕ be a $\mathbb{F}$-bilinear map. Then the* rank *of ϕ, $R(\phi)$ is the smallest $r \in \mathbb{N}$ such that*

$$\phi(u, v) = \sum_i^r f_i(u)g_i(v)w_i \tag{4}$$

For $w_i \in W$ and f and g are $\mathbb{F}$-linear forms (a linear transformation from a vector space to its scalar field).

The following proposition shows that $R(\langle n, n, n\rangle)$ is related to $M(n)$:

Proposition 1 ([BCS97] Prop 15.1)**.**

$$\omega = \inf\{\tau \in \mathbb{R} | R(\langle n, n, n\rangle) = O(n^\tau)\}. \tag{5}$$

Proof. Let the RHS in the statement of the proposition be θ and $\phi = \langle n, n, n\rangle$. Then

$$\begin{aligned}
\phi(a, b) &= \sum_i^{|M(n)|} f_i(a, b)g_i(a, b)w_i \\
&= \sum_i^{|M(n)|} (f_i(a, 0) + f_i(0, b))(g_i(a, 0) + g_i(0, b))w_i \\
&= \sum_i^{|M(n)|} f_i(a, 0)g_i(0, b)w_i + \sum_i^{|M(n)|} f_i(0, b)g_i(a, 0)w_i
\end{aligned} \tag{6}$$

The first step follows by bilinearity of ϕ and the second does also because the cross terms $f_i(a, 0)g_i(a, 0)$ and $f_i(0, b)g_i(0, b)$ are not bilinear.

This shows that $\frac{1}{2}R(\langle n, n, n\rangle) \leq M(n)$, which by the definition of ω immediately gives $\theta \leq \omega$.

By definition of θ for $\epsilon > 0$, there is some $m > 1$ such that $r = R(\langle m, m, m\rangle) \leq m^{\theta+\epsilon}$. Then there are linear forms f_i, g_i: $K^{m\times m} \to K$ and matrices $w_i \in K^{m\times m}$. Such that for all $A, B \in K^{m\times m}$:

$$AB = \sum_i^r f_i(A)g_i(B)w_i. \tag{7}$$

However, K does not need to be a field (i.e. $K = \mathbb{F}$); K could, in fact, be a matrix algebra: $K = \mathbb{F}^{m^i\times m^i}$ for some $i \in \mathbb{N}$. Recall, an algebra over a field is a vector space equipped with a bilinear vector product. Then two K matrices A and B can be multiplied over K using r multiplications between elements of K, a number of additions between elements of K and some multiplications between elements of $\mathbb{F}$ and elements of K.

Notice that $K^{m\times m} \simeq \mathbb{F}^{m^{i+1}\times m^{i+1}}$, as $\mathbb{F}$-algebras (block decomposition). This gives rise to the following recurrence: $M(m^{i+1}) \leq rM(m^i)+cm^{2i}$, where c is proportional to the number of additions and scalar multiplications used to compute f_i and g_i. Because c, m and r are constants, we can solve this recurrence in terms of i, getting $M(m^i) = O(r^i)$ (the multiplications are not easier than addition and scalar multiplication).

Because M is monotone: $M(n) = O(r^{\log_m n}) = O(n^{\log_m r}) = O(n^{\theta+\epsilon})$. This gives $\omega \leq \theta$. □

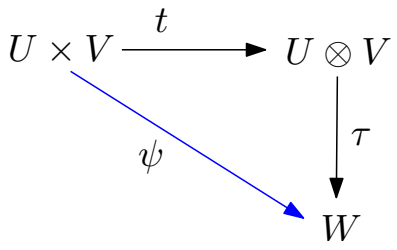

Figure 2: Universality of tensor product

Thus, bounding $R(\langle n, n, n\rangle)$, implies efficient multiplication algorithms.

Note that we showed $R(\langle 2, 2, 2\rangle) \leq 7$ in our discussion of Strassen's algorithm. Also R is sub-additive and sub-multiplicative (under direct sums and direct products respectively). In particular, $R(\langle n^l, n^l, n^l\rangle) = R(\bigotimes^l \langle n, n, n\rangle) \leq R(\langle n, n, n\rangle)^l$. So $R(\langle 2, 2, 2\rangle) \leq 7$, immediately gives $\omega < 2.81$, by the previous lemma.

