The Chasm at Depth Four, and Tensor Rank:
Old results, new insights

Suryajith Chillara [*]   Mrinal Kumar [†]   Ramprasad Saptharishi [‡]   V Vinay [§]

August 1, 2017

Abstract

Agrawal and Vinay [AV08] showed how any polynomial size arithmetic circuit can be thought of as a depth four arithmetic circuit of subexponential size. The resulting circuit size in this simulation was more carefully analyzed by Koiran [Koi12] and subsequently by Tavenas [Tav15]. We provide a simple proof of this chain of results. We then abstract the main ingredient to apply it to formulas and constant depth circuits, and show more structured depth reductions for them.

In an apriori surprising result, Raz [Raz10] showed that for any n and d, such that $\omega(1) \leq d \leq O\left(\frac{\log n}{\log \log n}\right)$, constructing explicit tensors $T : [n]^d \to \mathbb{F}$ of high enough rank would imply superpolynomial lower bounds for arithmetic formulas over the field $\mathbb{F}$. Using the additional structure we obtain from our proof of the depth reduction for arithmetic formulas, we give a new and arguably simpler proof of this connection. We also extend this result for homogeneous formulas to show that, in fact, the connection holds for any d such that $\omega(1) \leq d \leq n^{o(1)}$.

1   Introduction

Agrawal and Vinay [AV08] showed how any polynomial size[1] arithmetic circuit can be thought of as a depth four arithmetic circuit of subexponential size. This provided a new direction to seek lower bounds in arithmetic circuits. A long list of papers attest to increasingly sophisticated lower bound arguments, centered around the idea of shifted partial derivates due to Kayal, to separate the so called arithmetic version of P vs NP (cf. [Sap15]).

[*]Chennai Mathematical Institute, Research supported in part by a TCS PhD fellowship. Part of the work done while visiting Tel Aviv University. suryajith@cmi.ac.in

[†]Rutgers University, Research supported in part by a Simons Graduate Fellowship. Part of the work done while visiting Tel Aviv University. mrinal.kumar@rutgers.edu

[‡]Tel Aviv University ramprasad@cmi.ac.in. The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575.

[§]Limberlink Technologies Pvt Ltd and Chennai Mathematical Institute vinay@jed-i.in

[1]in fact, subexponential size

The depth reduction chasm was more carefully analyzed by Korian [Koi12] and subsequently by Tavenas [Tav15]. Given the importance of these depth reduction chasms, it is natural to seek new and/or simpler proofs. In this work, we do just that.

We use a simple combinatorial property to prove our result. We then show how this can be extended to showing chasms for formulas and constant depth circuits. In the case of formulas, we show the top layer of multiplication gates have a much larger number of factors and therefore has more structure than a typical depth reduced circuit. We hope that such structural properties lead to better lower bounds for formulas. In fact, we use this additional structure to give a new proof of a result of Raz [Raz10] which shows that for an appropriate range of parameters, constructing explicit tensors of high enough rank implies super-polynomial lower bounds for arithmetic formulas.

More formally, let $f \in \mathbb{F}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d]$ be a set multilinear polynomial of degree d in nd variables, where for every $i \in [d]$, $\mathbf{x}_i$ is a subset of variables of size n. In a natural way, f can be viewed as a tensor $f : [n]^d \to \mathbb{F}$. Raz [Raz10] showed if $\omega(1) \leq d \leq O(\log n / \log \log n)$ and f is computed by an arithmetic formula of size $\text{poly}(n)$, then the rank of f as a tensor is far from n^{d-1} (the trivial upper bound[2]). We use the additional structure obtained from our proof of depth reduction for formulas and constant depth arithmetic circuits, to give a very simple of proof of this result. As an extension, we also show that, in fact, the tensor rank of f is far from n^{d-1} as long as f is computed by a *homogeneous* formula of polynomial size and d is such that $\omega(1) \leq d \leq n^{o(1)}$.

This write up is organised as follows. We give new proofs of depth reduction for arithmetic circuits (Section 2), for homogeneous arithmetic formulas (Section 3) and for constant depth arithmetic circuits (Section 4). We end by applying the new proof of depth reduction for homogeneous formulas to show a simple proof of Raz's upper bound [Raz10] on the tensor rank of polynomials computed by small arithmetic formulas in Section 5.

For standard definitions concerning arithmetic circuits, arithmetic formulas etc, we refer the reader to the survey of Saptharishi [Sap15]. For an introduction to connections between tensor rank and arithmetic circuits, we refer the reader to an excellent summary of such results in Raz's original paper [Raz10]. Throughout this paper, unless otherwise stated, by *depth reduction*, we mean a reduction to homogeneous depth four circuits. By a $\Sigma\Pi\Sigma\Pi^{[b]}$ circuit, we denote a depth four circuit such that the fan-in of every product gate at the bottom level is *at most d*, and by $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit, we denote a $\Sigma\Pi\Sigma\Pi^{[b]}$ circuit which also has the property that the fan-in of every product gate adjacent to the output gate has fan-in *at least a*, i.e the polynomials computed at the gates adjacent to the output gate have have *at least a* non-trivial factors.

