Towards Optimal Depth Reductions for Syntactically Multilinear Circuits

Mrinal Kumar
University of Toronto, Canada
https://mrinalkr.bitbucket.io/
mrinalkumar08@gmail.com

Rafael Oliveira
University of Toronto, Canada
http://www.cs.utoronto.ca/~rafael/
rafael@cs.toronto.edu

Ramprasad Saptharishi
Tata Institute of Fundamental Research
https://www.tcs.tifr.res.in/~ramprasad/
ramprasad@tifr.res.in

Abstract

We show that any n-variate polynomial computable by a syntactically multilinear circuit of size $\mathrm{poly}(n)$ can be computed by a depth-4 syntactically multilinear ($\Sigma\Pi\Sigma\Pi$) circuit of size at most $\exp\left(O\left(\sqrt{n \log n}\right)\right)$. For degree $d = \omega(n/\log n)$, this improves upon the upper bound of $\exp\left(O(\sqrt{d}\log n)\right)$ obtained by Tavenas [14] for general circuits, and is known to be asymptotically optimal in the exponent when $d < n^{\varepsilon}$ for a small enough constant ε. Our upper bound matches the lower bound of $\exp\left(\Omega\left(\sqrt{n \log n}\right)\right)$ proved by Raz and Yehudayoff [12], and thus cannot be improved further in the exponent. Our results hold over all fields and also generalize to circuits of small individual degree.

More generally, we show that an n-variate polynomial computable by a syntactically multilinear circuit of size $\mathrm{poly}(n)$ can be computed by a syntactically multilinear circuit of product-depth Δ of size at most $\exp\left(O\left(\Delta \cdot (n/\log n)^{1/\Delta} \cdot \log n\right)\right)$. It follows from the lower bounds of Raz and Yehudayoff [12] that in general, for constant Δ, the exponent in this upper bound is tight and cannot be improved to $o\left((n/\log n)^{1/\Delta} \cdot \log n\right)$.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases arithmetic circuits, multilinear circuits, depth reduction, lower bounds

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.78

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at https://arxiv.org/abs/1902.07063.

Funding *Mrinal Kumar*: A part of this work was done during the postdoctoral stay at Harvard, during the lower bounds semester at Simons Institute for the Theory of Computing, Berkeley and while visiting TIFR, Mumbai.
Rafael Oliveira: Part of this work was done while visiting the Simons Institute for the Theory of Computing.
Ramprasad Saptharishi: Research supported by Ramanujan Fellowship of DST.

Acknowledgements We are extremely thankful to Ben Rossman, who pointed us towards this question, and for many stimulating discussions at various stages of this work. We also thank Shubhangi Saraf, Amir Shpilka and Ben Lee Volk for many helpful conversations. Mrinal also thanks Prahladh Harsha for accommodating him in his apartment for a part of the visit to TIFR, where a part of this paper was written.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi; Article No. 78; pp. 78:1–78:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

An algebraic circuit over a field $\mathbb{F}$ and variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is a directed acyclic graph whose internal vertices (called gates) are labeled as either $+$ (sum) or $\times$ (product), and leaves (vertices of indegree zero) are labeled by the variables in $\mathbf{x}$ or constants from $\mathbb{F}$. The gates of outdegree zero in a circuit are called its output gates. Algebraic circuits give a natural and succinct representation for multivariate polynomials; analogous to the way Boolean circuits give a succinct representation of Boolean functions. We refer the reader to the excellent survey of Shpilka and Yehudayoff [13] for an introduction to the area of algebraic circuit complexity. One of the main protagonists in the results in this paper will be the class of syntactically multilinear circuits which we now define.

▶ **Definition 1** (Syntactically Multilinear Circuits). *An algebraic circuit C is said to be syntactically multilinear if at every product gate v in C with inputs $u_1, u_2, \ldots, u_t$, the set of variables in the sub-circuits rooted at u_i are pairwise disjoint from each other.*

The size of an algebraic circuit is the number of edges in it, and its depth is the length of the longest path from an output gate to a leaf. Intuitively, the size of a circuit is an indicator of the time complexity of computing the polynomial, and its depth indicates how fast the polynomial can be computed in parallel.

We now introduce a sequence of fundamental structural results for algebraic circuits, that are collectively called depth reductions; this is the main focus of this paper.

Depth Reductions

In a beautiful, surprising and influential work, Valiant et al. [15] showed that every polynomial family which is efficiently computable by an algebraic circuit is also efficiently computable in parallel. Formally, they showed the following theorem.

▶ **Theorem 2** ([15]). *There is an absolute constant $c \in \mathbb{N}$ such that the following is true. If P be an n-variate homogeneous polynomial of degree d over any field $\mathbb{F}$ which can be computed by an algebraic circuit C of size s, then P can be computed by an algebraic circuit C' (of unbounded fan-in) of depth $c \log d$ and size $(snd)^c$.*

In particular, the theorem says that every polynomial family of polynomially bounded (in n) degree that is computable by a circuit of size $\mathrm{poly}(n)$ and arbitrary depth, is also efficiently computable by a circuit of size $\mathrm{poly}(\log n)$ and depth $O(\log n)$.

In a remarkable extension of Theorem 2, Agrawal and Vinay [1] showed that one can parallelize algebraic circuits even more (reducing the depth to a constant), at the cost of a larger (a non-trivial subexponential factor) blow up in the circuit size. The version of their theorem stated below is due to Tavenas [14], who optimized the parameters further.

▶ **Theorem 3** ([1, 8, 14]). *There is an absolute constant $c \in \mathbb{N}$ such that the following is true. If P is an n-variate homogeneous polynomial of degree d over any field $\mathbb{F}$ which can be computed by an algebraic circuit C of size s, then P can be computed by a homogeneous $\Sigma\Pi\Sigma\Pi$ algebraic circuit C' of size $(snd)^{c\sqrt{d}}$.*

Here, a $\Sigma\Pi\Sigma\Pi$ circuit is an algebraic circuit with four layers of alternating sum and product gates with the top layer being a sum layer. Throughout this paper, when we say a depth-4 circuit, we mean a $\Sigma\Pi\Sigma\Pi$ circuit.

