Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 176 (2015)

Some Lower Bound Results for Set-Multilinear Arithmetic Computations

V. Arvind S. Raja

Institute of Mathematical Sciences, Chennai.
{arvind,rajas}@imsc.res.in

March 2, 2016

Abstract

In this paper, we study the structure of set-multilinear arithmetic circuits and set-multilinear branching programs with the aim of showing lower bound results. We define some natural restrictions of these models for which we are able to show lower bound results. Some of our results extend existing lower bounds, while others are new and raise open questions. Specifically, our main results are the following:

- We observe that set-multilinear arithmetic circuits can be transformed into shallow set-multilinear circuits efficiently, following [VSBR83, RY08]. Hence, polynomial size set-multilinear circuits have quasi-polynomial size set-multilinear branching programs.
- We show that *k-narrow* set-multilinear ABPs computing the Permanent polynomial PER_n (or determinant DET_n) require $2^{\Omega(k)}$ size. As a consequence, we show that sum of r read-once oblivious ABPs computing PER_n requires size $2^{\Omega(\frac{n}{r})}$.
- We also show that set-multilinear branching programs are exponentially more powerful than *interval* multilinear circuits (where the index sets for each gate is restricted to be an interval w.r.t. some ordering), assuming the sum-of-squares conjecture. This further underlines the power of set-multilinear branching programs.
- Finally, we show exponential lower bounds for set-multilinear circuits with restrictions on the number of parse trees of monomials and prove exponential lower bounds results.

1 Introduction

Let $\mathbb{F}$ be a field and $X = X_1 \sqcup X_2 \sqcup \cdots \sqcup X_d$ be a partition of the variable set X. A *set-multilinear polynomial* $f \in \mathbb{F}[X]$ w.r.t. this partition is a homogeneous degree d multilinear polynomial such that every nonzero monomial of f has exactly one variable from $X_i, 1 \leq i \leq d$.

Both the Permanent polynomial PER_n and the Determinant polynomial DET_n are set-multilinear polynomials. The variable set is $X = \{x_{ij}\}_{1\leq i,j\leq n}$

ISSN 1433-8092

and the partition can be taken as the row-wise partition of the variable set. I.e. $X_i = \{x_{ij} \mid 1 \leq j \leq n\}$ for $1 \leq i \leq n$.

In this paper we will consider set-multilinear circuits and set-multilinear branching programs for computing set-multilinear polynomials. Set-multilinear circuits are well studied. The model of set-multilinear branching programs that we consider is more general than related notions of branching programs recently studied in the literature, like the read-once oblivious branching programs (ROABPs) [FS13].

A *set-multilinear arithmetic circuit* C computing f w.r.t. the above partition of X, is a directed acyclic graph such that each in-degree 0 node of the graph is labelled with an element from $X \cup \mathbb{F}$. Each internal node v of C is of indegree 2, and is either a + gate or × gate. With each gate v of we can associate a subset of indices $I_v \subseteq [d]$ and the polynomial C_v computed by the circuit at v is set-multilinear over the variable partition $\sqcup_{i \in I_v} X_i$. If v is a + gate then for each input u of v $I_u = I_v$, and v is a × gate with inputs v_1 and v_2 then $I_v = I_{v_1} \sqcup I_{v_2}$. Clearly, in a set-multilinear circuit every gate computes a set-multilinear polynomial (in a syntactic sense). The output gate of C is labeled by $[d]$ and computes the polynomial f. The *size* of C is the number of gates in it and its *depth* is the length of the longest path from an input gate to the output gate of C.

Additionally, if every gate in a set-multilinear circuit C has outdegree bounded by 1 then C is a *set-multilinear formula*.

A *set-multilinear algebraic branching program* smABP is a layered directed acyclic graph (DAG) with one in-degree zero vertex s and one out-degree zero vertex t. The vertices of the graph are partitioned into layers $0, 1, \ldots, d$, and edges go only from layer i to $i+1$ for each i. The source is the only vertex at level 0 and the sink is the only vertex at level d. We can associate an index set $I_v \subseteq [d]$ with each node v in the smABP, and the polynomial computed at v is set-multilinear w.r.t. the partition $\sqcup_{i \in I_v} X_i$. For any edge (u, v) in the branching program labeled by a homogeneous linear form ℓ, we have $I_v = I_u \sqcup \{i\}$ for some $i \in [d]$, and ℓ is a linear form over variables X_i. The size of the ABP is the number of vertices.

For any s-to-t directed path $\gamma = e_1, e_2, \ldots, e_d$, where e_i is the edge from level $i-1$ to level i, let ℓ_i denote the linear form labeling edge e_i. Let $f_\gamma = \ell_1 \cdot \ell_2 \cdots \ell_d$ be the product of the linear forms in that order. Then the ABP computes the set-multilinear degree d polynomial:

$$f = \sum_{\gamma \in \mathcal{P}} f_\gamma,$$

where $\mathcal{P}$ is the set of all directed paths from s to t.

Remark 1. *Showing a superpolynomial lower bound for set-multilinear circuits and even for set-multilinear ABPs for computing the Permanent polynomial is an open problem. In this paper we discuss some restricted versions of set-multilinear branching programs and show lower bounds.*

Summary of results

To begin with we observe that any set-multilinear arithmetic circuit of size s can be efficiently transformed into an $O(\log s)$ depth set-multilinear circuit with unbounded fanin $+$ gates and fanin 2 $\times$ gates of size polynomial in s. The proof is identical to the depth-reduction results of [VSBR83, RY08] for more general commutative circuits. The only extra point is that set-multilinearity is preserved in the process. As a result, size s set-multilinear circuits have $s^{O(\log s)}$ size set-multilinear branching programs. This is stated in Section 2. In order to keep the paper self-contained we present a proof, following [RY08], in the appendix.

In Section 3 we consider *narrow* set-multilinear branching programs: The *typewidth* of a set-multilinear ABP at layer k is the number of distinct index types of the ABP in its k^{th} layer. Read-once oblivious ABPs (ROABPs) considered in [FS13] are the set-multilinear ABPs of typewidth 1 in each layer. Thus, it is natural to investigate lower bounds for set-multilinear ABPs of restricted typewidth in order to extend known lower bounds [Nis91] for ROABPs. We say that a set-multilinear ABP computing a degree d polynomial is *k-narrow* if either in layer $d-k$ or layer k the typewidth is bounded by $\frac{d}{2k}$. We show that k-narrow set-multilinear ABPs for PER_n require $2^{\Omega(k)}$ size. The proof is based on standard techniques based on the rank of the partial derivative matrix. As a consequence, it follows that the sum of r ROABPs computing PER_n require size $2^{\Omega(\frac{n}{r})}$. Similarly, a set-multilinear ABP for PER_n, whose typewidth in all layers is bounded by $n^{1-\epsilon}$, requires size $2^{n^{\epsilon}}$. While these observations generalize the lower bound for ROABPs, it appears difficult to prove superpolynomial lower bounds for ABPs of larger typewidth (even $O(n)$). A lower bound result for general set-multilinear ABPs would imply, for instance, that the Permanent requires superpolynomial size noncommutative arithmetic circuits, which is an open problem for over two decades. We present these observations in Section 3.

In Section 4, we show that set-multilinear branching programs are exponentially more powerful than *interval* multilinear circuits (where the index sets for each gate is restricted to be an interval w.r.t. some ordering), assuming the sum-of-square conjecture [HWY10]. This further underlines the power of general set-multilinear branching programs.

Finally, in Section 5 we investigate lower bounds for a different generalization of ROABPs based on the structure of parse trees of monomials. We can define parse tree types for a set-multilinear circuit which is a binary tree with index types labeling all its nodes. We note in an ROABP all monomials have the same parse tree type (a single path of length d). Thus, a natural generalization is to consider set-multilinear ABPs (and circuits) with restrictions on parse trees of monomials. It turns out that if a set-multilinear circuit for PER_n has only r many parse tree types *in all*, then we can use the results of Section 3 to show a $2^{\Omega(\frac{n}{r\log n})}$ lower bound on its size. Another restriction are set-multilinear circuits such that each monomial has at most one parse tree type (but the total number of parse tree types in the circuit is unbounded). For such circuits too we can prove exponential lower bounds.

Our lower bound proofs are applications of the well-known partial derivative method.

2 Depth Reduction of Set-Multilinear Circuits

We follow the standard method of depth reduction of commutative arithmetic circuits [VSBR83], and use the exposition from Shpilka and Yehudayoff's survey article [SY10]. The general depth reduction was adapted to syntactic multilinear circuits by Raz and Yehudayoff [RY08]. Our additional observation essentially is that the depth reduction procedure can be carried out while preserving set-multilinearity as well.

Given a commutative set-multilinear circuit C of size s computing a set-multilinear polynomial f of degree d in the input variable $X = X_1 \sqcup \ldots \sqcup X_d$, we show that there is another circuit C' of size $poly(s)$ and depth $O(\log d \log s)$ computing f.

Theorem 2. *Let Φ be a set-multilinear arithmetic circuit of size s and degree d over the field $\mathbb{F}$ and over the variable set X, partitioned as $X = X_1 \sqcup \ldots \sqcup X_d$, computing a polynomial $f \in \mathbb{F}[X]$. Then we can efficiently compute from Φ a set-multilinear arithmetic circuit Ψ, with multiplication gates of fanin 2 and unbounded fanin $+$ gates, which is of size $O(s^3)$ and depth $O(\log d)$ computing the polynomial f.*

The proof of this theorem follows the same steps as in the depth reduction results in [VSBR83, RY08]. The only additional point we observe is that set-multilinearity is preserved by the construction. However, we have included a proof in the appendix to keep the paper self-contained.