---

[2]We know that there exist tensors $g : [n]^d \to \mathbb{F}$ of rank n^{d-1}/d.

2 Depth reduction for arithmetic circuits

We shall need the classical depth reduction of [VSBR83, AJMV98].

Theorem 2.1 ([VSBR83, AJMV98])**.** *Let f be an n-variate degree d polynomial computed by an arithmetic circuit Φ of size s. Then there is an arithmetic circuit Φ' computing f and has size $s' = \mathrm{poly}(s, n, d)$ and depth $O(\log d)$.*

Moreover, the reduced circuit Φ' has the following properties:

1. The circuit is homogeneous.
2. All multiplication gates have fan-in at most 5.
3. If u is any multiplication gate of Φ', all its children v satisfy $\deg(v) \leq \deg(u)/2$.

These properties can be inferred from their proof. A simple self-contained proof may be seen in [Sap15]. Agrawal and Vinay [AV08] showed that arithmetic circuits can in fact be reduced to depth four, and the result was subsequently strengthened by Koiran [Koi12] and by Tavenas [Tav15].

Theorem 2.2 ([AV08, Koi12, Tav15])**.** *Let f be an n-variate degree d polynomial computed by a size s arithmetic circuit. Then, for any $0 < t \leq d$, f can be computed by a homogeneous $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of top fan-in $s^{O(d/t)}$ and size $s^{O(t+d/t)}$.*

To optimize the size of the final depth four circuit, we should choose $t = \sqrt{d}$ to get a $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of size $s^{O(\sqrt{d})}$. Note that this implies that if we could prove a lower bound of $n^{\omega(\sqrt{d})}$ for such $\Sigma\Pi\Sigma\Pi^{[\sqrt{d}]}$ circuits, then we would have proved a lower bound for general circuits. In this section, we shall see a simple proof of Theorem 2.2.

Proof of Theorem 2.2. Using Theorem 2.1, we can assume that the circuit has $O(\log d)$ depth. If g is a polynomial computed at any intermediate node of C, then from the structure of C we have a homogeneous expression

$$g \quad = \quad \sum_{i=1}^{s} g_{i1} \cdot g_{i2} \cdot g_{i3} \cdot g_{i4} \cdot g_{i5} \tag{2.3}$$

where each g_{ij} is computed by a node in C as well, and $\deg(g_{ij}) \leq \deg(g)/2$. In particular, if g were the output gate of the circuit, the RHS may be interpreted as a $\Sigma\Pi\Sigma\Pi^{[d/2]}$ circuit of top fan-in s computing f. To obtain a $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit eventually, we shall perform the following steps on the output gate:

1. For each summand $g_{i1} \ldots g_{ir}$ in the RHS, pick the gate g_{ij} with largest degree (if there is a tie, pick the one with smaller index j). If g_{ij} has degree greater than t, expand g_{ij} in-place using (2.3).

2. Repeat this process until all g_{ij}'s on the RHS have degree at most t.
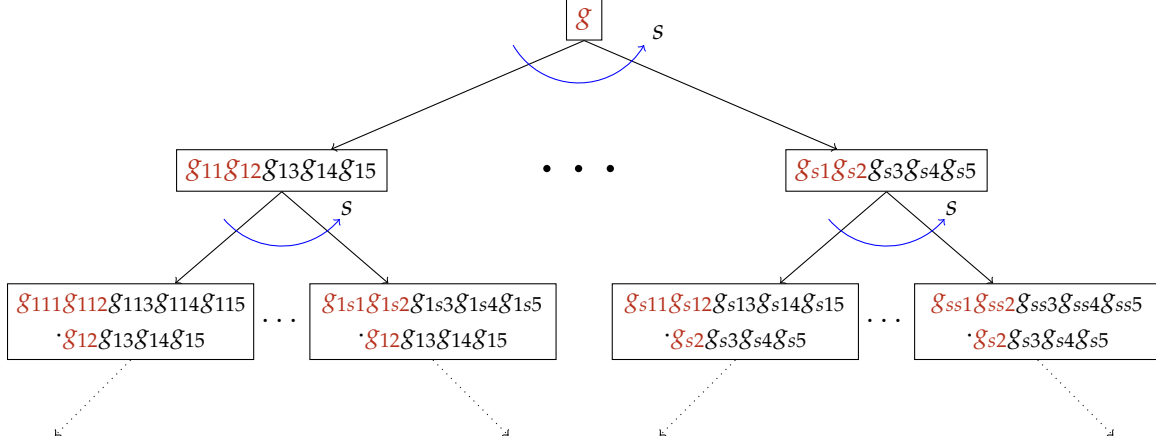
3

Figure 1: Depth reduction analysis

Each iteration of the above procedure increases the top fan-in by a multiplicative factor of s. If we could show that the in $O(d/t)$ iterations all terms on the RHS have degree at most t, then we would have obtained an $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of top fanin $s^{O(d/t)}$ computing f.