This can be stated more generally:

Proposition 2 ([BCS97] Prop 15.3)**.** *If $R(\langle n, n, n\rangle) \leq r$, then $n^\omega \leq r$.*

4 Approximation

Strassen's algorithm is exact. Can we use fewer multiplications if we only want to approximate the product of two matrices? We will show that the answer is yes and that even more remarkably, if you can approximate matrix multiplication efficiently you can also efficiently compute it exactly. The notion of border rank $\underline{R}(\langle n, n, n\rangle)$ is introduced to make this precise.

In the following discussion a tensor t is approximated instead of the bilinear map $\langle n, n, n\rangle$. Approximating a tensor with r "operations" in effect implies an approximation for $\langle n, n, n\rangle$. This follows from the fact that any bilinear map $\psi : U \times V \to W$, for vector spaces U,V and W can be uniquely expressed as $\psi = \tau \circ t$, (Figure 2) where t is the tensor product $t : U \times V \to U \otimes V$ and $\tau : U \otimes V \to W$ is a linear transformation (see Theorem 14.2, pp. 346 [Rom05]). Overall the rank (and border rank) of the bilinear map can be characterized in terms of its associated tensor t (Proposition 14.44, pp 366 [BCS97]).

The notion extends to trilinear forms. In particular, for any trilinear $\overline{\psi} : U \times V \times W \to Z$, we have $\overline{\psi} = \overline{\tau} \circ \overline{t}$ for the tensor $\overline{t} : U \times V \times W \to U \otimes V \otimes W$ and a linear transform $\overline{\tau} : U \otimes V \otimes W \to Z$.

Definition 6. *A tensor $t \in \mathbb{F}^{n \times n \times n}$ is said to degenerate (of order q) to $\langle r\rangle$ iff there exists vectors $u_i(\epsilon), v_i(\epsilon), w_i(\epsilon) \in \mathbb{F}[\epsilon]^n$ for $1 \leq i \leq r$ such that*

$$\epsilon^{q-1} t + \epsilon^q t'(\epsilon) = \sum_{\rho=1}^{r} u_\rho(\epsilon) \otimes v_\rho(\epsilon) \otimes w_\rho(\epsilon) \tag{8}$$

for some $t'(\epsilon) = \mathbb{F}^{n \times n \times n}$

In essence, a tensor t degenerates to $\langle r\rangle$ suggests that it can be computed by determining the coefficient of ϵ^{q-1} (the error term) in the expansion of the right hand side of Equation 8.

We show that it is possible to obtain exact algorithms from approximate ones. With ϵ as the indeterminate in $\mathbb{F}^n[\epsilon]$ say we have the following representations

$$u_\rho(\epsilon) = \sum_i u_\rho^i \epsilon^i \ ; \ v_\rho(\epsilon) = \sum_j v_\rho^j \epsilon^j \ ; \ w_\rho(\epsilon) = \sum_k w_\rho^k \epsilon^k \tag{9}$$

where coefficients $u_\rho^i, v_\rho^j, w_\rho^k \in \mathbb{F}^n$. Thus expanding for t, the coefficient of ϵ^{q-1}, in equation 8 we get

$$t = \sum_{\rho=1}^{r} \sum_{i+j+k=q-1} u_\rho^i v_\rho^j w_\rho^k \tag{10}$$

There are $\binom{q}{2}$ ways of setting i, j, k such that $i + j + k = q - 1$, so t can be expressed exactly using $(q(q+1)/2)r$ terms. Overall we have that if a tensor t degenerates (of order q) to $\langle r \rangle$ then its rank, $R(t) \leq q^2 r$.