Remark 3. *We note that the size of the depth-reduced circuit for general multilinear circuits [RY08] turns out to be $O(s^3d^6)$ because of the homogenization step, which is not required for set-multilinear circuits.*

Set-Multilinear Circuits to ABPs

Theorem 4. *Given a set-multilinear arithmetic circuit of size s and degree d over the field $\mathbb{F}$ and over the variable set $X = \sqcup_{i=1}^{d} X_i$, computing $f \in \mathbb{F}[X]$, we can transform it, in time $s^{O(\log d)}$, into a set-multilinear ABP of size $(sd)^{O(\log d)}$ that computes f.*

Proof. The proof of this theorem is fairly straightforward consequence of the depth reduction result (Theorem 2 in the previous section). By Theorem 2 we can assume to have computed a set-multilinear circuit Ψ of size $O(s^3)$ and depth $O(\log d)$ for computing f. By a standard bottom-up procedure we can transform the circuit Ψ into a formula F: at a gate g with fanout t we make t copies of the circuit computed at g. The resulting circuit is a formula of size $(sd)^{O(\log d)}$, because at every level of the circuit there is a factor of s increase in the size (as s bounds the fanout of all gates). The formula F thus constructed from C is clearly also homogeneous, set-multilinear, and of depth $O(\log d)$. The formula F is also semi-unbounded: the product gates are fanin 2 and plus gates have unbounded fanin.

Formula F could have subformulas that computes scalars. We perform the following transformation to remove such subformulas. Suppose g is a gate in F that evaluates to a scalar α, and g is input to gate g' in F. Then we remove the subformula at g and label the outgoing edge of g' by α. The transformed set-multilinear formula F' has only variables at the input gates and scalars labeling edges of the formula interpreted as follows: suppose g is a gate with g' as an input gate such that the edge from g' to g is labeled by scalar α. Then the contribution of gate g' to g is the polynomial αP, where P is the polynomial computed at g'.

Finally, we can apply a standard transformation (for e.g., see [Nis91]) to convert the formula F' into a homogeneous algebraic branching program (ABP). It is a bottom-up construction of the ABP: at a $+$ gate we can do a "parallel composition" of the input ABPs to simulate the $+$ gate. At a $\times$ gate it is a sequential composition of the two ABPs. Since the formula F is set-multilinear, the resulting ABP is also easily seen to be a set-multilinear ABP. ■

3 A Lower Bound Result for Set-Multilinear ABPs

As we have shown in Theorem 2, we can simulate set-multilinear circuits of size s and degree d using set-multilinear ABPs of size $s^{O(\log d)}$. Thus, proving even a lower bound of $n^{\omega(\log n)}$ for set-multilinear ABPs computing the $n \times n$ Permanent polynomial PER_n would imply superpolynomial lower bounds for general set-multilinear circuits computing PER_n which is a long-standing open problem.

However, in this section we show a lower bound result for set-multilinear ABPs with restricted *type width*, a notion that we now formally introduce.

Let P be a set-multilinear ABP computing a polynomial $f \in \mathbb{F}[X]$ of degree d with variable set $X = \sqcup_{i=1}^{d} X_i$. By definition, the ABP P is a layered directed acyclic graph with layers numbered $0, 1, \dots, d$. Each node v in layer k of the ABP is labeled by an index set $I_v \subseteq [d]$, and a degree k set-multilinear polynomial f_v over variables $\sqcup_{i \in I_v} X_i$ is computed at v by the ABP. We refer to I_v as the *type* of node v. The *type width* of the ABP at layer k is the number $\mathrm{tw}(k)$ of *different* types labeling nodes at layer k of the ABP.

The notion of type width is motivated by the fact that read-once oblivious ABPs (ROABPs defined in [FS13]) have type width 1 (as noted in the following proposition).

Proposition 5. *Suppose P is a set-multilinear ABP computing a polynomial $f \in \mathbb{F}[X]$ of degree d with variable set $X = \sqcup_{i=1}^{d} X_i$ such that the type-width of P is 1 at each layer. Then P is in fact an ROABP which is defined by a suitable permutation on the index set $[d]$.*

Proof. As each layer of P has type width one, the list of type $I_0 = \emptyset \subset I_1 \subset \cdots \subset I_d$ gives an ordering of the index set, where the i^{th} index in the ordering is $I_i \setminus I_{i-1}$. W.r.t. this ordering clearly P is an ROABP. ■

It is well-known that Nisan's rank argument [Nis91] (originally used for lower bounding noncommutative ABP size) also yields exponential lower bounds for any ROABP computing PER_n. In particular, it implies an exponential lower bound for set-multilinear ABPs of type-width 1. This suggests that a natural first step to showing lower bounds for general set-multilinear ABPs/circuits is to first deal with set-multilinear ABPs with restrictions on its type width.

3.1 Lower bounds for narrow set-multilinear ABPs

Definition 6. *A set-multilinear ABP computing a degree d polynomial in $\mathbb{F}[X]$ such that $X = \sqcup_{i=1}^{d} X_i$ is said to be k-*narrow*, for $1 \leq k \leq d$, if*

$$\min\{k\mathrm{tw}(k), (d-k)\mathrm{tw}(k)\} \leq d/2.$$

For example, ROABPs are a subclass of set-multilinear ABPs that are k-narrow for every k. As another example, we note that the sum of ℓ ROABPs is $\frac{d}{2\ell}$-narrow.

Theorem 7. *Any k-narrow set-multilinear ABP computing the permanent polynomial PER_n requires size 2^k.*[1]

Proof. Let P be a k-narrow set-multilinear ABP computing PER_n, and suppose $k\mathrm{tw}(k) \leq n/2$ (we note that if $(n-k)\mathrm{tw}(k) \leq n/2$ the proof proceeds analogously by reversing the roles of the source and sink nodes of the ABP). In the proof we shall assume that n is even in order to avoid floors and ceils.

Let V_k denote the set of nodes in the k^{th} layer of P. For each node $v \in V_k$ let I_v denote its index set. As $|I_v| = k$ and P is k-narrow, we have

$$|\bigcup_{v\in V_k} I_v| \leq k\mathrm{tw}(k) \leq n/2.$$

We choose and fix a size $n/2$ subset A of $[n]$ such that $\bigcup_{v\in V_k} I_v \subseteq A$.

For any set-multilinear polynomial $f \in \mathbb{F}[X]$ w.r.t. the partition $X = \sqcup_{i=1}^{n} X_i$, where $X_i = \{X_{ij} \mid 1 \leq j \leq n\}$, we can define the matrix M_f whose rows are indexed by degree $n/2$ set-multilinear monomials m and columns by degree $n/2$ set-multilinear monomials m' such that m is a monomial in variables $\sqcup_{i\in A} X_i$ and m' in variables $\sqcup_{i\notin A} X_i$ and

$$M_f(m, m') = f(mm'), \tag{1}$$

where $f(mm')$ denote the coefficient of monomial mm' in polynomial f.

For each node $v \in V_k$ let f_v and g_v denote the polynomials computed by P between start node s and sink node v, and start node v and sink node t, respectively. Since P computes PER_n, it follows that

$$M_{\mathrm{PER}_n} = \sum_{v\in V_k} M_{f_v g_v}.$$

Let $A = \{i_1, i_2, \ldots, i_{n/2}\}$ and $\overline{A} = \{j_1, j_2, \ldots, j_{n/2}\}$ be the indices listed in, say, increasing order. We set some variables of PER_n to zero: We rename the

[1] The same lower bound proof will work for the Determinant polynomial DET_n.

rows and columns of the $n \times n$ matrix X_{ij} as $i_1, j_1, i_2, j_2, \ldots, i_{n/2}, j_{n/2}$. Along the principal diagonal we now have $n/2$ many 2×2 matrices, where the r^{th} such matrix is indexed by i_r, j_r in both rows and columns. We retain only these $4r = 2n$ many variables in PER_n and set all other variables X_{ij} to 0. Let the resulting polynomial in these $2n$ variables be denoted $\widehat{\mathrm{PER}}_n$. Notice that $\widehat{\mathrm{PER}}_n$ is set-multilinear of degree n with each set $X'_i \subset X_i$ in partition having exactly two variables in it. Furthermore, let $\hat{f_v}$ and $\hat{g_v}$ be polynomials obtained from f_v and g_v by the same substitution for each $v \in V_k$. We clearly have

$$M_{\widehat{\mathrm{PER}}_n} = \sum_{v \in V_k} M_{\hat{f_v}\hat{g_v}}.$$

Clearly,

$$rank(M_{\widehat{\mathrm{PER}}_n}) \leq \sum_{v \in V_k} rank(M_{\hat{f_v}\hat{g_v}}). \tag{2}$$

The following two claims immediately yield the claimed lower bound of 2^k on the size of the ABP P.

Claim 8. *$rank(M_{\widehat{\mathrm{PER}}_n}) = 2^{n/2}$.*

Proof of Claim. The degree $n/2$ monomials labeling the rows of matrix $M_{\widehat{\mathrm{PER}}_n}$ are of the form $m = \prod_{t=1}^{r} X_{i_t a_t}$, where $a_t \in \{i_t, j_t\}$. Likewise, the degree $n/2$ monomials labeling the columns of matrix $M_{\widehat{\mathrm{PER}}_n}$ are of the form $m' = \prod_{t=1}^{r} X_{j_t a_t}$, where $a_t \in \{i_t, j_t\}$. Clearly, for each such m there is a unique m' such that their product mm' is a nonzero monomial of $\widehat{\mathrm{PER}}_n$. Thus, the matrix $M_{\widehat{\mathrm{PER}}_n}$ is a permutation matrix and hence it has rank exactly $2^{n/2}$.