Label a term g_{ij} *bad* if its degree is more than $t/8$. To bound the number of iterations, we count the number of bad terms in each summand. Since we would always maintain homogeneity, the number of bad terms in any summand is at most $8d/t$ (i.e., not too many). We show each iteration *increases* the number of bad terms by at least one. This bounds the number of iterations by $8d/t$.

In (2.3), if $\deg(g) = k$, the largest degree term of any summand on the RHS is at least $k/5$ (since the sum of the degrees of the five terms must add up to k) and so continues to be bad if $k > t$. But the largest degree term can have degree at most $k/2$. Hence the other four terms must together contribute at least $k/2$ to the degree. This implies that the second largest term in each summand has degree at least $k/8$. This term is bad too, if we started with a term of degree greater than t. Therefore, as long as we are expanding terms of degree more than t using (2.3), we are guaranteed its replacements have at least one additional bad term. As argued earlier, we can never have more than $8d/t$ such terms in any summand and this bounds the number of iterations by $8d/t$.

Observe that the above procedure can be viewed as a tree, as described in Figure 1, where each node represents an intermediate summand in the iterative process. From (2.3) it is clear that the tree is s-ary. Furthermore, the number of "bad" terms strictly increases as we go down in the tree (these are marked in red in Figure 1). Since the total number of bad terms in any node can be at most $8(d/t)$, the depth of the tree is at most $8(d/t)$. Therefore, the total number of leaves is at most $s^{(8d/t)}$. Moreover, since every polynomial with degree at most t can be written as a sum of at most $n^{O(t)}$ monomials, the total size of the resulting $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit is at most $s^{O(t+d/t)}$ (since $s \geq n$). □

4

3 Depth reduction for homogeneous formulas

For the class of homogeneous formulas and shallow circuits, we will show that they can be depth reduced to a more structured depth four circuit.

To quickly recap the earlier proof, we began with an equation $f = \sum_i g_{i1} \cdot g_{i2} \cdot g_{i3} \cdot g_{i4} \cdot g_{i5}$ and recursively applied the same expansion on all the large degree g_{ij}'s. The only property we really used was that in the above equation, there were at least two g_{ij} that had large degree.

For the case of homogeneous formulas and shallow circuits, there are better expansions that we could use as a starting point.

Theorem 3.1 ([HY11]). *Let f be an n-variate degree d polynomial computed by a size s homogeneous formula. Then, f can be expressed as*

$$f \quad = \quad \sum_{i=1}^{s} f_{i1} \cdot f_{i2} \cdots f_{ir} \tag{3.2}$$

where

1. *the expression is homogeneous,*

2. *for each i, j, we have $\left(\frac{1}{3}\right)^j d \leq \deg(f_{ij}) \leq \left(\frac{2}{3}\right)^j d$ and $r = \Theta(\log d)$,*

3. *each f_{ij} is also computed by homogeneous formulas of size at most s.*

With this, we are ready to prove a more structured depth reduction for homogeneous formulas.

Theorem 3.3. *Let f be a homogeneous n-variate degree d polynomial computed by a size s homogeneous formula. Then for any $0 < t \leq d$, f can be equivalently computed by a homogeneous $\Sigma\Pi^{[a]}\Sigma\Pi^{[t]}$ formula of top fan-in $s^{10(d/t)}$ where*

$$a > \frac{1}{10} \frac{d}{t} \log t.$$

The resulting depth four circuit is more structured in the sense that the multiplication gates at the second layer have a much larger fan-in (by a factor of $\log t$). In Theorem 2.2, we only know that the polynomials feeding into these multiplication gates have degree at most t. The theorem above states that if we were to begin with a homogeneous formula, the degree t polynomials factorize further to give $\Theta((d/t) \log t)$ non-trivial polynomials instead of $\Theta(d/t)$ as obtained in Theorem 2.2.

Proof. We start with equation (3.2) which is easily seen to be a homogeneous $\Sigma\Pi\Sigma\Pi^{[2d/3]}$ circuit with top fan-in s:

$$f \quad = \quad \sum_{i=1}^{s} f_{i1} \cdot f_{i2} \cdots f_{ir}$$

5

To obtain a $\Sigma\Pi^{[\Theta((d/t)\log t)]}\Sigma\Pi^{[t]}$ circuit eventually, we shall perform the following steps on the output gate:

1. For each summand $f_{i1} \ldots f_{ir}$ in the RHS, pick the gate f_{ij} with largest degree (if there is a tie, pick the one with smaller index j). If f_{ij} has degree more than t, expand that f_{ij} in-place using (3.2).