We note that while Theorem 3 as stated above reduces a homogeneous circuit of arbitrary depth to a homogeneous circuit of depth-4, but it easily follows from the proof that the depth reduction preserves syntactic restrictions. That is, if we start with a syntactically multilinear and homogeneous circuit, the resulting depth-4 circuit is also syntactically multilinear and homogeneous. This statement will be of particular interest as we study depth reductions for syntactically multilinear circuits in this paper.

On the optimality of reductions to depth-4

An immediate consequence of Theorem 2 and Theorem 3 is that strong enough lower bounds for algebraic circuits of bounded depth imply superpolynomial lower bounds for general algebraic circuits. Thus, the questions of proving lower bounds for bounded depth circuits, and that of understanding if the parameters in Theorem 3 can be improved further seem to be of fundamental interest. In the last few years, we have had significant progress on both these fronts. Following a long line of work starting with a work of Kayal [7] and Gupta et al. [5], we now know extremely good lower bounds for homogeneous depth-4 circuits.

▶ **Theorem 4** (Kumar and Saraf [9]). *There exists a polynomial family $\{f_n\}$, where f_n is a homogeneous n-variate polynomial of degree $d = n^{\varepsilon}$, for an absolute constant $\varepsilon > 0$, such that f_n is computable by an algebraic circuit of size $\mathrm{poly}(n)$, but any homogeneous depth-4 circuit computing f_n has size $n^{\Omega(\sqrt{d})}$.*

Moreover, the family $\{f_n\}$ is computable by a syntactically multilinear circuit of polynomial size.

If we allow the hard polynomial to be explicit but not necessarily have small circuits, then upper bound on the degree d in the above theorem can be increased to as large as $n^{1-\varepsilon}$ for any constant $\varepsilon > 0$.[1] Thus, in general, the exponent in the upper bound on the size of the depth-4 circuit obtained in Theorem 3 cannot be improved asymptotically. In fact, the theorem shows that we cannot even expect such an improvement for syntactically multilinear circuits in the setting when the degree d is sufficiently smaller than the number of variables n. A natural question here is to understand if Theorem 3 is also asymptotically tight in the exponent when the degree is larger. The following result of Raz and Yehudayoff goes a long way towards answering this question.

▶ **Theorem 5** ([12]). *There is a family of multilinear polynomials $\{f_n\}$ such that, for every n, the polynomial f_n is an n-variate degree $d = \Theta(n)$ polynomial that can be computed by a syntactically multilinear circuit of size $\mathrm{poly}(n)$, but any multilinear circuit of depth-4 computing f_n has size $n^{\Omega\left(\sqrt{n/\log n}\right)}$.*

More generally, for any constant Δ, any syntactically multilinear circuit of product-depth[2] Δ computing f_n must have size $n^{\Omega\left((n/\log n)^{1/\Delta}\right)}$.

For depth-4 circuits (or $\Delta = 2$), asimilar result was proved by Hegde and Saha [6] for the more general[3] class of circuits called multi-k-ic circuits, where the *formal degree* of any variable in the circuit is bounded by a parameter k (formally defined in Definition 17).

▶ **Theorem 6** ([6]). *There is an explicit family $\{f_n\}$ of n-variate multilinear polynomials of degree $d = \Theta(n)$ such that, for every $k \leq (n \log n)^{0.9}$, any multi-k-ic circuit of depth-4 computing f_n has size at least $n^{\Omega\left(\sqrt{n/(k \log n)}\right)}$.*

[1] Though this is not explicitly mentioned in these results, the proofs can be extended to this regime of parameters.

[2] Also referred to as a syntactically multilinear $(\Sigma\Pi)^{\Delta}$ circuit.

[3] A multilinear circuit is a multi-k-ic circuit for $k = 1$.

Thus, Theorem 5 and Theorem 6 shows that the exponent $\sqrt{d}$ in the exponent in Theorem 3 cannot be replaced by $o\left(\sqrt{n/\log n}\right)$. Thus, in the regime when $d = \Theta(n)$, there is a gap of $\sqrt{\log n}$ between the known lower bounds and what is potentially achievable via depth reduction. Raz and Yehudayoff [12] also observe that using their techniques, the lower bound cannot be improved to $n^{\omega(\sqrt{n/\log n})}$. Our main motivation for this work was to bridge this gap. In the light of Theorem 4, we believed the upper bound of $n^{O(\sqrt{d})}$ in Theorem 3 to be right bound for multilinear circuits for all d, and had hoped to improve the lower bound in Theorem 5 to $n^{\Omega(\sqrt{n})}$.

However, as we discuss next, the correct exponent for depth reduction to depth-4 in the high degree regime turns out to be $\sqrt{n/\log n}$. In addition to being surprising, this also offers a potentially viable approach to the question of proving superpolynomial lower bounds for syntactically multilinear circuits by extending Theorem 4 to the high degree regime. We now state our results and discuss the connections to multilinear circuit lower bounds.

1.1 Results

We start by stating our main theorems.

▶ **Theorem 7.** *Let C be a multi-k-ic circuit of size s computing a polynomial in n variables. Then, there is a multi-k-ic $\Sigma\Pi\Sigma\Pi$ circuit C' of size $s^{O\left(\sqrt{\frac{kn}{\log s}}\right)}$ computing the same polynomial.*

The ideas in the proof of Theorem 7 generalize to give the following statement about reduction to depth Δ circuits for any constant Δ.

▶ **Theorem 8.** *Let C be a multi-k-ic circuit of size s computing a polynomial in n variables. Then, there is a multi-k-ic $(\Sigma\Pi)^{\Delta}$ circuit C' computing the same polynomial whose size is at most*

$$s^{O\left(\Delta\cdot(nk/\log s)^{1/\Delta}\right)}.$$

Thus, for $s = \mathrm{poly}(n)$, $k = o(\log s)$ and $n \geq d \geq \omega\left(\frac{kn}{\log s}\right)$, the exponents in the upper bounds in Theorem 7 are asymptotically better than that in Theorem 3. An immediate consequence of Theorem 7 is the following corollary.