Definition 7. *The* border rank *of t, denoted as $\underline{R}(t)$, is the smallest r to which t degenerates, for some q.*

Intuitively, this suggests that if multiplication can be efficiently approximated, then it can be efficiently computed as well, because of the connection between rank, R, and ω. Formally, this gives:

Proposition 3 ([BCS97] Prop 15.10)**.** *If $\underline{R}(\langle n, n, n \rangle) \leq r$, then $n^\omega \leq r$.*

Bini et al. [BCRL79] showed that $\underline{R}(\langle 3, 2, 2 \rangle) \leq 5$ and extend this construction to achieve $\underline{R}(\langle 12, 12, 12 \rangle) \leq 1000$

Border rank, like rank, is sub-additive and sub-multiplicative under direct sums and direct products respectively. In particular, $\underline{R}(\langle 12^N, 12^N, 12^N \rangle) = \underline{R}(\langle 12, 12, 12 \rangle)^N$.

Hence we get the following inequality, $\underline{R}(\langle 12^N, 12^N, 12^N \rangle) \leq 10^{3N}$. Applying Proposition 3, we have $12^{N\omega} \leq 10^{3N}$. With sufficiently large N we get $12^\omega \leq 1000$, implying the following proposition.

Proposition 4 ([BCS97] Prop 15.9)**.** $\omega \leq \log_{12} 1000 = 2.78$

This can be further extended to direct sums with some effort (not stated in generality):

Theorem 1 (Schönhage's Theorem - [BCS97] 15.11)**.** *If $\underline{R}(\bigoplus_{i=1}^{s} \langle n, n, n \rangle) \leq r$ then $sn^\omega \leq r$.*

The intuitive meaning of this theorem is that if multiplication of many independent matrices (s sets) can be approximated efficiently then we can those approximate multiplications to compute exact multiplication more efficiently on single matrices. This can viewed as a generalization of Strassen's divide and conquer approach. We will not prove this theorem however the background and proof appears in [BCS97] Sections 15.3-5.

5 CW

Our goal is to get into a situation where we can apply Theorem 1, to get a bound on ω. We need to show that we can approximately compute many independent matrix multiplications in parallel efficiently. We start by looking at the trilinear version of Strassen's algorithm (which boils down to rewriting it):

$$
\begin{aligned}
&(\mathbf{A}_{1,1}+\mathbf{A}_{2,2})(\mathbf{B}_{1,1}+\mathbf{B}_{2,2})(\mathbf{C}_{1,1}+\mathbf{C}_{2,2})\\
&+(\mathbf{A}_{2,1}+\mathbf{A}_{2,2})\mathbf{B}_{1,1}(\mathbf{C}_{2,1}-\mathbf{C}_{2,2})\\
&+\mathbf{A}_{1,1}(\mathbf{B}_{1,2}-\mathbf{B}_{2,2})(\mathbf{C}_{1,2}+\mathbf{C}_{2,2})\\
&+\mathbf{A}_{2,2}(\mathbf{B}_{2,1}-\mathbf{B}_{1,1})(\mathbf{C}_{1,1}+\mathbf{C}_{2,1})\\
&+(\mathbf{A}_{1,1}+\mathbf{A}_{1,2})\mathbf{B}_{2,2}(\mathbf{C}_{1,1}+\mathbf{C}_{1,2})\\
&+(\mathbf{A}_{2,1}-\mathbf{A}_{1,1})(\mathbf{B}_{1,1}+\mathbf{B}_{1,2})\mathbf{C}_{2,2}\\
&+(\mathbf{A}_{1,2}-\mathbf{A}_{2,2})(\mathbf{B}_{2,1}+\mathbf{B}_{2,2})\mathbf{C}_{1,1}
\end{aligned}
\tag{11}
$$

$$
\begin{aligned}
=&(\mathbf{A}_{1,1}\mathbf{B}_{1,1}+\mathbf{A}_{1,2}\mathbf{B}_{2,1})\mathbf{C}_{1,1}\\
&+(\mathbf{A}_{1,1}\mathbf{B}_{1,2}+\mathbf{A}_{1,2}\mathbf{B}_{2,2})\mathbf{C}_{1,2}\\
&+(\mathbf{A}_{2,1}\mathbf{B}_{1,1}+\mathbf{A}_{2,2}\mathbf{B}_{2,1})\mathbf{C}_{2,1}\\
&+(\mathbf{A}_{2,1}\mathbf{B}_{1,2}+\mathbf{A}_{2,2}\mathbf{B}_{2,2})\mathbf{C}_{2,2}
\end{aligned}
$$

The sum of the 7 multiplication lines is exactly the trilinear form for 2-by-2 matrix multiplication.