2. Repeat this process until all f_{ij}'s on the RHS have degree at most t.

Each iteration again increases the top fan-in by a factor of s. Again, as long as we are expanding terms using (3.2) of degree $k > t$, we are guaranteed by Theorem 3.1 that each new summand has at least one more term of degree at least $k/9 > t/9$.

To upper bound the number of iterations, we use a potential function — the number of factors of degree strictly greater than $t/9$ in a summand. A factor that is of degree $k > t$ and which is expanded using (3.2) contributes at least two factors of degree $> t/9$ per summand. Thus, the net increase in the potential per iteration is at least 1. Since this is a homogeneous computation, there can be at most $9d/t$ such factors of degree $> t/9$. Thus, the number of iterations must be bounded by $9d/t$ thereby yielding a $\Sigma\Pi\Sigma\Pi^{[t]}$ of top fan-in at most $s^{9(d/t)}$ and size $s^{(t+9d/t)}$. This argument is similar to the argument in the proof of Theorem 2.2.

We now argue that the fan-in of every product gate at the second level in the $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit obtained is $\Theta(d/t \log t)$.

To this end, we shall now show that we require $\Theta(d/t)$ iterations to make all the factors have degree at most t. This, along with the fact that every iteration introduces a certain number of non-trivial factors in every product will complete the proof. We will say a factor is *small* if degree is at most t and *big* otherwise. To prove a lower bound on the number of iterations, we shall use a different potential function — the total degree of all the big factors.

Given the geometric progression of degrees in Theorem 3.1, we can easily see that the total degree of all the small factors in any summand is bounded above by $3t$. Hence, the total degree of all the big terms is $d - 3t$. But whenever (3.2) is applied on a big factor, we introduce several small degree factors with total degree of at most $3t$. Hence, the potential drops by at most $3t$ per iteration. This implies that we require $(d/3t)$ iterations to make it a constant.

Since every expansion via (3.2) introduces at least $(\log_3 t)$ non-trivial terms, it would then follow that every summand at the end has $\frac{1}{(3\log 3)}\frac{d}{t}\log t > \frac{1}{10}\frac{d}{t}\log t$ non-trivial factors. $\qquad\square$

3.1 An alternate proof

While we proved Theorem 3.3 along the lines of Theorem 2.2, it is possible to provide an alternate proof of it. We provide a sketch. Starting with a homogeneous formula, by Theorem 2.2 we get a

$\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of the form

$$\sum_{i=1}^{s'} Q_{i1} \dots Q_{ir}$$

where $\deg(Q_{ij}) \le t$ and $s' = s^{O(d/t)}$. From the innards of this proof, it can be observed that each of the Q_{ij}'s is indeed computable by a homogeneous formula (formula, not a circuit) of size at most s. By multiplying several polynomials (if necessary) of degree at most $t/2$, we may assume that there are $\Theta(d/t)$ polynomials Q_{ij} in each summand, with their degree between $t/2$ and t.

Each of these polynomials may be expanded using (3.2). Since each such expansion adds $O(\log t)$ additional factors and increases the fan-in by a factor of s, the overall top fan-in is now $s' \cdot s^{O(d/t)}$. The number of factors however increases from $\Theta(d/t)$ to $\Theta((d/t)\log t)$. The resulting circuit is thus a $\Sigma\Pi^{[\Theta((d/t)\log t)]}\Sigma\Pi^{[t]}$ circuit of top fan-in $s^{O(d/t)}$.

4 Depth reduction for constant depth circuits

In the same vein, a natural question is if we can obtain more structure for a constant depth circuit. For example, is the resulting depth four circuit more structured when we begin with a depth 100 circuit? By suitably adapting the expansion equation, our approach can answer this question.

Lemma 4.1. *Let f be an n-variate degree d polynomial computed by a size s circuit of product-depth[3] Δ. Then f can be expressed as*

$$f \quad = \quad \sum_{i=1}^{s^2} f_{i2} \cdot f_{i3} \cdots f_{ir} \; \cdot \; g_{i1} \cdots g_{i\ell} \tag{4.2}$$

where

1. *the expression is homogeneous,*

2. *for each i, j, we have $\left(\frac{1}{3}\right)^j d \le \deg(f_{ij}) \le \left(\frac{2}{3}\right)^j d$ and $r = \Theta(\log d)$,*

3. *each f_{ij} and g_{ij} is also computed by homogeneous formulas of size at most s and product-depth Δ.*

4. *$\ell = \Omega(d^{1/\Delta})$*

5. *all g_{ij}, f_{ij} are polynomials of degree at least 1.*

Using this equation for the depth reduction yields the following theorem.