▶ **Corollary 9.** *Let $\{f_n\}$ be an explicit family of multilinear polynomials, such that f_n is an n variate polynomial of degree $d = \omega(n/\log n)$, and any multilinear $\Sigma\Pi\Sigma\Pi$ circuit computing f_n has size at least $n^{\Omega(\sqrt{d})}$. Then, $\{f_n\}$ requires superpolynomial size syntactically multilinear circuits.*

The corollary is of interest since by Theorem 4, we know $n^{\Omega(\sqrt{d})}$ lower bounds for homogeneous multilinear $\Sigma\Pi\Sigma\Pi$ circuits, when $d = n^{\varepsilon}$. Thus extending these bounds so that they hold for higher degree polynomials will imply superpolynomial lower bounds for multilinear circuits. The current best lower bound known for multilinear circuits is a nearly quadratic lower bound in a recent work of Alon et al. [3]. The standard technique for proving lower bounds for multilinear models is via the rank of the *partial derivative matrix* under a random partition of variables (due to Raz [10]). This has been useful in almost all of the known lower bounds for multilinear models, such as super polynomial lower bounds for multilinear formulas [10], exponential lower bounds for constant depth multilinear circuits [12] as well as the currently known superlinear and nearly quadratic lower bounds for multilinear circuits [11, 3]. However, this technique is too weak to yield even super-cubic lower bounds for syntactically multilinear circuits. Thus, currently we do not even have potential approaches to proving superpolynomial lower bounds for multilinear circuits. In the light of this, it certainly seems worth exploring if the partial derivative based methods used in the proof of Theorem 4 can be extended to work for multilinear polynomials whose degree $d = \omega(n/\log n)$ is high. As far as we understand, there does not seem to be strong evidence one way or the other about this.

For multi-k-ic circuits, we do not even know superpolynomial lower bounds for formulas or even constant depth formulas. Based on the discussion above, Theorem 7 does seem to offer a potentially viable approach to prove these lower bounds.

Finally, we note again that the upper bound on the size of the depth-4 circuit obtained in Theorem 7 cannot be further improved asymptotically in the exponent as Theorem 5 shows.

1.2 Proof Overview

We focus on giving an outline of the proof of Theorem 7 for the multilinear case (or $k = 1$). The proof follows the strategy of the proof of Theorem 3 with some key differences, which we point out as we go along. There are two main steps and we now give a sketch of both of them.

Balancing a syntactically multilinear circuit

For this step, the key notion is that of a *balanced* circuit. We say that a circuit C is balanced with respect to a potential function $\Phi : C \to \mathbb{N}$ (e.g. degree, number of variables), if the fan-in of every product g in C is a constant, and $\Phi(g) \geq 2\Phi(h)$ for every child h of g. In the proof of Theorem 3, the authors essentially use the results of Valiant et al. [15] to balance a homogeneous circuit with the potential function Φ being the formal degree of a gate. For our proof, we show that a syntactically multilinear circuit can in fact be balanced with the potential function being the number of variables in the sub-circuit rooted at a gate. Our proof of this part involves the machinery of *gate quotients* and *frontier decompositions* developed by Valiant et al. in their original proof, although there are some crucial differences which require some non-trivial (albeit simple) insights.

One such challenge stems from the fact that in a homogeneous circuit, the formal degree of any two children of a product gate is the same and equal to the formal degree of the parent, whereas the children might depend on very different (even completely disjoint) sets of variables. To get around this, our notion of *frontier* is different from that of Valiant et al [15]. In [15], frontier is defined with respect to vertices, whereas we define frontier with respect to edges. As a consequence, our frontier decomposition statements are slightly different from those in [15], although they continue to have a natural semantic meaning. This is detailed in Section 5.

Reduction to depth-4 from a balanced circuit

In the second part of our proof, we show that any balanced syntactically multilinear circuit of size s computing a polynomial in n variables can be depth reduced to a syntactically multilinear depth-4 circuit of size $s^{O(\sqrt{n/\log n})}$. The proof is along the lines of the proof of the analogous statement in the homogeneous (non-multilinear) setting by Chillara et al. [4]. The high level idea of the proof is the following : in a balanced circuit C, the polynomial computed at any gate g can be written as a sum of product of *terms*, where the product fan-in is a constant, the sum fan-in is upper bounded by the size of the circuit, and the number of variables in any of the terms is at most half of the number of variables in g. Moreover, each of the terms is a polynomial computed by a gate in C, so this decomposition can be recursively applied. We apply this decomposition repeatedly till every term in the sum of products expression of the output depends on at most t variables. We argue that the sum fan-in of this sum of products expression is at most $s^{O(n/t)}$. Now, we expand each of the terms (which is a multilinear polynomial) as a sum of multilinear monomials in t variables. Thus, the total size of the $\Sigma\Pi\Sigma\Pi$ circuit obtained is $2^t \cdot s^{O(n/t)}$ which is $s^{O(\sqrt{n/\log s})}$ for $t = \sqrt{n \log s}$.

In the proof of the analogous statement for homogeneous non-multilinear circuits, at the end of the repeated applications of the decomposition, each of the terms is of degree at most t. Thus, a sum of product expansion of each such term has size $\binom{n}{t}$, and so the total size of the $\Sigma\Pi\Sigma\Pi$ circuit obtained is $n^t \cdot s^{O(n/t)}$, which for $s = \text{poly}(n)$ is minimized for $t = \sqrt{n}$ and equals $s^{O(\sqrt{n})}$. This explains the gain in the size obtained by Theorem 7.

2 Preliminaries

In this section, we describe the notion of parse-trees and gate quotients which are crucial to our proof and set up some of the machinery we need for the proof.