Claim 9. *For each $v \in V_k$ we have $rank(M_{\hat{f_v}\hat{g_v}}) \leq 2^{n/2-k}$.*

Proof of Claim. It suffices to show that we can write matrix $M_{\hat{f_v}\hat{g_v}}$ as a sum of $2^{n/2-k}$ rank 1 matrices. To this end, let $\hat{m}$ be a set-multilinear monomial of degree $n/2 - k$ whose variables come from $\sqcup_{i \in A \setminus I_v} X_i$ (where we recall that I_v is the index set of node v in the ABP). Let $M^{(\hat{m})}$ denote the $2^{n/2} \times 2^{n/2}$ matrix obtained from $M_{\hat{f_v}\hat{g_v}}$ as follows: for each degree k set-multilinear monomial m over $\sqcup_{i \in I_v} X_i$, the row labeled $m\hat{m}$ of $M_{\hat{f_v}\hat{g_v}}$ is the same for $M^{(\hat{m})}$. All other rows of $M^{(\hat{m})}$ are zero. Clearly,

$$M_{\hat{f_v}\hat{g_v}} = \sum_{\hat{m}} M^{(\hat{m})}.$$

Furthermore, the rank of $M^{\hat{m}}$ is at most 1 because it can be expressed as the product of a $2^{n/2} \times 1$ matrix and a $1 \times 2^{n/2}$ matrix (using the coefficient vectors of the polynomials $\hat{f_v}$ and $\hat{g_v}$).

Clearly, the above claims combined with Equation (2) imply that $|V_k| \geq 2^k$ which implies that the size of P is lower bounded by 2^k. ■

As a consequence of Theorem 7 we immediately obtain the following lower bound on the size of a sum of r many ROABPs for computing the permanent.

Corollary 10.

- *If P is a set-multilinear ABP computing* PER_n *such that the type width of every layer is bounded by r then the size of P is $2^{\Omega(n/r)}$. A special case is the next part.*

- *Let $P_i, 1 \leq i \leq r$, be ROABPs such that $\sum_{i=1}^{r} P_i$ is the permanent polynomial* PER_n *(or the determinant polynomial* DET_n*). Then at least one of the P_i is of size $2^{\Omega(n/r)}$.*

Proof. The first part clearly implies that the type width at layer $\frac{n}{2r}$ is bounded by r and we can apply the above theorem. For the second part, it suffices to observe that the sum of r many ROABPs is an smABP whose type width at each layer is bounded by r. ■

Thus, if $r = O(\frac{n}{\log^2 n})$, we get superpolynomial ($n^{\Omega(\log n)}$) lower bound for sum of $O(\frac{n}{\log^2 n})$-many ROABPs computing PER_n or DET_n.

Remark 11. *We note that there is a polynomial-time (white-box) identity testing algorithm for the sum of a* constant *number of ROABPs [GKST15]. Their black-box PIT is quasi-polynomial time.*

It remains an interesting problem to prove a superpolynomial lower bound for the sum of $\mathrm{poly}(n)$ *many (or even $O(n)$ many) ROABPs computing* PER_n.

4 Interval multilinear circuits and ABPs

For variable partition $X = \sqcup_{i=1}^{d} X_i$ let $f \in \mathbb{F}[X]$ be a set-multilinear polynomial.

For a permutation $\sigma \in S_d$, a *σ-interval multilinear circuit C* for computing f is a special kind of set-multilinear arithmetic circuit: for every gate of the circuit the corresponding index set is a *σ-interval* $\{\sigma(i), \sigma(i+1), \ldots, \sigma(j)\}$, $1 \leq i \leq j \leq d$.

Similarly, a *σ-interval multilinear ABP* is a set-multilinear ABP such that the index set associated to every node is some σ-interval.

The aim of the present section is to compare the computational power of interval multilinear circuits with general set-multilinear circuits. Clearly, σ-interval multilinear circuits are restricted by the ordering. In essence, σ-interval multilinear circuits are restricted to compute like noncommutative circuits (with respect to the ordering prescribed by σ). This property needs to be exploited to prove the separations. We show the following result:

> Assuming the sum-of-squares conjecture [HWY10], we show that there are set-multilinear polynomials $f \in \mathbb{R}[X]$ with monotone set-multilinear circuits (even ABPs) of size linear in d, but require $2^{\Omega(d)}$ size σ-interval multilinear circuits for every $\sigma \in S_d$.

A new aspect is that the polynomial we construct is only partially explicit. We use the probabilistic method to pick certain parameters that define the polynomial.

The polynomial construction

Let $X = \sqcup_{i=1}^{2d} X_i$ be the variable set, where

$$X_i = \{x_{0,i}, x_{1,i}\}, 1 \leq i \leq 2d.$$

For every binary string $b \in \{0,1\}^d$ we define the monomials:

$$w_b = \prod_{i=1}^{d} x_{b_i,i} \text{ and } w'_b = \prod_{i=1}^{d} x_{b_i,d+i}.$$

We define the set-multilinear polynomial

$$P = \sum_{\bar{b}\in\{0,1\}^d} w_b w'_b.$$

For any permutation $\sigma \in S_{2d}$ permuting the indices in $[2d]$, we define monomials:

$$\sigma(w_b) = \prod_{i=1}^{d} x_{b_i,\sigma(i)} \text{ and } \sigma(w'_b) = \prod_{i=1}^{d} x_{b_i,\sigma(d+i)},$$

and the corresponding polynomial $\sigma(P)$

$$\sigma(P) = \sum_{\bar{b}\in\{0,1\}^d} \sigma(w_b)\sigma(w'_b).$$

Definition 12. *In the polynomial $\sigma(P)$ we refer to the indices $\sigma(j)$ and $\sigma(d+j)$ as a* matched pair of indices*, since in the monomial $\sigma(w_b)\sigma(w'_b)$ it is required that the variables $x_{b_j,\sigma(j)}$ and $x_{b_j,\sigma(d+j)}$ have the same first index b_j.*

For $\sigma = id$, the matched pairs are $(j, d+j)$ for $1 \leq j \leq d$.

Lemma 13.

- *The set-multilinear polynomial $\sigma(P)$ can be computed by a monotone set-multilinear ABP of size $O(d)$.*
- *For any $\sigma_1, \sigma_2, \ldots, \sigma_s \in S_{2d}$ the polynomial $\sum_{j=1}^{s} \sigma_i(P)$ can be computed by a monotone set-multilinear ABP of size $O(sd)$.*

Proof. As

$$\sigma(P) = \prod_{i=1}^{d} (x_{0,\sigma(i)} x_{0,\sigma(d+i)} + x_{1,\sigma(i)} x_{1,\sigma(d+i)}),$$

clearly $\sigma(P)$ has an $O(d)$ monotone set-multilinear formula. Hence, it has a monotone set-multilinear ABP of size $O(d)$. The second part of the lemma is an immediate consequence. ■

We will show that there exist a set of d permutations $\sigma_1, \sigma_2, \ldots, \sigma_d \in S_{2d}$ with the following property: for any permutation $\tau \in S_{2d}$ there is a σ_i from this set such that any τ-interval multilinear circuit that computes $\sigma_i(P)$ requires size $2^{\Omega(d)}$.

Note that, in contrast, given a $\sigma(P)$ there is always a permutation $\tau \in S_{2d}$ such that $\sigma(P)$ can be computed by a small τ-interval multilinear circuit. Indeed, the ABP computing $\sigma(P)$ described in Lemma 13 is an interval-multilinear ABP w.r.t. the ordering $\sigma(1), \sigma(d+1), \sigma(2), \sigma(d+2), \ldots, \sigma(d), \sigma(2d)$.

Interval multilinear circuits and noncommutative circuits

We first observe that a τ-interval multilinear circuit computing a set-multilinear polynomial in $\mathbb{F}[X]$, $X = \sqcup_{i=1}^{d} X_i$, is essentially like a noncommutative circuit computing a noncommutative polynomial over the variables X, whose monomials can be considered as words of the form $x_{i_1} x_{i_2} \ldots x_{i_d}$, where $x_{i_j} \in X_{\tau(j)}$ for $1 \leq j \leq d$.

In [HWY10], Hrubes et al have related the well-known sum-of-squares (in short, SOS) conjecture (also see [Sha00]) to lower bounds for *noncommutative arithmetic circuits*. Our results in this section are based on their work. We recall the conjecture.

The sum-of-squares (SOS) conjecture: Consider the question of expressing the biquadratic polynomial

$$SOS_k(x_1, \ldots, x_k, y_1, \ldots, x_k) = (\sum_{i \in [k]} x_i^2)(\sum_{i \in [k]} y_i^2)$$

as a sum of squares $(\sum_{i \in [s]} f_i^2)$ for the least possible s, where each f_i is a homogeneous bilinear polynomial. The conjecture states that $s = \Omega(k^{1+\epsilon})$ over the field of complex numbers $\mathbb{C}$ (or the algebraic closure of any field $\mathbb{F}$ such that $char(\mathbb{F}) \neq 2$).

The following lower bound is shown in [HWY10] assuming the SOS conjecture.

Theorem 14. [HWY10] *Assuming the SOS conjecture over field $\mathbb{F}$, any noncommutative circuit computing the polynomial $ID = \sum_{w \in \{x_0, x_1\}^d} ww$ in* noncommuting variables *x_0 and x_1 requires size $2^{\Omega(d)}$.*

The above theorem implies the following conditional lower bound for interval multilinear circuits.