As we stated previously Strassen algorithm shows $R(\langle 2,2,2\rangle) \leq 7$, and applying Theorem 1 gives $\omega < 2.81$. We take the analogous approach in the next few subsections where we will prove the following theorem:

Theorem 2. *For $q, N \in \mathbb{N}$, $\underline{R}(\bigoplus^s \langle q^N, q^N, q^N\rangle) \leq (q+2)^{3N}$. Where $s \geq \frac{1}{4}\left(\binom{2N}{N}+1\right)^{-1-o(1)}\binom{3N}{N,N,N}$.*

With this in hand we can plug into Theorem 1 to get a bound on ω (in Subsection 5.5).

5.1 CW's construction

Now we write down the analogous trilinear form construction from CW (this discussion is presented in Section 5 of [CW87] or Section 6 of [CW90]):

$$
\begin{aligned}
&\sum_{i=1}^{q}\epsilon^{-2}(a_0^0+\epsilon a_i^1)(b_0^0+\epsilon b_i^1)(c_0^0+\epsilon c_i^1)\\
&-\epsilon^{-3}(a_0^0+\epsilon^2\sum_{i=1}^{q}a_i^1)(b_0^0+\epsilon^2\sum_{i=1}^{q}b_i^1)(c_0^0+\epsilon^2\sum_{i=1}^{q}c_i^1)\\
&+(\epsilon^{-3}-q\epsilon^{-2})a_0^0b_0^0c_0^0
\end{aligned}
\tag{12}
$$

$$
=\sum_{i=1}^{q}(a_0^0b_i^1c_i^1+a_i^1b_0^0c_i^1+a_i^1b_i^1c_0^0)+O(\epsilon)\equiv t+O(\epsilon).
$$

Observe several things:

1. This identity is entirely symmetric in a, b, c.
2. There are $3(q+1)$ variables, $q+1$ for each of a, b, c.

3. The superscripts are in direct correlation with the subscript. If the subscript is 0 the superscript is 0, if the subscript is in $\{1, ..., q\}$, the superscript is 1. Let $A^0 = \{a_0\}$ and $A^1 = \{a_1, ..., a_q\}$, with B and C defined symmetrically. Then A^I, B^J, C^K are *blocks* of variables.

4. The RHS of the identity contains only terms with exactly one 0 superscript. So when looking at the sums individually, for example, $\sum_{i=1}^{q} a_0^0 b_i^1 c_i^1$ represents the matrix product $\langle 1, 1, q\rangle$ (a scalar times a row vector), the other two sums represent $\langle q, 1, 1\rangle$ and $\langle 1, q, 1\rangle$ respectively. Thus, the RHS of the identity does *not* represent a matrix multiplication, like it does in Strassen's algorithm.

5. This identity implies that the border rank of the RHS is at most $q + 2$, since there are $q + 2$ multiplications being used to represent it approximately.

This construction does not meet our goal yet (to efficiently approximate many independent matrix multiplication), by Observation 4, the RHS of the identity is not a matrix multiplication. However we can make it one. We will take the tensor product of this construction with itself $3N$ times, $t' = t^{\otimes 3N}$,for some $N \in \mathbb{N}$.

Lets consider the impact of the $3N$ tensor power (symmetric things happen for the objects associated with b and c):

$$\begin{gathered} t \to t^{\otimes 3N} \\ A^I, I \in \{0,1\} \to A^I, I \in \{0,1\}^{3N} \\ a_i^I, i \in \{0,1,...,q\}, I \in \{0,1\} \to a_i^I, i \in \{0,1,...,q\}^{3N}, I \in \{0,1\}^{3N} \\ \underline{R}(t) \leq q+2 \to \underline{R}(t^{\otimes 3N}) \leq \underline{R}(t)^{3N} \leq (q+2)^{3N} \end{gathered} \tag{13}$$

So now we're using $(q+2)^{3N}$ multiplications to represent t', and there are $(q+1)^{3N}$ variables a_i^I, for a total of $3(q+1)^{3N}$ variables among a, b, c. We consider the capital letters blocks of variables in the same sense as before ($a_i^I \in A^{I'}$ iff $I = I'$), moreover, each variable is in exactly one block.