2.1 Parse-trees and quotients

▶ **Definition 10** (Parse-trees). *Let C be an algebraic circuit. For any $u_0 \in C$, a* parse-tree T rooted at u_0 *is a subcircuit of C that satifies the following properties:*

- *the node $u_0 \in T$,*
- *if $u \in T$ is a multiplication gate of C with $u = v_1 \times v_2$, then v_1, v_2 are also in T,*
- *if $u \in T$ is an addition gate of C with $u = v_1 + v_2$, then* exactly one *of v_1 or v_2 is in T.*

Any such sub-circuit computes just a monomial, and this shall be called the value the parse-tree. Although the parse-tree defined above need not be a tree, it shall unfolded to a tree.

If T is a parse-tree rooted at u, and v is a node that appears on its right-most path, then the tree T' obtained by replacing v only on the right-most path by a leaf labelled 1 is said to be a v-snipped parse-tree rooted at u.

▶ **Definition 11** (Var operator). *For any nodes $u \in C$, we denote by $\text{Var}(u)$ the vector $(d_1, \ldots, d_n) \in \mathbb{N}^n_{\geq 0}$ where d_i is the maximum x_i-degree over all parse-trees rooted at u.*

Similarly, for any pair of nodes $u, v \in C$, we denote by $\text{Var}(u : v)$ the vector $(d_1, \ldots, d_n)$ where d_i is the maximum x_i-degree over all v-snipped parse-tree rooted at u.

We shall also define $|(d_1, \ldots, d_n)| = \sum d_i$.

For a syntactically multilinear circuit C, note that $|\text{Var}(g)|$ for any gate $g \in C$ is precisely the number of distinct variables in the sub-circuit rooted at g.

▶ Remark 12. Throughout this discussion, we will assume that the circuit is *right heavy*. This means that for every multiplication gate, $w = w_L \times w_R$, $\text{Var}(w_R) \geq \text{Var}(w_L)$. Note that this is without loss of generality, since *left* and *right* are merely labels that we can assign arbitrarily to the children of every gate in the circuit.

▶ **Definition 13** (Gate Quotient). *For every two gates u, v in C, the* gate quotient *of u with respect to v, denoted by $[u : v]$ is defined inductively as follows.*

- *If $u = v$, then $[u : v] = 1$.*
- *If $u = u_1 + u_2$, then $[u : v] = [u_1 : v] + [u_2 : v]$.*
- *If $u = u_L \times u_R$, then $[u : v] = [u_L][u_R : v]$.*
- *If v does not appear in the subcircuit rooted at u, then $[u : v] = 0$.*

▶ **Lemma 14.** *Let $u, v \in C$. Then, the polynomial $[u]$ is the sum of values of all parse-trees rooted at u. Furthermore, the polynomial $[u : v]$ is the sum of the values of all the v-snipped parse-trees T that are rooted at u.*

The above lemma is almost folklore and a proof of it can be seen in the work of Allender et al. [2].

2.2 Syntactic restrictions on parse-trees

We remark that throughout this paper, by degree, we mean the syntactic or formal degree, which could be much larger than the actual or semantic degree. The following observation records some basic properties of the Var operator.

▶ **Observation 15.** *Let C be any algebraic circuit. Then,*

- $\mathrm{Var}(u)$ *is monotonically non-increasing as u moves towards the leaves. That is, if u is an ancestor of v, then ever coordinate of* $\mathrm{Var}(u)$ *is at least as large as the corresponding coordinate in* $\mathrm{Var}(v)$.

  Similarly, for any fixed v, the vector $\mathrm{Var}(u : v)$ *is monotonically non-increasing as u moves towards the leaves.*
- *For any multiplication gate $u = u_1 \times u_2$, we have* $\mathrm{Var}(u) = \mathrm{Var}(u_1) + \mathrm{Var}(u_2)$. *Similarly for any v, we have* $\mathrm{Var}(u : v) = \mathrm{Var}(u_1) + \mathrm{Var}(u_2 : v)$.
- *For any addition gate $u = u_1 + u_2$, we have* $\mathrm{Var}(u) = \max(\mathrm{Var}(u_1), \mathrm{Var}(u_2))$, *the coordinate-wise max of the two vectors. Similarly for any v,* $\mathrm{Var}(u : v) = \max(\mathrm{Var}(u_1 : v), \mathrm{Var}(u_2 : v))$.

Proof. The proofs immediately follow from the definitions. ◀

For two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{N}^n_{\geq 0}$, we shall say $\mathbf{v}_1 \preceq \mathbf{v}_2$ if each coordinate of $\mathbf{v}_1$ is at most the corresponding coordinate in $\mathbf{v}_2$.

▶ **Observation 16.** *Suppose $u \in C$ and w is a node in C such that there is some parse-tree rooted at u with w appearing on its rightmost path. Then,*

$$\mathrm{Var}(u : w) + \mathrm{Var}(w) \preceq \mathrm{Var}(u).$$

Similarly, suppose w is a node in C such that there is some v-parse-tree rooted at u with w appearing on its rightmost path. Then,

$$\mathrm{Var}(u : w) + \mathrm{Var}(w : v) \preceq \mathrm{Var}(u : v).$$

Proof. The proof is straightforward; we just give the proof of the second equation. Fix a coordinate i. If $d_i = (\mathrm{Var}(u : w))_i$ then there is some w-snipped parse-tree T_i rooted at u whose x_i-degree equals d_i. Similarly if $e_i = (\mathrm{Var}(w : v))_i$, then there is some v-snipped parse-tree T_i' rooted at w whose x_i-degree is e_i. Clearly the *gluing* of T_i and T_i' obtained by replacing the snipped vertex w in T_i with the tree T_i' is a v-snipped parse-tree rooted at u with x_i-degree $d_i + e_i$. Therefore $d_i + e_i \leq (\mathrm{Var}(u : v))_i$ and the claim follows. ◀

▶ **Definition 17** (Syntactically multilinear and multi-k-ic circuits)**.** *A circuit C is said to be* syntactically multilinear *if* $\mathrm{Var}(u) \in \{0, 1\}^n$ *for all $u \in C$.*

A circuit C is said to be syntactically multi-k-ic *if* $\mathrm{Var}(u) \in \{0, 1, \ldots, k\}^n$ *for all $u \in C$.*