Corollary 15. *Assuming the SOS conjecture, for any $\sigma \in S_{2d}$ a σ-interval multilinear circuit computing the set-multilinear polynomial $\sigma(P)$ requires size $2^{\Omega(d)}$.*

Proof. Let C be a size s σ-interval multilinear circuit for $\sigma(P)$. The SOS conjecture combined with the following claim clearly yields the corollary.

Claim 16. *From C we can obtain a size s homogeneous noncommutative circuit for the noncommutative polynomial $\sum_{w \in \{x_0, x_1\}^d} ww$.*

Proof of Claim. By definition, the index sets of every gate in circuit C is a σ-interval. We construct a noncommutative circuit C' of size s from C as explained below:

1. Replace variable $X_{b,\sigma(i)}$ by x_b for $1 \leq i \leq 2d$ and $b \in \{0,1\}$ at the inputs.
2. Interpret all × gates as noncommutative product gate with inputs ordered left to right as per their respective interval locations.

By construction, each gate g in C' has an interval I_g in $[1-(2d)]$ associated with it. In particular, if a gate g of degree k in C has interval $\{\sigma(i), \sigma(i+1), \cdots, \sigma(i+k-1)\}$ for some $i \leq d-k$, then the corresponding gate g in C' has interval $I_g = \{i, i+1, \cdots, i+k-1\}$. We claim that the output gate of C' computes the polynomial $\sum_{w\in\{x_0,x_1\}^d} ww$. We prove this by an inductive argument on the circuit structure.

In particular, we show that gate g in C computes a monomial $\prod_j x_{b_j,\sigma(j)}$ with coefficient $\alpha \in \mathbb{F}$ iff the corresponding gate g in C' computes monomial $\prod_j x_{b_j}$ with coefficient $\alpha \in \mathbb{F}$. Let $\text{Coeff}_g(m')$ denote the coefficient of monomial m' in the polynomial computed by g in C'. Similarly, let $\text{Coeff}_g(m)$ denote the coefficient of monomial m in the polynomial computed by g in C.

- For the base case, suppose gate g in C' is an input gate with label x_{b_i} and interval $[i]$, then clearly gate g in C is an input gate with variable $x_{b_i,\sigma(i)}$.
- Suppose gate g in C' is a $+$ gate of degree k, where $g = g_1 + g_2$, and all three gates have interval $I = [i, i+1, \ldots, i+k-1]$ labeling them. Further, suppose $m' = x_{b_i}x_{b_{i+1}}\cdots x_{b_{i+k-1}}$ is a monomial computed at gate g with coefficient $\alpha_{m'} \in \mathbb{F}$. Then $\alpha_{m'} = \text{Coeff}_{g_1}(m') + \text{Coeff}_{g_2}(m')$. By induction hypothesis, $\text{Coeff}_{g_1}(m')$, $\text{Coeff}_{g_2}(m')$ in C' equal $\text{Coeff}_{g_1}(m)$, $\text{Coeff}_{g_2}(m)$ in C respectively, where monomial $m = \prod_{j=i}^{i+k-1} x_{b_j,\sigma(j)}$. Thus, $\text{Coeff}_g(m') = \text{Coeff}_g(m)$.
- Suppose g is a × gate in C' of degree k. Let $g = g_1 \times g_2$, and $\deg(g_1) = k_1$. Let $I = I_1 \uplus I_2$ be the intervals corresponding to gates g, g_1, and g_2, and $I = [i, i+1, \ldots, i+k-1]$. Let $m' = \prod_{j=i}^{i+k-1} x_{b_j}$ be a monomial with coefficient $\alpha_{m'} \in \mathbb{F}$ computed at gate g'. Then we can write $\alpha_{m'} = \text{Coeff}_{g_1}(m'_1) \times \text{Coeff}_{g_2}(m'_2)$. By induction hypothesis, the coefficients $\text{Coeff}_{g_1}(m'_1)$ and $\text{Coeff}_{g_2}(m'_2)$ in C' are the same as the coefficients $\text{Coeff}_{g_1}(m_1)$, $\text{Coeff}_{g_2}(m_2)$ in C, respectively. Here, notice that the monomials m_1 and m_2 are uniquely defined from m'_1 and m'_2 and the intervals I_1 and I_2. It follows that $\text{Coeff}_g(m')$ in C' equals $\text{Coeff}_g(m)$ in C.

■

Remark 17. *In [HWY10], the connection between the SOS conjecture and lower bounds for the noncommutative polynomial $\sum_{w\in\{x_0,x_1\}^d} ww$ is made by considering first writing it as $\sum_{w_1,w_2\in\{x_0,x_1\}^{d/2}} w_1w_2w_1w_2$. Then the noncommutative degree-$d/2$ monomials w_1 and w_2 are treated as single commuting variables which allows the polynomial to be written as a product of two quadratics*

$(\sum w_1^2)(\sum w_2^2)$. Here w_1 and w_2 run over two sets of $2^{d/2}$ distinct variables each.

Now, the SOS conjecture applied to this biquadratic polynomial implies that writing $(\sum w_1^2)(\sum w_2^2)$ as $\sum_{i=1}^t f_i^2$, for bilinear forms f_i, implies $t = \Omega(2^{d/2(1+\epsilon)})$ for a constant $\epsilon > 0$. In [HWY10, Corollary 5.4] it is shown that any size s noncommutative circuit for $\sum_{w\in\{x_0,x_1\}^d} ww$ can be transformed into a sum of squares $\sum_{i=1}^t f_i^2$ such that $t = O(d^3 s 2^{d/2})$ which implies the lower bound of $2^{\Omega(d)}$ on the circuit size s.

Motivated by the connection described in the above remark we obtain the following easy lemma.

Lemma 18. *For even d, partition $[2d]$ into four intervals of size $d/2$ each: $I_1 = [1,2,\ldots,d/2]$, $I_2 = [d/2+1,\ldots,d]$, $I_3 = [d+1,\ldots,3d/2]$, and $I_4 = [3d/2+1\ldots 2d]$. Let $\sigma \in S_{2d}$ be any permutation such that $\sigma(I_j) = I_j, 1 \le j \le 4$. Then, assuming the SOS conjecture, any id-interval multilinear circuit computing the polynomial $\sigma(P)$ requires size $2^{\Omega(d)}$.*

Proof. By definition, $\sigma(P) = \sum_{b\in\{0,1\}^d} \sigma(w_b)\sigma(w'_b)$. Since σ stabilizes each $I_j, 1 \le j \le 4$, we can write $\sigma(w_b) = w_1 w_2$ and $\sigma(w'_b) = w'_1 w'_2$, where the matched pairs are between w_1 and w'_1, and between w_2 and w'_2, respectively. Now, by substituting the *same* variable for each matched pair, we obtain the polynomial $(\sum w_1^2)(\sum w_2^2)$. As explained in Remark 17, we can treat this as a biquadratic polynomial, where w_1 and w_2 are single variables that run over variable sets of size $2^{d/2}$ each. The proof argument in [HWY10] can now be applied to yield that any *id*-interval circuit size computing $\sigma(P)$ has size $2^{\Omega(d)}$, assuming the SOS conjecture for the biquadratic polynomial $(\sum w_1^2)(\sum w_2^2)$. ■

We will use the probabilistic method to show the existence of the set of permutations $\sigma_i, 1 \le i \le d$ in S_{2d}, such that for each $\tau \in S_{2d}$ there is some $\sigma_i(P), i \in [d]$ that requires size $2^{\Omega(d)}$ τ-interval multilinear circuits. We will require the following concentration bound.

Lemma 19. [DP09, Theorem 5.3, page 68] *Let $X_1, \cdots, X_n$ be any n random variables and let f be a function of $X_1, X_2 \ldots, X_n$. Suppose for each $i \in [n]$ there is $c_i \ge 0$ such that*

$$|\mathbb{E}[f|X_1,\cdots,X_i] - \mathbb{E}[f|X_1,\cdots,X_{i-1}]| \le c_i.$$

Then for any $t > 0$, we have the bound $\text{Prob}[f < \mathbb{E}[f] - t] \le exp(-\frac{2t^2}{c})$*, where $c = \sum_{i\in[n]} c_i^2$.*

Lemma 20. *Let $\sigma \in S_{2d}$ be a permutation picked uniformly at random. For any $\tau \in S_{2d}$, the probability that $\sigma(P)$ is computable by a τ-interval multilinear circuit of size $2^{o(d)}$ is bounded by $e^{-\Omega(d)}$, assuming the SOS conjecture.*

Proof. In the polynomial $P = \sum_{b\in\{0,1\}^d} w_b w'_b$ the *matched pairs*, as defined earlier, are the index pairs $(i, d+i), 1 \le i \le d$. As in Lemma 18, we partition the index set $[2d]$ into four consecutive $d/2$-size intervals $I_1 = [1, \ldots, d/2]$, $I_2 = [d/2+1, \ldots, d]$, $I_3 = [d+1, \ldots, 3d/2]$, and $I_4 = [3d/2+1, \ldots, 2d]$. Note that $d/2$ of the matched pairs are between I_1 and I_3 and the remaining $d/2$ are between I_2 and I_4. Consider the following two subsets of matched pairs of size $d/8$ each:

$$\begin{aligned} E_1 &= \{(i, d+i) \mid 1 \leq i \leq d/8\} \\ E_2 &= \{(d/2+i, 3d/2+i) \mid 1 \leq i \leq d/8\}. \end{aligned}$$

The pairs in E_1 are between I_1 and I_3 and pairs in E_2 are between I_2 and I_4. Let $\sigma \in S_{2d}$ be a permutation picked uniformly at random. We say $(i, d+i) \in E_1$ is *good* if $\sigma(i) \in I_1$ and $\sigma(d+i) \in I_3$. Similarly, $(d/2+i, 3d/2+i) \in E_2$ is called *good* if $\sigma(d/2+i) \in I_2$ and $\sigma(3d/2+i) \in I_4$. Let X_i, $1 \leq i \leq d/8$, be indicator random variables which take the value 1 iff the pair $(i, d+i) \in E_1$ is good. Similarly, define indicator random variables X_i' corresponding to pairs $((d/2+i, 3d/2+i) \in E_2, 1 \leq i \leq d/8$.