5.2 Restriction 1

t' is still not sufficient to apply Theorem 1, we need to massage t' and remove some terms so that it actually represents the sum of independent multiplications. To accomplish this goal we will zero some of the blocks A^I, B^J and C^K. If A^I is zeroed all variables in it will be set to 0 (i.e. every variable with a superscript that matches). The first restriction that we apply is to set to zero every block whose superscript has hamming weight not equal to $2N$: $A^I = 0$ if $|I| \neq 2N$, similar for B^J's and C^K's. Call this restriction R_1. Notice that under this restriction every block contains exactly q^{2N} variables.

Now, before we proceed with the rest of the algorithm let us consider how this restriction effects t'. The case for $N = 1$ will be illuminating:

$$
\begin{aligned}
t &= \sum_{i=1}^{q} (a_0^0 b_i^1 c_i^1 + a_i^1 b_0^0 c_i^1 + a_i^1 b_i^1 c_0^0) \\
t \otimes t &= \sum_{i=1}^{q} (a_0^0 b_i^1 c_i^1 + a_i^1 b_0^0 c_i^1 + a_i^1 b_i^1 c_0^0) \otimes \sum_{j=1}^{q} (a_0^0 b_j^1 c_j^1 + a_j^1 b_0^0 c_j^1 + a_j^1 b_j^1 c_0^0) \\
&= \sum_{i=1,j=1}^{q} (a_{00}^{00} b_{ij}^{11} c_{ij}^{11} + a_{0j}^{01} b_{i0}^{10} c_{ij}^{11} + a_{0j}^{01} b_{ij}^{11} c_{i0}^{10} \\
&+ a_{i0}^{10} b_{0j}^{01} c_{ij}^{11} + a_{ij}^{11} b_{00}^{00} c_{ij}^{11} + a_{ij}^{11} b_{0j}^{01} c_{i0}^{10} \\
&+ a_{i0}^{10} b_{ij}^{11} c_{0j}^{01} + a_{ij}^{11} b_{i0}^{10} c_{0j}^{01} + a_{ij}^{11} b_{ij}^{11} c_{00}^{00})
\end{aligned}
\tag{14}
$$

Before we continue, note that for $N = 1$, a_{00}^{00}, b_{00}^{00} and c_{00}^{00} will be zeroed by our restriction (this eliminates the three diagonal terms in the last step of the previous equation). Keeping the restriction in mind, we have that:

$$
(t \otimes t \otimes t)|_{R_1} = \sum_{i=1,j=1,k=1}^{q} (a_{0jk}^{011} b_{i0k}^{101} c_{ij0}^{110} + a_{0jk}^{011} b_{ij0}^{110} c_{i0k}^{101} + a_{i0k}^{101} b_{0jk}^{011} c_{ij0}^{110} + a_{ij0}^{110} b_{0jk}^{011} c_{i0k}^{101} + a_{i0k}^{101} b_{ij0}^{110} c_{0jk}^{011} + a_{ij0}^{110} b_{i0k}^{101} c_{0jk}^{011}). \tag{15}
$$

Looking more closely at the first term $\sum_{i,j,k}^{q} a_{0jk}^{011} b_{i0k}^{101} c_{ij0}^{110} = A^{011} B^{101} C^{110}$. This is exactly the trilinear form of multiplication for q by q matrices. A similar situation occurs in the remainder of the terms.

$$
(t \otimes t \otimes t)|_{R_1} = A^{011} B^{101} C^{110} + A^{011} B^{110} C^{101} + A^{101} B^{011} C^{110} + A^{110} B^{011} C^{101} + A^{101} B^{110} C^{011} + A^{110} B^{101} C^{011}. \tag{16}
$$

It is easy to see that this generalizes to the $3N^{th}$ tensor power for arbitrary N. So $t'|_{R_1}$ is the sum of matrix multiplications, however, not all the products are independent. In the previous equation A^{011} occurs in both the first and second term. The remainder of the argument will show that it is possible to zero a few terms to make the remaining products independent while not eliminating too many products.