Theorem 4.3. *Let f be an n-variate degree d polynomial computed by a size s homogeneous formula of product-depth Δ. Then for any parameter $t = o(d)$, we can compute f equivalently by a homogeneous $\Sigma\Pi^{[\Theta((d/t)\cdot t^{1/\Delta})]}\Sigma\Pi^{[t]}$ circuit of top fan-in at most $s^{O(d/t)}$ and size $s^{O(t+d/t)}$.*

---

[3]the product depth is the number of multiplication gates encountered in any path from root to leaf

The multiplication gates at the second layer of the resulting depth four circuit have a much larger fan-in than what is claimed in Theorem 2.2 or Theorem 3.3. When we begin with additional structure in the circuit, it seems we get additional structure in the resulting depth four circuit. Specifically, let us fix $t = \sqrt{d}$. The fan-in of the outer product gate would be $\Theta(\sqrt{d})$ for a general circuit (Theorem 2.2), $\Theta(\sqrt{d} \cdot \log d)$ for a homogeneous formula (Theorem 3.3), and $\Theta(\sqrt{d} \cdot d^{1/100})$ for a circuit of depth 100 (Theorem 4.3).

Proof of Lemma 4.1. Let Φ be the product depth-Δ formula computing f. By Theorem 3.1, we get

$$ f \quad = \quad \sum_{i=1}^{s} f_{i1} \cdot f_{i2} \cdots f_{ir} \tag{4.4} $$

with the required degree bounds. From the proof of Theorem 3.1, it follows that each f_{ij} is in fact a product of disjoint sub-formulas of Φ, and hence in particular f_{i1} is computable by size s formulas of product-depth Δ. We shall expand f_{i1} again to obtain the g_{ij}s.

Since f_{i1} is a polynomial of degree at least $d/3$ computed by a size s formula Φ' of product-depth Δ, there must be some multiplication gate h in Φ' of fan-in $\Omega(d^{1/\Delta})$. Therefore,

$$ f_{i1} \quad = \quad A \cdot [h] \quad + \quad B. $$

Here, $[h]$ is the polynomial computed at the gate h. Since B is computed by Φ' with $h = 0$, we can induct on B to obtain

$$ f_{i1} \quad = \quad A_1[h_1] + \cdots + A_s[h_s] $$

where each h_i is a multiplication gate of fan-in $\Omega(d^{1/\Delta})$. Plugging this in (4.4), and replacing $[h_i]$'s by the factors, gives (4.2). $\qquad\square$

5   An Application: Tensor rank and formula lower bounds

Tensors are a natural *higher dimensional* analogue of matrices. For the purposes of this short note, we shall take the equivalent perspective of *set-multilinear polynomials*. A detailed discussion on this can be seen in [Sap15].

Definition 5.1 (Set-multilinear polynomials)**.** *Let $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ be a partition of variables and let $|\mathbf{x}_i| = m_i$. A polynomial $f(\mathbf{x})$ is said to be* set-multilinear *with respect to the above partition if every monomial m in f satisfies $|m \cap X_i| = 1$ for all $i \in [d]$.* $\diamondsuit$

In other words, each monomial in f picks up one variable from each part in the partition. It is easy to see that many natural polynomials such as the determinant, the permanent are all set-multilinear for an appropriate partition of variables.

With this interpretation, a rank-1 tensor is precisely a *set-multilinear* product of linear forms such as

$$f(\mathbf{x}) \quad = \quad \ell_1(\mathbf{x}_1) \cdots \ell_d(\mathbf{x}_d)$$

where each $\ell_i(\mathbf{x}_i)$ is a linear form in the variables in $\mathbf{x}_i$.

Definition 5.2 (Tensor rank, as set-multilinear polynomials)**.** *For polynomial $f(\mathbf{x})$ that is set-multilinear with respect to $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$, the tensor rank of f (denoted by $\mathrm{TensorRank}(f)$) is the smallest r for which f can be expressed as a* set-multilinear $\Sigma\Pi\Sigma$ *circuit:*

$$f(\mathbf{x}) \quad = \quad \sum_{i=1}^{r} \ell_{i1}(\mathbf{x}_1) \cdots \ell_{id}(\mathbf{x}_d).$$

$\Diamond$

However, even computing the rank of an degree-3 tensor is known to be NP-hard [Hås90]. But one could still ask if one can prove good upper or lower bounds for some specific tensors, or try to find an explicit tensor with large rank.

Properties of tensor rank

The following are a couple of basic properties that follow almost immediately from the definitions.