For each $i \in [d/8]$, $Y_i \in \{X_i, X_i'\}$ we have

$$\text{Prob}_{\sigma \in S_{2d}}[Y_i = 1] \geq (\frac{(d/2 - d/8)}{2d})^2 = 9/256 > 1/64.$$

Let $f = \sum_{i=1}^{d/8} X_i$ and $f' = \sum_{i=1}^{d/8} X_i'$. Clearly, we have

$$\text{E}[f] \geq \frac{d}{8 \cdot 64} = \frac{d}{512}.$$

Furthermore, we also have for each $i : 1 \leq i \leq d/8$

$$|\mathbb{E}[f | X_1, X_2, \ldots, X_i] - \mathbb{E}[f | X_1, X_2, \ldots, X_{i-1}]| \leq 1$$

Applying Lemma 19, with $t = d/1024$ we deduce that

$$\text{Prob}_{\sigma \in S_{2d}}[f < \frac{d}{1024}] \quad \leq \quad e^{-\alpha d},$$

where $\alpha > 0$ is some constant independent of d. Similarly, we also have

$$\text{Prob}_{\sigma \in S_{2d}}[f' < \frac{d}{1024}] \quad \leq \quad e^{-\alpha d},$$

Combining the two bounds yields

$$\text{Prob}_{\sigma \in S_{2d}}[f \geq \frac{d}{1024} \text{ and } f' \geq \frac{d}{1024}] \geq 1 - 2e^{-\alpha d}.$$

Thus, with probability $1 - 2e^{-\alpha d}$ there are $d/1024$ pairs $(\sigma(i), \sigma(d+i))$ such that $\sigma(i) \in I_1$ and $\sigma(d+i) \in I_3$, and there are $d/1024$ pairs $(\sigma(d/2+i), \sigma(3d/2+$

$i)$) such that $\sigma(d/2+i) \in I_2$ and $\sigma(3d/2+i) \in I_4$. If we set all other variables in the polynomial $\sigma(P)$ to 1, we can apply Lemma 18 to the resulting polynomial (with d replaced by $d/1024$ in the lemma) which will yield the lower bound of $2^{\Omega(d)}$ for any id-interval multilinear circuit computing $\sigma(P)$, for a random σ with probability $1 - 2e^{-\alpha d}$. For any τ-interval multilinear circuit too the same lower bound applies because $\tau\sigma$ is also a random permutation in S_{2d} with uniform distribution.

■

We are now ready to state and prove the main result of this section.

Theorem 21. *There is a set-multilinear polynomial $f \in \mathbb{F}[X]$, where $X = \sqcup_{i=1}^{\log d+2d} X_i$ and $X_i = \{x_{0,i}, x_{1,i}\}, 1 \le i \le \log d+2d$ such that f has an $O(d^2)$ size monotone set-multilinear ABP. Furthermore, assuming the SOS conjecture, for all $\tau \in S_{\log d+2d}$, any τ-interval multilinear circuit computing f has size $2^{\Omega(d)}$.*

Proof. Suppose the SOS conjecture holds. Let $\sigma \in S_{2d}$ be some permutation. By Lemma 20, if we pick permutations $\sigma_1, \sigma_2, \ldots, \sigma_d \in S_{2d}$ independently and uniformly at random, then the probability that there is a $\sigma \in S_{2d}$ such that each $\sigma_i(P)$ can be computed by some σ-interval multilinear circuit is bounded by $(2e^{-\alpha d})^d = e^{-\Omega(d^2)}$. As $|S_{2d}| = (2d)!$, by the union bound it follows that there exist permutations $\sigma_1, \sigma_2, \ldots, \sigma_d \in S_{2d}$ such that for any $\sigma \in S_{2d}$ at least one of the $\sigma_i(P)$ requires $2^{\Omega(d)}$ size σ-interval multilinear circuits.

We fix such a set of permutations $\sigma_1, \sigma_2, \ldots, \sigma_d \in S_{2d}$ and define the polynomial f by "interpolating" the $\sigma_i(P)$. To this end, we need the fresh $\log d$ variable sets. For each $c : 1 \le c \le d$, let its binary encoding also be denoted by c, where $c \in \{0,1\}^{\log d}$. Let u_c denote the monomial

$$u_i = \prod_{j=2d+1}^{2d+\log d} x_{c_j,j}.$$

Hence the monomial u_c can also be seen as an encoding of $c : 1 \le c \le d$. We define

$$f = \sum_{c \in \{0,1\}^{\log d}} u_c \sigma_c(P).$$

Clearly, $f \in \mathbb{F}[X]$ and for each 0-1 assignment $c \in \{0,1\}^d$ to the variables in $X_j, 2d+1 \le j \le 2d+\log d$, the polynomial f becomes $\sigma_c(P)$.

Let $\tau \in S_{\log d+2d}$ be an arbitrary permutation. Let $\hat{\tau} \in S_{2d}$ denote the relative ordering of the indices in $[1, 2, \ldots, d]$ induced by τ.

Suppose f has a $2^{o(d)}$ size τ-interval multilinear circuit. Then, by different 0-1 assignments $c \in \{0,1\}^d$ to variables in $X_j, 2d+1 \le j \le 2d+\log d$ we will obtain a $2^{o(d)}$ size $\hat{\tau}$-interval multilinear circuit for each $\sigma_c(P)$ which is a contradiction to the choice of the σ_i. ■

Finally, we show for monotone circuits that the analogue of Theorem 21 holds unconditionally.

Theorem 22. *There is a set-multilinear polynomial $f \in \mathbb{F}[X]$, where $X = \sqcup_{i=1}^{\log d+2d} X_i$ and $X_i = \{x_{0,i}, x_{1,i}\}, 1 \leq i \leq \log d + 2d$ such that for any $\tau \in S_{\log d+2d}$ any monotone τ-interval multilinear circuit computing f has size $2^{\Omega(d)}$.*

Proof. Let

$$f = \sum_{c \in \{0,1\}^{\log d}} u_c \sigma_c(P),$$

be the polynomial defined in Theorem 21. By Corollary 13, f has small monotone set-multilinear ABPs. Let C be any monotone τ-interval multilinear circuit of size s computing f.

The *bilinear complexity* of $(\sum_{i=1}^k x_i^2)(\sum_{j=1}^k y_j^2)$ [HWY10, Section 1.3] is the minimum t such that $(\sum_{i=1}^k x_i^2)(\sum_{j=1}^k y_j^2)$ equals $\sum_{i=1}^t f_i f_i'$, for bilinear forms f_i and f_i' (both f_i and f_i' are bilinear in the variable sets $\{x_1, x_2, \ldots, x_k\}$ and $\{y_1, y_2, \ldots, y_k\}$). It is shown in [HWY10, Theorem 1.6] that a lower bound of $\Omega(k^{1+\epsilon})$ for constant $\epsilon > 0$ yields a $2^{\Omega(d)}$ size lower bound for noncommutative circuits computing $ID = \sum_{w \in \{x_0,x_1\}^d} ww$ (over $\mathbb{F}$ algebraically closed such that $char\mathbb{F} \neq 2$). The connection to the SOS conjecture comes (for fields of characteristic different 2) because the bilinear complexity is related by a constant factor to the minimum sum-of-squares expression.

When f_i and f_i' are monotone bilinear forms we have the following.

Claim 23. *If $(\sum_{i=1}^k x_i^2)(\sum_{j=1}^k y_j^2)$ equals $\sum_{i=1}^t f_i f_i'$, for monotone bilinear forms f_i and f_i' then $t \geq k^2$.*

Proof of Claim. By assumption, the f_i and f_i' are all monotone bilinear in variable sets $\{x_1, x_2, \ldots, x_k\}$ and $\{y_1, y_2, \ldots, y_k\}$. For each $i \in [t]$, all nonzero terms in the product $f_i f_i'$ must be of the form $x_j^2 y_\ell^2$ because there are no cancellations. Therefore, if f_i has a nonzero term of the form $x_j y_\ell$ then f_i' can have only $x_j y_\ell$ as a nonzero term and no other terms, because the coefficients are nonnegative in f_i and f_i' and there are no cancellations. Thus, each product $f_i f_i'$ involves precisely one pair (x_j, y_ℓ) for some $j, \ell \in [k]$. This forces $t \geq k^2$.

It follows from the proof of [HWY10, Corollary 5.4] and the above claim that any monotone noncommutative circuit for $ID = \sum_{w \in \{x_0,x_1\}^d} ww$ is of size $2^{\Omega(d)}$. We observe that Corollary 15, Lemma 18, Lemma 20, and Theorem 21 all hold for monotone interval multilinear circuits unconditionally, because the SOS conjecture is true in the monotone case by the above claim. This completes the proof. ■

5 Parse tree restrictions on set-multilinear circuits

In this section we investigate lower bounds for set-multilinear circuits computing the Permanent that satisfy some "semantic" restrictions on the parse trees of monomials.

For an arithmetic circuit C, a *parse tree* for a monomial m is a multiplicative subcircuit of C rooted at the output gate defined by the following process starting from the output gate:

- At each $+$ gate retain exactly one of its input gates.
- At each $\times$ gate retain both its input gates.
- Retain all inputs that are reached by this process.
- The resulting subcircuit is multiplicative and computes the monomial m (with some coefficient).