Thus, $t'|_{R_1}$ consists of matrix products $A^I B^J C^K$, in trilinear form, for $|I| = |J| = |K| = 2N$.

5.3 Restriction 2

Set $M = 2\binom{2N}{N} + 1$ and let H be the dense 3-arithmetic progression free subset of $[M]$, with $|H| > M^{1-o(1)}$. Select integers $0 \leq w_j < M$, for $j = 0, ..., 3N$ uniformly at random. Define for A^I, B^J and C^K, the follow hash functions ($\{0,1\}^{3N} \to (\mathbb{Z}/M)$):

$$h_A(I) \equiv \sum_{j=1}^{3N} I_j w_j (\text{mod } M)$$
$$h_B(J) \equiv w_0 + \sum_{j=1}^{3N} J_j w_j (\text{mod } M) \tag{17}$$
$$h_C(K) \equiv (w_0 + \sum_{j=1}^{3N} (2 - K_j) w_j)/2 (\text{mod } M)$$

Then we have that $h_A(I) + h_B(J) - 2h_C(K) \equiv 0 (\text{mod } M)$. This is because $I_j + J_j + K_j = 2$ for all $j = 1, ..., 3N$, by the basic construction of t (Observation 4).

Now we introduce a second restriction, R_2. For all I, (respectively J, K) such that $h_A(I) \notin H$ ($h_B(J) \notin H, h_C(K) \notin H$), set A^I to zero (B^J, C^K).

This means that the remaining products $A^I B^J C^K$ in $t'|_{R_1,R_2}$ have $h = h_A(I) = h_B(J) = h_C(K)$ for some $h \in H$, because $h_A(I) + h_B(J) \equiv 2h_C(K) (\text{mod } M)$ and $h_A(I), h_B(J), h_C(K) \in H$; so by the definition of an ap-3 free set we have equality.

5.4 Restriction 3

Recapping after two restrictions t' looks like:

$$t'|_{R_1,R_2} = \sum_{|I|=|J|=|K|=2N, I\cap J\cap K=\emptyset, h_A(I), h_B(J), h_C(K)\in H} A^I B^J C^K. \tag{18}$$

For a fixed $h \in H$, let L_h be the list of all products of the form $A^I B^J C^K$ in $t'|_{R_1,R_2}$ with $h = h_A(I) = h_B(J) = h_C(K)$.

Claim 1. $\mathrm{E}_w[|L_h|] = \binom{3N}{N,N,N} M^{-2}$.

Proof. Recalling that R_1 forces $|I| = |J| = |K| = 2N$ and Observation 4 we can see that number of way to assign superscripts to A, B and C is so that each index exactly one is zero, is $\binom{3N}{N,N,N}$. The probability that $h_A(I) = h_B(J) = h$ is M^{-2}, since the hash functions are independent (w_0 is not present in h_A). Then R_2 forces $h_C(J) = h$ because the other two are. By linearity, the expected number of items in L_h is as claimed. □

We need another probabilistic claim:

Claim 2. *The expected number of unordered pairs of products* $(A^I B^J C^K), (A^{I'} B^{J'} C^{K'}) \in L_h$ *with* $K' = K$ *is*

$$\frac{1}{2}\binom{3N}{N,N,N}\left(\binom{2N}{N} - 1\right) M^{-3}. \tag{19}$$

Proof. Again $\binom{3N}{N,N,N}$ is the number of valid superscript combinations for I, J, K. Fix $A^I B^J C^K$. $\binom{2N}{N} - 1$ counts the number of the other J' such that are compatible with C^K (I' is forced by K and J'). Then the probability that $h_C(K) = h_B(J) = h_B(J') = h$ is M^{-3}, like the argument in the previous claim. The factor of $\frac{1}{2}$ follows because we're counting unordered pairs. This completes the claim. □

The previous claim is symmetric around the choice of which variable block has to be the same, so there are at most 3 times as many pairs sharing any block.