3 Frontier edges and quotient

▶ **Definition 18** (Frontier edges). *For a circuit C, an edge between two gates g_1, g_2 (where g_1 is the parent) is said to be an m-*frontier edge *(for a parameter m) if*

$$|\mathrm{Var}(g_1)| \geq m \text{ and } |\mathrm{Var}(g_2)| < m.$$

We use $\mathcal{F}_m^{\times}$ to denote the set of all m-frontier edges (g_1, g_2) where g_1 is a multiplication gate and g_2 is its right child*. We use $\mathcal{F}_m^{+}$ to denote those where g_1 is an addition gate.*

*Furthermore, if $v \in C$ is a fixed gate, we shall say that (g_1, g_2) is an m-*frontier edge with respect to v *if*

$$|\mathrm{Var}(g_1 : v)| \geq m \text{ and } |\mathrm{Var}(g_2 : v)| < m.$$

We will use $\mathcal{F}_{m,v}^{\times}$ to denote the set of all edges (g_1, g_2) that are m-frontier edges with respect to v where g_1 is a multiplication gate (and g_2 is its right child), and $\mathcal{F}_{m,v}^{+}$ to denote those where g_1 is an addition gate (and g_2 is any child).

4 Decomposition via gate quotients

In this section, we prove the following lemma, which is the key technical observation needed for our proofs.

▶ **Lemma 19.** *Let u, v be gates in an algebraic circuit C with $|\mathrm{Var}(u)| \geq m$ and $|\mathrm{Var}(v)| < m$. Then,*

$$[u] = \sum_{(w,z)\in\mathcal{F}_m^{\times}} [u : w] \cdot [w_L] \cdot [z] \;+\; \sum_{(w,z)\in\mathcal{F}_m^{+}} [u : w] \cdot [z] \tag{1}$$

$$[u : v] = \sum_{(w,z)\in\mathcal{F}_{m,v}^{\times}} [u : w] \cdot [w_L] \cdot [z : v] \;+\; \sum_{(w,z)\in\mathcal{F}_{m,v}^{+}} [u : w] \cdot [z : v] \tag{2}$$

We shall give an informal sketch using the concept of parse-trees. A complete formal proof of the lemma can be found in Appendix A. For any u, v, we have that $[u : v]$ is the sum of all v-snipped parse-trees rooted at u. For any parse-tree, since $|\mathrm{Var}(u)| \geq m$ and $|\mathrm{Var}(v)| < m$ and $\mathrm{Var}(\cdot)$ is a monotonically non-increasing function as we move towards the leaves, there must be a unique edge $(w, z) \in \mathcal{F}_{m,v}^{\times} \cup \mathcal{F}_{m,v}^{+}$ on its right-most path such that $|\mathrm{Var}(w)| \geq m$ and $|\mathrm{Var}(z)| < m$.

If $(w, z) \in \mathcal{F}_{m,v}^{\times}$, then $w = w_L \times z$ is a multiplication gate. Therefore, the sum of the values of all v-snipped parse-trees with w (and hence the edge (w, z)) on its rightmost path is exactly $[u : w][w : v] = [u : w][w_L][z : v]$.

If $(w, z) \in \mathcal{F}_{m,v}^{+}$, then $w = w_1 + z$ is an addition gate. Then, $[u : w] \cdot [w : v]$ is the sum of all v-snipped parse-trees with w on its rightmost path and $[u : w][w : v] = [u : w][w_1 : v] + [u : w][z : v]$. Each v-snipped parse-tree with w on its rightmost path either has (w, w_1) on the rightmost path or (w, z). The term $[u : w][w_1 : v]$ is precisely the sum of the values of such[4] parse-trees with (w, w_1) on its rightmost path, and $[u : w][z : v]$ is precisely the sum of the values of those parse-trees with (w, z) on its rightmost path.

[4] v-snipped parse-trees rooted at u that have w on its rightmost path

Since the rightmost path of any v-snipped parse-tree rooted at u has a *unique* edge $(w,z) \in \mathcal{F}^{\times}_{m,v} \cup \mathcal{F}^{+}_{m,v}$, summing over all such potential edges gives

$$[u:v] = \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u:w]\cdot[w_L]\cdot[z:v] \;+\; \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v].$$

5 Balancing syntactically multilinear circuits

In this section, we prove the following theorem.

▶ **Theorem 20.** *Suppose C is an algebraic circuit of size s. Then, there is a circuit C' of size* $\mathrm{poly}(s)$ *computing the same polynomial with the following structural properties.*

- *all addition gates in C' have fan-in $O(s^4)$,*
- *all multiplication gates in C' have fan-in at most 5,*
- *for any multiplication gate $g \in C'$, any child h of g satisfies $|\mathrm{Var}(h)| \leq |\mathrm{Var}(g)|/2$.*

Furthermore, if C is syntactically multi-k-ic, then so is C'.

Proof. Without loss of generality, we may assume that the circuit is *right-heavy* in the sense that for every multiplication gate $u = u_1 \times u_2$ we have $|\mathrm{Var}(u_2)| \geq |\mathrm{Var}(u_1)|$. We shall build a new circuit C' that computes all $[u:v]$'s and $[u]$'s for gates $u, v \in C$ using the equations in Lemma 19.

We shall assume inductively that we have already computed all $[w]$'s with $|\mathrm{Var}(w)| < t$ and also all $[w,v]$ with $|\mathrm{Var}(w,v)| < t$. Suppose $u \in C$ such that $|\mathrm{Var}(u)| = t$. Using (1) from Lemma 19 with $m = t/2$ we have

$$[u] = \sum_{(w,z)\in\mathcal{F}^{\times}_{m}} [u:w]\cdot[w_L]\cdot[z] \;+\; \sum_{(w,z)\in\mathcal{F}^{+}_{m}} [u:w]\cdot[z].$$

By Observation 16, $|\mathrm{Var}(w)| \geq t/2$ implies that $|\mathrm{Var}(u:w)| \leq t/2$. Furthermore, $|\mathrm{Var}(z)| \leq t/2$ by the choice of the frontier edge and $|\mathrm{Var}(w_L)| \leq t/2$ since C is right-heavy. This allows us to compute all nodes of the form $[u]$ with $|\mathrm{Var}(u)| \leq t$.