Lemma 5.3 (Sub-additivity of tensor rank)**.** *Let f and g be two set-multilinear polynomials on $\mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$. Then, $\mathrm{TensorRank}(f + g) \leq \mathrm{TensorRank}(f) + \mathrm{TensorRank}(g)$.*

Lemma 5.4 (Sub-multiplicativity of tensor rank)**.** *Let $f(\mathbf{y})$ be set-multilinear on $\mathbf{y} = \mathbf{y}_1 \sqcup \cdots \sqcup \mathbf{y}_a$ and $g(\mathbf{z})$ be set-multilinear on $\mathbf{z} = \mathbf{z}_1 \sqcup \cdots \mathbf{z}_b$ with $\mathbf{y} \cap \mathbf{z} = \emptyset$. Then polynomial $f \cdot g$ that is set-multilinear on $\mathbf{y} \cup \mathbf{z} = \mathbf{y}_1 \sqcup \cdots \sqcup \mathbf{y}_a \sqcup \mathbf{z}_1 \sqcup \cdots \mathbf{z}_b$ satisfies[4]*

$$\mathrm{TensorRank}(f \cdot g) \leq \mathrm{TensorRank}(f) \cdot \mathrm{TensorRank}(g).$$

The following is a trivial upper bound for the tensor rank of any degree d set-multilinear polynomial f.

Lemma 5.5. *Let f be a set-multilinear polynomial with respect to $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ and say $n_i = |\mathbf{x}_i|$. Then,*

$$\mathrm{TensorRank}(f) \quad \leq \quad \frac{\prod_{i=1}^{d} n_i}{\max_i n_i}.$$

In particular, if all $n_i = n$, then $\mathrm{TensorRank}(f) \leq n^{d-1}$.

---

[4]Tensor rank, in general, **does not** satisfies the relation $\mathrm{TensorRank}(f \cdot g) = \mathrm{TensorRank}(f) \cdot \mathrm{TensorRank}(g)$. For a concrete counter example, see [CJZ17].

A counting argument would imply that there do exist tensors of rank at least n^{d-1}/d as each elementary tensor has *nd degrees of freedom* and an arbitrary tensor has n^d *degrees of freedom.* [5]

So, it is a natural question to understand if we can construct explicit tensors of high rank? Raz [Raz10] showed that in certain regimes of parameters involved, an answer to the above question would yield arithmetic formula lower bounds. We elaborate on this now.

5.1   Tensor rank of small formulas

Henceforth, the variables in $\mathbf{x}$ are partitioned as $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ with $|\mathbf{x}_i| = n$ for all $i \in [d]$. The main motivating question of Raz [Raz10] was the following:

> If f is a set-multilinear polynomial that is computed by a small formula, what can one say about its tensor rank?

Raz gave a partial[6] answer to this question by showing the following result.

Theorem 5.6. *Let Φ be a formula of size $s \leq n^c$ computing a set-multilinear polynomial $f(\mathbf{x})$ with respect to $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$. If $d = O(\log n / \log \log n)$, then,*

$$\mathrm{TensorRank}(f) \quad \leq \quad \frac{n^d}{n^{d/\exp(c)}}.$$

To prove Theorem 5.6, Raz [Raz10] first showed that when d is small compared to n (specifically, $d = O(\log n / \log \log n)$), any small formula can be converted to a *set-multilinear* formula with only a polynomial over-head. Formally, he shows the following theorem, which is interesting and surprising in its own right[7].

Definition 5.7 (Set-multilinear formulas)**.** *A formula Φ is said to be a* set-multilinear formula *if every gate in the formula computes a set-multilinear polynomial syntactically.*

That is, if f and g are polynomials computed by children of a $+$ gate, then both f and g are set-multilinear polynomials of the same degree over $\mathbf{x}$, with possibly different partitions. And if f and g are polynomials computed by children of a $\times$ gate, then both f and g are set-multilinear polyomials on disjoint sets of variables.   ◇

Theorem 5.8 ([Raz10])**.** *Suppose $d = O\left(\frac{\log n}{\log \log n}\right)$. If Φ is a formula of size $s = \mathrm{poly}(n)$ that computes a set-multilinear polynomial $f(\mathbf{x}_1, \cdots, \mathbf{x}_d)$, then there is a set-multilinear formula of $\mathrm{poly}(s)$ size that computes f as well.*

---

[5]One might think that the above upper bound of n^{d-1} should be tight. Bizarrely, it is not! For example (cf. [Pam85]), the maximum rank of any tensor of shape $2 \times 2 \times 2$ is 3 and not 4 as one might expect! Tensor rank also behaves in some strange ways under *limits* unlike the usual matrix rank.

[6]Partial in the sense that we do not know if the bound is tight.