Definition 24. *Let C be a set-multilinear circuit computing $f \in \mathbb{F}[X]$ for variable partition $X = \sqcup_{i=1}^{d} X_i$.*

- *A parse tree T for a monomial is, in fact, a* binary tree *with leaves labeled by variables (ignoring the leaves labeled by constants) and internal nodes labeled by gate names (the $\times$ gates of C occurring in the parse tree). By set-multilinearity, in each parse tree there is exactly one variable from each subset X_i, and each variable occurs at most once in a parse tree.*
- *With each parse tree T we can associate its* parse tree type *$\hat{T}$ which is a binary tree with d leaves. Each node v of T is labeled by an index set $I_v \subseteq [d]$: The root is labeled by $[d]$, each leaf is labeled by a distinct singleton set $[i], 1 \leq i \leq d$, and if v has children v_1 and v_2 in the tree then $I_v = I_{v_1} \sqcup I_{v_2}$.*

Thus, given a set-multilinear circuit C we can consider: (a) the set of parse tree types of the entire circuit C, and (b) the set of parse tree types of a given monomial.

For set-multilinear ABPs computing a degree d polynomial, parse trees of monomials are just simple paths of length d, and the corresponding parse tree types are also simple paths of length d. Furthermore, every ROABP have a unique parse tree type. As exponential lower bounds for ROABPs computing the permanent are known [Nis91] using the partial derivative method, a natural question is whether we can obtain lower bounds when more than one parse tree type is allowed in the set-multilinear circuit. We consider the following two restrictions and prove lower bounds.

1. Set-multilinear circuits with few parse tree types.
2. *Unambiguous* set-multilinear circuits: i.e. circuits in which each monomial has a unique parse tree type (but the number of different parse tree types in the circuit is unbounded).

5.1 Set-multilinear circuits with few parse tree types

Let C be a set-multilinear circuit computing a degree-n polynomial $f \in \mathbb{F}[X]$ for variable partition $X = \sqcup_{i=1}^{n} X_i$ such that *the total number* of parse tree types in the circuit is bounded, say, by a polynomial in n. Can we prove superpolynomial lower bounds for such circuits?

We are able to show non-trivial lower bounds when the number of parse trees are small enough. More precisely, suppose C is a set-multilinear circuit that computes PER_n and C has at most r parse tree types. Then we show that C is of size $2^{\Omega(\frac{n}{r \log n})}$. Here is a brief outline of the proof: We decompose C into a sum of r many set-multilinear formulas $C_i, 1 \leq i \leq r$, such that each C_i has a unique proof tree type. Next, we convert each such C_i into an ROABP A_i. Thus, the sum of these ROABPs $A_i, 1 \leq i \leq r$ computes the permanent PER_n and we can apply Corollary 10 to the sum of these A_i's and obtain the claimed lower bound. We now present the details.

Lemma 25. *Let C be a set-multilinear circuit of size s computing a degree-d polynomial $P \in \mathbb{F}[X]$. If all parse trees in C have the same parse tree type T, then C can be efficiently transformed into a set-multilinear formula C' of size $s^{O(\log d)}$ such that in C' too all parse trees have the same parse tree type T', where T' depends only on T (and not on the circuit C).*

Proof. The proof is a standard depth reduction argument as, for example Hyafil's result [Hya79]. The only extra work we need to do is argue about the parse tree type.

We prove the lemma by induction on the size of the index set of the output gate of C (i.e., degree of P). At the input gates, where index set is a singleton set, it clearly holds. Suppose the index set of the output gate is of size at least 2. Let T_C denote the unique parse tree type for all parse trees in C. Each node v of T_C is labelled by its index set $I_v \subseteq [d]$. As T_C is a binary tree, there is a vertex u such that $\frac{d}{3} \leq |I_u| \leq \frac{2d}{3}$. Let $S_u = \{v \in C \mid I_v = I_u\}$. Let $\hat{C}_v$ denote the set-multilinear circuit obtained from C by (i) setting to zero all the gates in $S_u \setminus \{v\}$, and (ii) replacing the gate v by the constant 1. Let Q_v denote the polynomial computed at the output gate of $\hat{C}_v$. Its index set is $[d] \setminus I_v$. Let P_v denote the polynomial computed at a gate v of C. Then we can clearly write

$$P = \sum_{v \in S_u} P_v Q_v.$$

Let C_v denote the subcircuit of C with output gate v. Note that

$$\frac{d}{3} \leq deg(P_v), deg(Q_v) \leq \frac{2d}{3}.$$

Thus, for each $v \in S_u$ both P_v and Q_v are set-multilinear polynomials computed by set-multilinear circuits (C_v and $\hat{C}_v$, respectively) of size at most s. Furthermore, these circuits also have the property that all parse trees has the same parse tree type (otherwise, C would not have the property).

By induction hypothesis, for each $v \in S_u$ we have set-multilinear formulas F_v and $\hat{F}_v$ such that:

- F_v and $\hat{F}_v$ compute P_v and Q_v, respectively.

- The size of F_v as well as $\hat{F}_v$ is bounded by $s^{O(\log \frac{2d}{3})}$.

- All parse trees in F_v have a unique parse tree type. All parse trees in $\hat{F}_v$ have a unique parse tree type.

Furthermore, the circuit C has the following stronger property: suppose v and v' are two gates with the same index set $I_v = I_{v'}$. Then the unique parse tree type associated with subcircuit C_v is the same as the unique parse tree type for subcircuit $C_{v'}$. Otherwise, the circuit C would not have a unique parse tree type associated with it.

Since each subcircuit $C_v, v \in S_u$ has the same index set and, thus, the same parse tree type associated to it, it follows by induction hypothesis that all the formulas $F_v, v \in S_u$ also have the same unique parse tree type. The same property holds for $\hat{C}_v, v \in S_u$ and hence $\hat{F}_v, v \in S_u$.

Therefore, each of the product polynomials $P_vQ_v, v \in S_u$, computed by the formulas $F_v \times \hat{F}_v, v \in S_u$, with a $\times$ output gate, all have the same parse tree type. Thus, since $|S_u| \leq s$ the polynomial $P = \sum_{v \in S_u} P_vQ_v$ has a set multilinear formula C' of size $\leq s(2s^{O(\log \frac{2d}{3})}) \leq s^{O(\log d)}$ and all the parse trees of C' have the same parse tree type T'. Furthermore, it is clear that T' depends only on T_C. This completes the proof of the lemma. ■

Lemma 26. *Let C be a set-multilinear formula of size s computing a degree d polynomial P, such that C has a unique parse tree type T. Then C can be transformed into a set-multilinear ABP that has a unique parse tree type T' which depends only on T and not on the formula C.*

Proof. We proof the lemma by induction on the size of formula C. Suppose the output gate of C is a $+$ gate. Let C_1 and C_2 be the two subformulas. Since C has a unique parse tree type T, both subcircuits C_1 and C_2 have the same unique tree type T. By induction hypothesis the two subformulas C_1 and C_2 of C with same parse tree T can be converted into set-multilinear ABPs A_1 and A_2, respectively, such that both A_1 and A_2 have the same unique parse tree type T'. The set-multilinear ABP A for their sum is obtained by "parallel composition" of the two ABPs A_1 and A_2. Clearly, A has the same unique parse tree type T'.

Next, suppose output gate of formula C is a $\times$ gate. Since C has unique proof tree type T, the subcircuits C_1 and C_2 of C have unique parse tree types T_1 and T_2, respectively. Let T_1 be the left subtree of T and T_2 be the right subtree. By induction hypothesis both C_1 and C_2 have ABPs A_1 and A_2 with unique parse tree types T_1 and T_2 respectively. In order to compute their product, we can take the "series composition" of A_1 and A_2, which yields the desired set-multilinear ABP with unique parse tree type. ■

Lemma 27. *Let C be a set-multilinear circuit of size s computing polynomial $P \in \mathbb{F}[X]$ of degree d such that C has r distinct parse tree types. Then from C we can construct set-multilinear circuits $C_i, 1 \leq i \leq r$ such that $\sum_{i \in [r]} C_i$ computes polynomial P, each C_i is of size bounded by s, and each C_i has a unique parse tree type.*

Proof. Let the parse tree types of C be $T_1, T_2, \cdots, T_r$. We describe the construction of circuit C_i from C corresponding to parse tree type T_i. In C_i, we label 0 for all the outgoing edges of gates v in C whose index set $I_v \subseteq [n]$ is not equal to any of the index sets of parse tree T_i. Clearly, the parse trees of C_i are precisely all parse trees of parse tree type T_i present in circuit C and with the same coefficients. Therefore, $\sum_{i=1}^{r} C_i$ computes polynomial P. This completes the proof. ■

As a consequence of the above lemmas we obtain the following.

Theorem 28. *Let C be a set-multilinear circuit computing the permanent polynomial PER_n (or determinant DET_n) such that C has at most r distinct parse tree types. Then the size of C is $\Omega(2^{\frac{n}{r \log n}})$.*

Proof. Let C be of size s. The idea is to convert C into a narrow set multilinear ABPs and apply the lower bound for narrow set multilinear ABPs (Corollary 10). First, by Lemma 27 we compute set-multilinear circuits $C_i, 1 \leq i \leq r$, of size s each, such that C_i has unique parse tree type T_i. Next, by Lemma 25, each C_i can be converted into a set-multilinear formula C_i' of size $s^{O(\log n)}$, also of unique parse tree type. Finally, by Lemma 26 C_i' can be transformed into a homogeneous d-layer set-multilinear ABP A_i of size $s^{O(\log n)}$ which has unique parse tree type. Their sum, $\sum_{i=1}^{r} A_i$, obtained by "parallel composition", is a set-multilinear ABP A with at most r many parse tree types. Clearly, it follows that at each layer $i \in [n]$ of the ABP A, the typewidth is bounded by r.