The final restriction, R_3, will be to zero blocks so that there are no pairs of products in the list which share a block, if we do is for all L_h, the products remaining in $t'|_{R_1,R_2,R_3}$ will all be disjoint. To this end, suppose we wish to zero a block B^J be because the pair of products $A^I B^J C^K$ and $A^{I'} B^{J'} C^K$ conflict (sharing a C-block). If v products are eliminated from L_h by setting B^J to zero, then at least $\binom{v}{2} + 1$ pairs are removed from the list, the ones between the v products containing B^J and the original conflict between $A^I B^J C^K$ and $A^{I'} B^{J'} C^K$. Since $\binom{v}{2} + 1 \geq v$ more pairs are removed than products. We get the following lemma by combining the previous discussion and the two claims (and noting that $(\binom{2N}{N} - 1)/M < \frac{1}{2}$).

Lemma 1.

$$\mathrm{E}_w[|L_h|] \geq \binom{3N}{N,N,N} M^{-2} - \frac{3}{2}\binom{3N}{N,N,N}\left(\binom{2N}{N} - 1\right) M^{-3} \geq \frac{1}{4}\binom{3N}{N,N,N} M^{-2}. \tag{20}$$

The preceding lemma say that we can eliminate the sharing of blocks without eliminating too many products.

Then $\mathrm{E}_w[\sum_{h\in H} |L_h|] \geq \frac{1}{4}|H|\binom{3N}{N,N,N} M^{-2} = \frac{1}{4}\binom{3N}{N,N,N} M^{-1-o(1)}$. Fix w such which realizes at least this expectation. Because of the restrictions we applied the only products remaining within $t'|_{R_1,R_2,R_3}$ are variable disjoint, represent q^N square matrix products and must number at least $\frac{1}{4}\binom{3N}{N,N,N} M^{-1-o(1)}$. Thus we have completed the proof of Theorem 2.

5.5 The Final Step

Now we apply Theorem 1 to Theorem 2, with $r = (q+2)^{3N}$, $s = \frac{1}{4}\binom{3N}{N,N,N} M^{-1-o(1)}$ and $n = q^N$. This immediately gives

$$(q+2)^{3N} \geq \frac{1}{4}\binom{3N}{N,N,N} M^{-1-o(1)} q^{N\omega}. \tag{21}$$

What remains is to optimize ω with respect to the choice of q. Expand the binomial coefficient using Stirling's approximation, take N to infinity and then take the N^{th} root gives:

$$(q+2)^3 \geq \frac{3^3}{2^2} q^{\omega}. \tag{22}$$

Picking $q = 8$ will optimize the this inequality resulting in $\omega \leq log_8(4000/27) < 2.404$.

Theorem 3 ([CW87],[CW90])**.** $\omega < 2.404$.

6 Conclusion

The more complex version presented in [CW90] Sections 7-8 achieves $\omega < 2.376$.

References

[BCRL79] D. Bini, M. Capovani, F. Romani, and G. Lotti, $O(n^{2.7799})$ *complexity for* $n \times n$ *approximate matrix multiplication*, Inf. Process. Lett. **8** (1979), 234–235.

[BCS97] P. Bürgisser, M. Clausen, and M.A. Shokrollahi, *Algebraic complexity theory*, Springer Verlag, 1997.

[CW87] D. Coppersmith and S. Winograd, *Matrix multiplication via arithmetic progressions*, Proceedings of the nineteenth annual ACM symposium on Theory of computing, ACM New York, NY, USA, 1987, pp. 1–6.

[CW90] ———, *Matrix multiplication via arithmetic progressions*, Journal of Symbolic Computation **9** (1990), no. 3, 251–280.

[Rom05] S. Roman, *Advanced Linear Algebra*, Springer Verlag, 2005.

[Str69] V. Strassen, *Gaussian elimination is not optimal*, Numerische Mathematik **13** (1969), no. 4, 354–356.

[Wik09] Wikipedia, *Strassen algorithm — wikipedia, the free encyclopedia*, 2009, [Online; accessed 1-December-2009].