[7]Indeed, it was believed that even transforming a formula into a homogeneous formula would cause a superpolynomial blow up in its size if the degree of the polynomial computed by the formula is growing with n.

He then proceeds to show that set-mutlilinear formulas of polynomial size can only compute polynomials with tensor rank non-trivially far from the upper bound of n^{d-1}. More formally, he shows the following theorem.

Theorem 5.9 ([Raz10]). *Let Φ be a set-multilinear formula of size $s \leq n^c$ computing a polynomial $f(\mathbf{x}_1, \cdots, \mathbf{x}_d)$. Then,*

$$\text{TensorRank}(f) \leq \frac{n^d}{n^{d/\exp(c)}}.$$

It is immediately clear that Theorem 5.9 and Theorem 5.8 imply Theorem 5.6. In this section, we give a simple proof of Theorem 5.9 using Theorem 3.3. We refer the reader to Raz's paper [Raz10] or [Sap15] for a full proof of Theorem 5.8.

Proof of Theorem 5.9. We shall start with the set-multilinear formula Φ of size n^c and reduce it to depth-4 via Theorem 3.3 for a bottom degree parameter t that shall be chosen shortly. It is fairly straightforward to observe that the depth reduction preserves multilinearity and set-multilinearity as well. Therefore we now have a set-multilinear expression of the form

$$f = T_1 + \cdots + T_{s'}$$

where $s' \leq s^{10(d/t)} = n^{10c(d/t)}$ and each $T_i = Q_{i1} \cdots Q_{ia_i}$ is a set-multilinear product. Let us fix one such term $T = Q_1 \cdots Q_a$ and we know that this is a set-multilinear product with $a \geq \frac{d \log t}{10t}$ non-trivial factors (by Theorem 3.3). Let $d_i = \deg(Q_i)$. By the sub-multiplicativity of tensor rank (Lemma 5.4) and the trivial upper bound (Lemma 5.5) we have

$$\begin{aligned}
\text{TensorRank}(T) &\leq n^{d_1-1} \cdots n^{d_a-1} \\
&= n^{d-a} \\
\implies \text{TensorRank}(f) &\leq s' \cdot n^{d-a} \qquad\qquad (\text{Lemma 5.3}) \\
&= \frac{n^d}{n^{a-10c(d/t)}}
\end{aligned}$$

Let us focus on the exponent of n in the denominator. Using the lower bound on a from Theorem 3.3, we get

$$a - 10c(d/t) \geq \frac{d \log t}{10t} - 10c\frac{d}{t} = \frac{d}{t}\left(\frac{\log t}{10} - 10c\right)$$

If we set $\frac{\log t}{10} = 11c$, then we get $a - 10c(d/t) \geq cd/t = d/\exp(c)$. Hence,

$$\text{TensorRank}(f) \leq \frac{n^d}{n^{d/\exp(c)}} \qquad\qquad\qquad \square$$

We would like to remark that, in spirit, a tensor rank upper bound for formulas is essentially a form of non-trivial reduction to set-multilinear depth three circuits. In this sense, this connection between tensor rank upper bound and reduction to depth four is perhaps not too un-natural.

Also, observe that if instead of a general set-multilinear formula, we had started with a constant depth set-multilinear formula, we would have obtained a slightly better upper bound (better dependence on c) on the tensor rank of f. The improvement essentially comes from the fact that the depth reduction for formulas with product depth Δ to $\Sigma\Pi\Sigma\Pi^{[t]}$ guarantees that the fan-in of product gates at the second level is at least $\Theta\left(\frac{d \cdot d^{1/\Delta}}{t}\right)$ (Section 4). We skip the details for the reader to verify.

5.2 An improvement

The result of Raz [Raz10] required $d = O(\log n / \log\log n)$ to be able to *set-multilinearize* the formula without much cost. However, with this alternate proof via the improved depth reduction, we can delay the set-multilinearization until a later stage and thus get the same upper bound on the tensor rank for much larger d, provided that the formula we started with was homogeneous.

Theorem 5.10. *Let f be a set-multilinear polynomial with respect to $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ that is computed by a homogeneous formula (not necessarily set-multilinear) Φ of size $s = n^c$. If d is* sub-polynomial *in n, that is $\log d = o(\log n)$, then*