By Lemma 25, the size of ABP A is bounded by $rs^{O(\log n)}$. As A computes PER_n (or DET_n), by Corollary 10 the size of A is lower bounded by $\Omega(2^{\frac{n}{r}})$. Thus, $rs^{O(\log n)} = \Omega(2^{\frac{n}{r}})$, which implies that $s = \Omega(2^{\frac{n}{r \log n}})$. ■

5.2 Unambiguous set-multilinear circuits

Definition 29. *A set-multilinear circuit C computing a degree d polynomial $f \in \mathbb{F}[X]$, with variable partition $X = \sqcup_{i=1}^{d} X_i$, is said to be* unambiguous *if for every monomial $m \in X^d$ has a* unique *parse tree type in circuit C.*

In unambiguous circuits different monomials are allowed to have different parse tree types. Furthermore, each monomial can have many parse trees, only the parse tree types have to be all identical. Unambiguous boolean circuits and unambiguous computation in general are well-studied in complexity theory.

Clearly, ROABPs are a special case of unambiguous set-multilinear ABPs. Also, unambiguous set-multilinear circuits can have many parse tree types, unlike what we considered in Section 5.1.

Theorem 30. *Let C be an unambiguous set-multilinear circuit with variable partition $X = \sqcup_{i=1}^{n} X_i$, where $X_i = \{X_{ij} \mid 1 \leq j \leq n\}$, such that C computes the permanent polynomial PER_n. Then C is of size $2^{\Omega(n)}$.*[2]

Proof. Suppose C is an unambiguous size set-multilinear circuit of size s computing the permanent polynomial PER_n. Let $G_{n/3}$ denote the set of all product gates g in C such that $deg(g) > n/3$ and $deg(g_1) \leq n/3$ and $deg(g_2) \leq n/3$, where g_1 and g_2 are the gates that are input to g. It follows that $n/3 < deg(g) \leq 2n/3$. Furthermore, every parse tree of the circuit C has at least one gate from $G_{n/3}$ and at most two gates from $G_{n/3}$.

[2] The same lower bound result holds for DET_n.

Since C is unambiguous, every monomial of PER_n has a unique parse tree type. Consequently, by the pigeon-hole principle there is an index set $I \subseteq [n]$ of size $n/3 < |I| \leq 2n/3$ with the following property: for at least $n!/s$ many monomials m of PER_n, every parse tree of m has a gate in $G_{n/3}$ of index set I. Let

$$\hat{G} = \{g \in G_{n/3} \mid \text{ index set of } g \text{ is } I\}.$$

We will lower bound $|\hat{G}|$. For $g \in \hat{G}$ let C_g be the subcircuit of C rooted at the gate g. Let $\partial_g C$ denote the partial derivative of the output gate of C w.r.t. gate g as defined in Section 2. A circuit for $\partial_g C$ can be obtained from C as follows:

- For each gate $h \in \hat{G}$ such that $h \neq g$, label by 0 all outgoing edges from h.
- Replace gate g with constant 1.
- For all gates $h \in G_{n/3}$ such that $I_h \cap I \neq \emptyset$ label by 0 all outgoing edges of h.

Now, consider the circuit

$$C' = \sum_{g \in \hat{G}} C_g \partial_g C. \tag{3}$$

Since C_g and $\partial_g C$ are both set-multilinear circuits of size at most s, clearly the size of the set-multilinear circuit C' is bounded by $2s^2$.

Let M denote the set of monomials m of PER_n such that every parse tree of m has a gate in $\hat{G}$. By choice of I

$$|M| \geq \frac{n!}{s}.$$

Claim 31. *Let $f \in \mathbb{F}[X]$ denote the polynomial computed by C'. Then*

- *The coefficient of every monomial M in f is* 1.
- *The coefficient of any monomial not in M is* 0 *in f.*

Proof of Claim. Let $m \in X^n$ be any monomial. Since C is an unambiguous circuit, if some parse tree of m in C has a gate in $\hat{G}$ then every parse tree of m has a gate in C. Hence every parse tree of such a monomial m is accounted for in the circuit C'. Thus the net contribution of any such monomial m is the coefficient of m in PER_n. In particular, monomials in M have coefficient 1 and all other monomial in X^n have coefficient 0.

Similar to Equation 1, for the set-multilinear polynomial $f \in \mathbb{F}[X]$ with variable partition $X = \sqcup_{i=1}^n X_i$, where $X_i = \{X_{ij} \mid 1 \leq j \leq n\}$, we define matrix M_f whose rows are indexed by degree $|I|$ set-multilinear monomials m_1

and columns by degree $n-|I|$ set-multilinear monomials m_2 such that m_1 is a monomial in variables $\sqcup_{i\in I}X_i$ and m_2 in variables $\sqcup_{i\notin I}X_i$ and

$$M_f(m_1,m_2)=f(m_1m_2).$$

For any degree n set-multilinear monomial $m\in X^n$ we can uniquely write it as $m=m_1m_2$, where m_1 is a monomial in variables $\sqcup_{i\in I}X_i$ and m_2 in variables $\sqcup_{i\notin I}X_i$. By the above claim

$$M_f(m_1,m_2)=\begin{cases}1, & \text{if } m_1m_2\in M,\\ 0, & \text{otherwise.}\end{cases}$$

Furthermore, notice that for any other factorization $m=m_1'm_2'$ of m, the entry $M_f(m_1',m_2')=0$, because the circuit C is unambiguous.

Claim 32. *$s\geq rank(M_f)\geq \frac{\binom{n}{n/3}}{s}$.*

Proof of Claim. To see that $s\geq rank(M_f)$ it suffices to note from Equation 3 that

$$M_f=\sum_{g\in\hat{G}}M_{C_g\partial_gC},$$

where each $M_{C_g\partial_gC}$ is a rank 1 matrix. Thus, $s\geq|\hat{G}|\geq rank(M_f)$.

Now we show the other inequality in the claim. For each subset $S\in\binom{[n]}{|I|}$ we group together the rows of matrix M_f indexed by monomials $m_1=X_{i_1j_1}X_{i2j_2}\ldots X_{i_kj_k}$, where $I=\{i_1,i_2,\ldots,i_k\}$ and $S=\{j_1,j_2,\ldots,j_k\}$. Likewise, for each subset $T\in\binom{[n]}{n-|I|}$ we group together the columns indexed by monomials $m_2=X_{i_1j_1}X_{i2j_2}\ldots X_{i_\ell j_\ell}$, where $[n]\setminus I=\{i_1,i_2,\ldots,i_\ell\}$ and $T=\{j_1,j_2,\ldots,j_\ell\}$.

The matrix M_f consists of different (S,T) blocks, corresponding to subsets $S\in\binom{[n]}{|I|}$ and $T\in\binom{[n]}{n-|I|}$. For each such S, only the $(S,[n]\setminus S)$ block has nonzero entries. All other blocks in the row corresponding to S or the columns corresponding to $[n]\setminus S$ are zero. Furthermore, we note that the number of entries in each $(S,[n]\setminus S)$ block is clearly bounded by $(|I|!)(n-|I|)!$. Therefore, as there are $n!/s$ many 1's in matrix M_f, there are at least

$$\frac{n!}{s(n-|I|)!|I|!}=\frac{\binom{n}{|I|}}{s}$$

many *nonzero* $(S,[n]\setminus S)$ blocks, each of which contributes at least 1 to the rank of M_f. Hence,

$$rank(M_f)\geq\frac{\binom{n}{|I|}}{s}.$$

As $\binom{n}{|I|}\geq\binom{n}{n/3}=2^{\Omega(n)}$, it follows that $s=2^{\Omega(n)}$, which completes the lower bound proof. ■

Remark 33. *It suffices to assume that the "top half" of the parse tree types are unambiguous for each monomial. More precisely, a* truncated parse tree type *is obtained from a parse tree type by deleting all nodes v such that $|I_v| \leq d/3$. Let C be a set-multilinear circuit computing* PER_n *such that C has the following property: each monomial $m \in X^n$ has at most one truncated parse tree type. The above proof yields the same lower bounds on the size of C.*

6 Summary and open problems

In this paper we investigated lower bound questions for certain set-multilinear arithmetic circuits and ABPs. By imposing a restriction on the number of set types for set-multilinear ABPs, or by restricting the number of parse trees in set-multilinear circuits, we could prove nontrivial lower bounds for the Permanent. We also showed a separation between set-multilinear circuits and interval multilinear circuits, assuming the SOS conjecture.

Some interesting open questions arise from our work: can we show lower bounds for $f(n)$-narrow set-multilinear ABPs for $f(n) = O(n)$? Another question is proving lower bounds for set-multilinear circuits with polynomially (or even $O(n)$) many parse trees computing PER_n.

We believe that for set-multilinear ABPs/circuits the determinant and permanent are equally hard (like for noncommutative circuits [AS10]). Given a set-multilinear ABP for the determinant can we transform it into a set-multilinear ABP for the permanent by somehow "removing" the signs of all the monomials?

Acknowledgments. We thank Joydeep Mukherjee for suggesting that Lemma 19 might be useful in the proof of Lemma 20. We are very grateful to the referees for their insightful remarks and suggestions which have greatly improved the previous version of this paper.

References

[AS10] Vikraman Arvind and Srikanth Srinivasan, *On the hardness of the noncommutative determinant*, Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, 2010, pp. 677–686.