If $u, v \in C$ such that $|\mathrm{Var}(u:v)| = t$. Using (2) from Lemma 19 with $m = t/2$, we have

$$[u:v] = \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u:w]\cdot[w_L]\cdot[z:v] \;+\; \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v].$$

We can restrict the edges in the RHS to only those edges (w,z) that is present in at least one v-snipped parse-tree rooted at u (if not, this edge's contribution to the RHS is zero). Therefore by Observation 16, $\mathrm{Var}(w:v) + \mathrm{Var}(u:w) \preceq \mathrm{Var}(u:v)$ and therefore we have $|\mathrm{Var}(u:w)| \leq t/2$. Furthermore, by the choice of the frontier, we also have $|\mathrm{Var}(z:v)| \leq t/2$. The non-trivial case is $\mathrm{Var}(w_L)$ which could in principle be large but again $\mathrm{Var}(w_L) \preceq \mathrm{Var}(w:v) \preceq \mathrm{Var}(u:v)$ as any parse-tree rooted w_L is a sub-tree of a v-snipped tree rooted at u. Since we have already computed all gates $[w]$ with $\mathrm{Var}(w) \leq t$, we can write

$$\begin{aligned}[u:v] &= \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u:w]\cdot[w_L]\cdot[z:v] \;+\; \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v] \\ &= \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u:w]\cdot\left(\sum_{(p,q)\in\mathcal{F}^{\times}_{m w}} [w_L:p]\cdot[p_L]\cdot[q] \;+\; \sum_{(p,q)\in\mathcal{F}^{+}_{m w}} [w_L:p]\cdot[q]\right)\cdot[z:v]\end{aligned}$$

$$+ \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v],$$

where $m_w = \mathrm{Var}(w_L)/2$.

The required structural properties of C' are readily seen from the above construction. ◀

6 Reduction to depth four from balanced circuits

We now show how to reduce a balanced circuit to a depth-4 circuit. This would complete the proof of our main theorem. We shall use the notation $\Sigma\Pi\,(\Sigma\Pi)_t$ to refer to $\Sigma\Pi\Sigma\Pi$ circuits computing polynmomials of the form

$$F = \sum_i \prod_j Q_{ij},$$

with $|\mathrm{Var}(Q_{ij})| \leq t$.

The proof of this part follows the outline of a similar argument in Chillara et al. [4] of reducing to depth-4 from a balanced circuit. However, there are some differences: our potential is $|\mathrm{Var}(\,\cdot\,)|$ and not the degree (as is usually the case). Since this potential function also falls as we go from a sum (+) gate to its children, we need one more simple observation in our argument to bound the number of steps in the recursion in the proof. We now provide the details.

▶ **Lemma 21.** *Let C be a multi-k-ic circuit of size s such that every multiplication gate g in C has fan-in at most 5 and for every child h of g in C, $\mathrm{Var}(h) \leq \mathrm{Var}(g)/2$.*

Then, for any positive integer $0 \leq t \leq kn$, there is an equivalent multi-k-ic $\Sigma\Pi\,(\Sigma\Pi)_t$ circuit C' that computes the same polynomial, with the following properties:

- *the top fan-in of C' is at most $s^{O(kn/t)}$,*
- *the size of C' is at most $2^t \cdot s^{O(kn/t)}$,*
- *each of the $(+)$-gates closer to the leaves compute polynomials that are computed by gates in C.*

Proof. Since C is balanced, with product fan-in at most 5, every gate g in C can be written as

$$g = \sum_{i=1}^{s} \prod_{j=1}^{5} g_{i,j}\,, \tag{3}$$

where each $g_{i,j}$ is also computed by a gate in the circuit C, $|\,\mathrm{Var}(g_{i,j})| \leq |\,\mathrm{Var}(g)|/2$. With this notation, (3) applied on the root of C says that C, which is a syntactically multi-k-ic circuit, can be trivially written as a $\Sigma\Pi\,(\Sigma\Pi)_{kn/2}$. A natural idea would be to apply (3) on the $g_{i,j}$'s until we get a $\Sigma\Pi\,(\Sigma\Pi)_t$ circuit. All that is needed is to bound the number of summands (or the top fan-in of the resulting $\Sigma\Pi\,(\Sigma\Pi)_t$ circuit) at the end of this process. Observe that for every $i \in \{1,2,\ldots,s\}$, we could have that $\left|\mathrm{Var}(\prod_{j=1}^{5} g_{i,j})\right|$ is *much* smaller than $|\mathrm{Var}(g)|$ itself. To handle this, we shall pretend that

We will view the process as a tree in the natural way. The root of the tree corresponds to the root of the circuit, and all other nodes in the tree correspond to products of addition gates in C. The children of a node in the tree correspond to the summands in the sum of product representation of that node obtained by expanding one of its factors according to (3). The leaves of this tree are products of addition gates $\prod g'_i$ such that $|\mathrm{Var}(g'_i)| \leq t$ for each factor g'_i. The tree has a branching factor of at most s, hence it suffices to get a bound on the depth of the tree to get a bound on the number of leaves which would be the top fan-in of the $\Sigma\Pi\left(\Sigma\Pi\right)_t$ representation.

Let $g\prod_\ell w_\ell$ be an internal node in the tree with $|\mathrm{Var}(g)| > t$. After applying (3) on g, we get

$$g\left(\prod_\ell w_\ell\right) = \sum_{i=1}^{s}\left(\prod_{j=1}^{5} g_{i,j}\cdot\prod_\ell w_\ell\right).$$

We now consider two cases.