$$\text{TensorRank}(f) \quad \leq \quad \frac{n^d}{n^{d/\exp(c)}}.$$

Proof. As earlier, we shall start with the formula Φ of size n^c and reduce it to a $\Sigma\Pi\Sigma\Pi^{[t]}$ formula Φ' of size $n^{10c(d/t)}$ for a t that shall be chosen shortly. Again, Φ' is a sum of terms of the form $T = Q_1 \cdots Q_a$, a product of $a \geq \frac{d \log t}{10t}$ non-trivial factors. The difference here is that this is not necessarily a set-multilinear product. Let $d_i = \deg(Q_i)$. Among the monomials in Q_i, there may be some that are divisible by two or more variables from some part $\mathbf{x}_j$ and others that are products of variables from distinct parts. For any $S \subset [d]$ let, $Q_{i,S}$ be the sum of monomials of Q_i that is a product of exactly only variable from each $\mathbf{x}_j$ for $j \in S$. Note that no monomials of Q_i that is a product of two or more variables from some $\mathbf{x}_j$ can contribute to a set-multilinear monomial of f. Thus, if $\text{SML}(T)$ is the restriction of T to just the set-multilinear monomials of T, then

$$\text{SML}(T) \quad = \quad \sum_{\substack{S_1 \sqcup \cdots \sqcup S_a = [d] \\ |S_i| = d_i}} Q_{1,S_1} \cdots Q_{a,S_a}$$

Here, $S_1, S_2, \ldots, S_a$ form a partition of the set $[d]$. We can observe that the tensor rank of each summand is upper bounded by $n^{d_1-1} n^{d_2-1} \cdots n^{d_a-1}$ and the number of summands is at most

$\binom{d}{d_1}\binom{d-d_1}{d_2}\cdots\binom{d-\sum_{i=1}^{a-1}d_i}{d_a}$). Using Lemma 5.3 and Lemma 5.4, we get the following.

$$
\begin{aligned}
\text{TensorRank}(\text{SML}(T)) &\leq \frac{n^d}{n^a}\cdot\binom{d}{d_1\ d_2\ \cdots\ d_a}\\
&\leq n^{d-a}\cdot d^d\\
&= n^{d-a}\cdot n^{d\log d/\log n}\\
\implies \quad \text{TensorRank}(f) &\leq n^d/n^{a-10c(d/t)-d\log d/\log n}
\end{aligned}
$$

Again, let us focus on the exponent in the denominator

$$
a - \frac{10c\cdot d}{t} - \frac{d\log d}{\log n} \geq \frac{d}{t}\left(\frac{\log t}{10} - 10c - \frac{t\log d}{\log n}\right)
$$

Once again we shall set $t = 2^{O(c)}$ so that $\frac{\log t}{10} - 10c = c$ and since $\log d = o(\log n)$ it follows that

$$
\frac{d}{t}\left(\frac{\log t}{10} - 10c - \frac{t\log d}{\log n}\right) \geq \frac{d}{\exp(c)}
$$

Hence,

$$
\text{TensorRank}(f) \quad\leq\quad \frac{n^d}{n^{d/\exp(c)}} \qquad\qquad \square
$$

Acknowledgements

References

[AJMV98] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-Commutative Arithmetic Circuits: Depth Reduction and Size Lower Bounds. *Theoretical Computer Science*, 209(1-2):47–86, 1998. Pre-print available at eccc:TR95-043.

[AV08] Manindra Agrawal and V. Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 67–75, 2008. Pre-print available at eccc:TR08-062.

[CJZ17] Matthias Christandl, Asger Kjærulff Jensen, and Jeroen Zuiddam. Tensor rank is not multiplicative under the tensor product. *CoRR*, abs/1705.09379, 2017.

[Hås90]   Johan Håstad. Tensor Rank is NP-complete. *J. Algorithms*, 11(4):644–654, December
           1990. Preliminary version in the *16th International Colloquium on Automata, Languages
           and Programming (ICALP 1989)*.

[HY11]   Pavel Hrubeš and Amir Yehudayoff. Homogeneous Formulas and Symmetric Polyno-
           mials. *Computational Complexity*, 20(3):559–578, 2011.

[Koi12]   Pascal Koiran. Arithmetic Circuits: The Chasm at Depth Four Gets Wider. *Theoretical
           Computer Science*, 448:56–65, 2012. Pre-print available at `arXiv:1006.4700`.

[Pam85]   P. Pamfilos. On the maximum rank of a tensor product. *Acta Mathematica Hungarica*,
           45(1):95–97, 1985.

[Raz10]   Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. In *Proceedings of the
           42nd Annual ACM Symposium on Theory of Computing (STOC 2010)*, pages 659–666, 2010.
           Pre-print available at `eccc:TR10-002`.

[Sap15]   Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity.
           Github survey: `https://github.com/dasarpmar/lowerbounds-survey/releases/`,
           2015.

[Tav15]   Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Com-
           put.*, 240:2–11, 2015. Preliminary version in the *38th Internationl Symposium on the Math-
           ematical Foundations of Computer Science (MFCS 2013)*.

[VSBR83]   Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Com-
           putation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–
           644, 1983. Preliminary version in the *6th Internationl Symposium on the Mathematical
           Foundations of Computer Science (MFCS 1981)*.