[DP09] Devdatt P. Dubhashi and Alessandro Panconesi, *Concentration of measure for the analysis of randomized algorithms*, Cambridge University Press, 2009.

[FS13] Michael A. Forbes and Amir Shpilka, *Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs*, 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA, 2013, pp. 243–252.

[GKST15] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf, *Deterministic identity testing for sum of read-once oblivious arithmetic branching programs*, 30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA, 2015, pp. 323–346.

[HWY10] Pavel Hrubes, Avi Wigderson, and Amir Yehudayoff, *Non-commutative circuits and the sum-of-squares problem*, Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, 2010, pp. 667–676.

[Hya79] Laurent Hyafil, *On the parallel evaluation of multivariate polynomials*, SIAM J. Comput. **8** (1979), no. 2, 120–123.

[Nis91] Noam Nisan, *Lower bounds for non-commutative computation (extended abstract)*, STOC, 1991, pp. 410–418.

[RY08] Ran Raz and Amir Yehudayoff, *Balancing syntactically multilinear arithmetic circuits*, Computational Complexity **17** (2008), no. 4, 515–535.

[Sha00] D. B. Shapiro, *Composition of quadratic forms*, W. de Gruyter Verlag, 2000.

[SY10] Amir Shpilka and Amir Yehudayoff, *Arithmetic circuits: A survey of recent results and open questions*, Foundations and Trends in Theoretical Computer Science **5** (2010), no. 3-4, 207–388.

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff, *Fast parallel computation of polynomials using few processors*, SIAM J. Comput. **12** (1983), no. 4, 641–644.

Appendix

Proof of Theorem 2

Proof. By definition, Φ is a homogeneous arithmetic circuit. We assume that Φ is non-redundant (i.e , for all gates v in Φ the polynomial f_v computed at v is nonzero). Since Φ is set-multilinear, at each gate v in Φ there is an associated index set $I_v \subseteq [d]$ such that the polynomial f_v is set-multilinear of degree $|I_v|$ over the variable set X_{I_v}, where

$$X_{I_v} = \sqcup_{i \in I_v} X_i.$$

We denote the subcircuit rooted at the gate v by Φ_v.

Partial Derivative of f_v by a gate w

Let v, w be any two gates in circuit Φ. Following the exposition in [SY10], let $\Phi_{w=y}$ denote the circuit obtained by removing any incoming edges at w and labeling w with a new input variable y and $f_{v,w}$ denote the polynomial (in $X \cup \{y\}$) computed at gate v in circuit $\Phi_{w=y}$. Define

$$\partial_w f_v = \partial_y f_{v,w}.$$

Note that $f_{v,w}$ is linear in y. Clearly, if w does not occur in Φ_v then $\partial_w f_v = 0$. If w occurs in Φ_v, since Φ is set-multilinear the polynomial $f_{v,w}$ is linear in y and is of the form

$$f_{v,w} = h_{v,w} y + g_{v,w}.$$

Therefore, $\partial_w f_v = h_{v,w}$. We make the following immediate observations from the set-multilinearity of Φ.

- Either $\partial_w f_v = 0$ or $\partial_w f_v$ is a homogeneous set-multilinear polynomial of degree $deg(v) - deg(w)$ over variable set $X \setminus X_{I_w}$.
- If $deg(v) < 2.deg(w)$ and v is a product gate with children v_1, v_2 such that $deg(v_1) \geq deg(v_2)$, then $\partial_w f_v = f_{v_2}.\partial_w f_{v_1}$.

For a positive integer m, let G_m denote the set of product gates t with inputs t_1, t_2 in Φ such that $m < deg(t)$ and $deg(t_1), deg(t_2) \leq m$. We observe the following claims (analogous to [SY10]) which are easily proved.

Claim 34. *Let Φ be a set-multilinear nonredundant arithmetic circuit over variable set $X = \sqcup_{i=1}^{d} X_i$. Let v be a gate in Φ such that $m < deg(v) \leq 2m$ for a positive integer m. Then $f_v = \sum_{t \in G_m} f_t.\partial_t f_v$.*

Claim 35. *Let Φ be a set-multilinear non-redundant arithmetic circuit over the field $\mathbb{F}$ and over the set of variables X. Let v and w be gates in Φ such that $0 < deg(w) \leq m < deg(v) < 2deg(w)$. Then $\partial_w f_v = \sum_{t \in G_m} \partial_w f_t.\partial_t f_v$*

Construction of Ψ:

We now explain the construction of the depth-reduced circuit Ψ. The construction is done in stages. Suppose upto Stage i we have computed, for $1 \leq j \leq i$ the following:

- All polynomials f_v for gates v such that $2^{j-1} < deg(v) \leq 2^j$.
- All partial derivatives of the form $\partial_w f_v$ for gates v and w such that $2^{j-1} < deg(v) - deg(w) \leq 2^j$ and $deg(v) < 2deg(w)$.
- Furthermore, inductively assume that the circuit computed so far is set-multilinear of $O(i)$ depth, such that all product gates are fanin 2, sum gates are of unbounded fanin.

We now describe Stage $i+1$ where we will compute all f_v for gates v such that $2^i < deg(v) \leq 2^{i+1}$ and also all partial derivatives of the form $\partial_w f_v$ for gates v and w such that $2^i < deg(v) - deg(w) \leq 2^{i+1}$ and $deg(v) < 2deg(w)$. Furthermore, we will do this by adding a depth of $O(1)$ to the circuit and poly(d, s) many new gates maintaining set-multilinearity.

Stage i+1: We describe the construction at this stage in two parts:

Computing f_v

Let v be a gate in Φ such that $2^i < deg(v) \leq 2^{i+1}$ and let $m = 2^i$. By Claim 34, we have

$$f_v = \sum_{t \in T} f_t \partial_t f_v = \sum_{t \in T} f_{t_1} f_{t_2} \partial_t f_v,$$

where T is the set of gates $t \in G_m$, with children t_1 and t_2 such that t is in Φ_v. Note that if t is not in Φ_v, then $\partial_t f_v = 0$. Let $t \in T$ be a gate with inputs t_1 and t_2. Thus. $m < deg(t) \leq 2m$, $deg(t_1) \leq m$, $deg(t_2) \leq m$. Hence $deg(v) - deg(t) \leq 2^{i+1} - 2^i = 2^i$ and $deg(v) \leq 2^{i+1} < 2.deg(t)$. Therefore, f_{t_1}, f_{t_2} and $\partial_t f_v$ are already computed. Thus, in order to compute f_v we need $O(s)$ many $\times$ gates and $O(1)$ many $+$ gates. Overall, with $O(s^2)$ many new gates and $O(1)$ increase in depth we can compute all f_v such that $2^i < deg(v) \leq 2^{i+1}$. Furthermore, we note that f_{t_1}, f_{t_2} and $\partial_t f_v$ are all set-multilinear polynomials with disjoint index sets, and the union of their index sets is I_v for each $t \in T$. Thus, the new gates introduced all preserve set-multilinearity.

Computing $\partial_w f_v$

Let v and w be gates in Φ such that $2^i < deg(v) - deg(w) \leq 2^{i+1}$ and $deg(v) < 2deg(w)$. Let $m = 2^i + deg(w)$. Thus, $deg(w) \leq m < deg(v) < 2deg(w)$. Let T' denote the set of gates in Φ_v that are contained in G_m. Note that $\partial_t f_v = 0$ if $t \notin T'$. Hence by Claim 35 we can write

$$\partial_w f_v = \sum_{t \in T'} \partial_w f_t \partial_t f_v,$$

For a gate $t \in T'$, we have $deg(t) \leq deg(v) < 2deg(w)$. Suppose t_1 and t_2 are the gates input to t in the circuit Φ, and $\mathrm{deg}(t_1) \geq deg(t_2)$. Then we can write

$$\partial_w f_v = \sum_{t \in T'} f_{t_2} \partial_w f_{t_1} \partial_t f_v.$$

We claim that f_{t_2}, $\partial_w f_{t_1}$, and $\partial_t f_v$ are already computed.

- Since $deg(v) \leq 2^{i+1} + deg(w) \leq 2^{i+1} + deg(t_1) = 2^{i+1} + deg(t) - deg(t_2)$, we have $deg(t_2) \leq 2^{i+1} + deg(t) - deg(v) \leq 2^{i+1}$. Hence f_{t_2} is already computed (in first part of stage $i+1$).

- Since $deg(t_1) - deg(w) \leq 2^i$, the polynomial $\partial_w f_{t_1}$ is already computed in an earlier stage.

- Since $deg(t) > m$, we have $deg(v) - deg(t) \leq deg(v) - m \leq 2^{i+1} - 2^i = 2^i$.

- Thus, since $deg(v) \leq 2^{i+1} + deg(w) \leq 2(2^i + deg(w)) < 2deg(t)$, the polynomial $\partial_t f_v$ is already computed in an earlier stage.

As before, for each such pair of gates w and v, we can compute $\partial_w f_v$ with $O(s)$ new gates (using the polynomials already computed in previous stages), and this increases the circuit depth by $O(1)$. Since we consider pairs of gates (w, v) such that $2^i < deg(v) - deg(w) \leq 2^{i+1}$ at the i^{th} stage, the total number of pairs (w, v) considered over all the stages is bounded by s^2. Hence the number of new gates added over all the stages is $O(s^3)$. Thus the size of the depth-reduced circuit obtained is $O(s^3)$ and its depth is $O(\log d)$. Furthermore, the new gates included clearly also have the set-multilinearity property. This completes the proof of the theorem. ■

ECCC ISSN 1433-8092

http://eccc.hpi-web.de