- $\left|\mathbf{Var}\left(\prod_{j=1}^{5} g_{i,j}\right)\right| < 3t/4$ **:** In this case, $\left|\mathrm{Var}\left(\prod_{j=1}^{5} g_{i,j}\cdot\prod_\ell w_\ell\right)\right| \leq \left|\mathrm{Var}\left(g\cdot\prod_\ell w_\ell\right)\right| - t/4.$
- $\left|\mathbf{Var}\left(\prod_{j=1}^{5} g_{i,j}\right)\right| \geq 3t/4$ **:** Since $\mathrm{Var}(g) \succeq \mathrm{Var}(g_{i,1}\cdots g_{i,5}) = \mathrm{Var}(g_{i,1})+\cdots+\mathrm{Var}(g_{i,5})$ and $|\mathrm{Var}(g_{i,j})| \leq t/2$, it follows that the number of factors h in $\prod_{j=1}^{5} g_{i,j}\cdot\prod_\ell w_\ell$ with $|\mathrm{Var}(h)| \geq t/16$ is at least one more than the number of such factors in $g\cdot\prod_\ell w_\ell$. This is because besides the factor $g_{i,j}$ with largest $|\mathrm{Var}(g_{i,j})|$, the other four factors together must contribute at least $(3t/4)-(t/2) = (t/4)$ to $|\mathrm{Var}(g_{i,1}\cdots g_{i,5})|$ and hence at least one of them must have $|\mathrm{Var}(g_{i,k})| \geq t/16$.

Thus, in any edge of the tree, either $|\mathrm{Var}(\,\cdot\,)|$ decreases by $t/4$ or the number of factors with $|\mathrm{Var}(\,\cdot\,)| \geq t/16$ increases by one. The root node g_0 has $|\mathrm{Var}(g_0)| \leq kn$. Hence, the depth of the tree is bounded by $(16+4)(kn/t) = O(nk/t)$. Therefore, C can be computed by a syntactically multi-k-ic $\Sigma\Pi\left(\Sigma\Pi\right)_t$ circuit of top fan-in at most $s^{O(nk/t)}$.

To get the bound on the overall size of the $\Sigma\Pi\left(\Sigma\Pi\right)_t$ circuit, we need to bound the sparsity of the polynomials computed by bottom two layers. Note that if $\mathrm{Var}(f) = (d_1,\ldots,d_n)$, then f can have at most $\prod(1+d_i)$ monomials. Since $2^x \geq 1+x$ for all positive integers x, it follows that $|\mathrm{Var}(f)| \leq t$ implies that f has at most 2^t monomials. Therefore, the total size of the $\Sigma\Pi\left(\Sigma\Pi\right)_t$ circuit is $2^t\cdot s^{O(kn/t)} = 2^{O\left(t+\frac{kn\log s}{t}\right)}$. ◀

From Theorem 20 and setting $t = \sqrt{kn\log s}$ in Lemma 21, we get Theorem 7 restated below.

▶ **Theorem 7.** *Let C be a multi-k-ic circuit of size s computing a polynomial in n variables. Then, there is a multi-k-ic $\Sigma\Pi\Sigma\Pi$ circuit C' of size $s^{O\left(\sqrt{\frac{kn}{\log s}}\right)}$ computing the same polynomial.*

6.1 Reduction to higher depths

We now prove Theorem 8 which shows that similar savings can be obtained in depth reductions to larger depth.

▶ **Theorem 8.** *Let C be a multi-k-ic circuit of size s computing a polynomial in n variables. Then, there is a multi-k-ic $(\Sigma\Pi)^\Delta$ circuit C' computing the same polynomial whose size is at most*

$$s^{O\left(\Delta\cdot(nk/\log s)^{1/\Delta}\right)}.$$

Proof of Theorem 8. We shall assume, without loss of generality, that the circuit C is balanced (by applying Theorem 20 if necessary). The proof follows via repeated applications of Lemma 21.

Applying Lemma 21 with $t = nk/(nk/\log s)^{1/\Delta}$, we obtain a $\Sigma\Pi\,(\Sigma\Pi)_t$ circuit C' of the form

$$C' = \sum_{i=1}^{s'} \prod_{j} g_{ij},$$

with $s' = s^{O\left((kn/\log s)^{1/\Delta}\right)}$ and $|\mathrm{Var}(g_{ij})| \le t$ for all i, j. Furthermore, since each g_{ij} being a polynomial computed by a gate in C, they are computable by multi-k-ic circuits of size at most s. By induction, each g_{ij} has a multi-k-ic $(\Sigma\Pi)^{\Delta-1}$ circuit of size at most

$$s^{O\left((\Delta-1)\cdot(t/\log s)^{1/(\Delta-1)}\right)} = s^{O\left((\Delta-1)\cdot(nk/\log s)^{1/\Delta}\right)}.$$

Replacing each g_{ij} by this circuit, we obtain a $(\Sigma\Pi)^{\Delta}$ circuit of size at most

$$s' \cdot s^{O\left((\Delta-1)\cdot(nk/\log s)^{1/\Delta}\right)} = s^{O\left(\Delta\cdot(kn/\log s)^{1/\Delta}\right)}. \qquad ◀$$

7 Open problems

The most interesting question that comes out of this work is to prove a lower bound of $n^{\omega(\sqrt{n/\log n})}$ for syntactically multilinear circuits of depth-4 for an explicit polynomial. A natural and first approach to this could be to understand if the shifted partials based methods can prove a lower bound of $n^{\Omega(\sqrt{d})}$ for homogeneous depth-4 circuits for a polynomial family with degree $d = \omega(n/\log n)$.

Another question of interest would be to understand the *correct* exponent for the depth reduction results to depth-4 (and also to higher depth) for various regimes of the degree d. From [9], we know that for $d = O(n^{\varepsilon})$ for a small enough constant ε, $\sqrt{d}$ is the correct exponent, whereas for d being nearly n, the results in this paper and those of Raz and Yehudayoff [12] show that the correct exponent is $\sqrt{n/\log n}$. But we do not understand this phenomenon for other values of d.

References

1 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. doi:10.1109/FOCS.2008.32.

2 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. doi:10.1016/S0304-3975(97)00227-2.

3 Noga Alon, Mrinal Kumar, and Ben Lee Volk. Unbalancing Sets and an Almost Quadratic Lower Bound for Syntactically Multilinear Arithmetic Circuits. In *CCC 2018*, pages 1–16, 2018. arXiv:1708.02037. doi:10.4230/LIPIcs.CCC.2018.11.

4 Suryajith Chillara, Mrinal Kumar, Ramprasad Saptharishi, and V. Vinay. The Chasm at Depth Four, and Tensor Rank : Old results, new insights. *CoRR*, 2016. arXiv:1606.04200.

5 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *Journal of the ACM*, 61(6):33:1–33:16, 2014. Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*. doi:10.1145/2629541.

6 Sumant Hegde and Chandan Saha. Improved Lower Bound for Multi-r-ic Depth Four Circuits as a Function of the Number of Input Variables. *Proceedings of Indian National Science Academy*, 83(4):907–922, 2017. doi:10.16943/ptinsa/2017/49224.

7 Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. In *Electronic Colloquium on Computational Complexity (ECCC)TR12-081*, 2012. URL: http://eccc.hpi-web.de/report/2012/081/.

8 Pascal Koiran. Arithmetic Circuits: The Chasm at Depth Four Gets Wider. *Theoretical Computer Science*, 448:56–65, 2012. doi:10.1016/j.tcs.2012.03.041.

9 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, pages 364–373, 2014. doi:10.1109/FOCS.2014.46.

10 Ran Raz. Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size. *J. ACM*, 56(2):8:1–8:17, 2009. Preliminary version in the *36th Annual ACM Symposium on Theory of Computing (STOC 2004)*. doi:10.1145/1502793.1502797.

11 Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. Preliminary version in the *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*. doi:10.1137/070707932.

12 Ran Raz and Amir Yehudayoff. Lower Bounds and Separations for Constant Depth Multilinear Circuits. *Computational Complexity*, 18(2):171–207, 2009. Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*. doi:10.1007/s00037-009-0270-8.

13 Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. doi:10.1561/0400000039.

14 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. Preliminary version in the *38th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2013)*. doi:10.1016/j.ic.2014.09.004.

15 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM J. Comput.*, 12(4):641–644, 1983. Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*. doi:10.1137/0212043.

A Proof of Lemma 19

Proof of Lemma 19. The proof shall proceed by induction on the height of u (leaves are at height 0). We shall present the proof of (2); the proof of (1) is analogous.

Case 1: $u = u_L \times u_R$. For any w, we have that $[u : w] = 1$ if $u = w$, and $[u : w] = [u_1] \cdot [u_2 : w]$ whenever $u \neq w$. In particular, since $|\mathrm{Var}(v)| < m \leq |\mathrm{Var}(u)|$ the LHS is $[u : v] = [u_L] \cdot [u_R : v]$.

If $|\mathrm{Var}(u_R)| \geq m$, then for any $(w, z) \in \mathbb{F}^{+}_{m,v}$ or $\mathcal{F}^{\times}_{m,v}$ we have $w \neq u$. Inducting on u_R,

$$\begin{aligned}
\text{LHS} &= [u_L] \cdot [u_R : v] \\
&= [u_L] \cdot \left(\sum_{(w,z) \in \mathcal{F}^{\times}_{m,v}} [u_R : w] \cdot [w_L] \cdot [z : v] \; + \sum_{(w,z) \in \mathcal{F}^{+}_{m,v}} [u_R : w] \cdot [z : v] \right) \\
&= \sum_{(w,z) \in \mathcal{F}^{\times}_{m,v}} [u_L] \cdot [u_R : w] \cdot [w_L] \cdot [z : v] \; + \sum_{(w,z) \in \mathcal{F}^{+}_{m,v}} [u_L] \cdot [u_R : w] \cdot [z : v] \\
&= \sum_{(w,z) \in \mathcal{F}^{\times}_{m,v}} [u : w] \cdot [w_L] \cdot [z : v] \; + \sum_{(w,z) \in \mathcal{F}^{+}_{m,v}} [u : w] \cdot [z : v] \; = \; \text{RHS.}
\end{aligned}$$

On the other hand, if $|\mathrm{Var}(u_R)| < m$ then $[u : w] = 0$ for any $w \neq u$ with $|\mathrm{Var}(w)| \geq m$. Hence,

$$\begin{aligned}\text{RHS} &= \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u:w]\cdot[w_L]\cdot[z:v] \;+ \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v]\\ &= [u:u]\cdot[u_L]\cdot[u_R:v] = [u:v] = \text{LHS}.\end{aligned}$$

Case 2: $u = u_1 + u_2$. For any w, we have that $[u : w] = 1$ if $u = w$, and $[u : w] = [u_1 : w] + [u_2 : w]$ whenever $u \neq w$. In particular, since $|\mathrm{Var}(v)| < m \leq |\mathrm{Var}(u)|$ the LHS is $[u : v] = [u_1 : v] + [u_2 : v]$.

Since u is a + gate, $(u, u_j) \notin \mathcal{F}^{\times}_{m,v}$ for any j. If $|\mathrm{Var}(u_j)| < m$ for some j, then the edge $(u, u_j) \in \mathcal{F}^{+}_{m,v}$. Hence,

$$\begin{aligned}\text{RHS} &= \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u:w]\cdot[w_L]\cdot[z:v] \;+ \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v]\\ &=: \qquad\qquad T_1 \qquad\qquad + \qquad\qquad T_2\end{aligned}$$

In T_1, since every $(w, z) \in \mathcal{F}^{\times}_{m,v}$ has $w \neq u$ we have

$$\begin{aligned}T_1 &:= \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} \left(\sum_i [u_i:w]\right)\cdot[w_L]\cdot[z:v]\\ &= \sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} \left(\sum_{i:|\mathrm{Var}(u_i)|\geq m} [u_i:w]\right)\cdot[w_L]\cdot[z:v] \qquad \text{(since } [u_j:w]=0 \text{ if } |\mathrm{Var}(u_j)|<m)\\ &= \sum_{i:|\mathrm{Var}(u_i)|\geq m}\;\sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u_i:w]\cdot[w_L]\cdot[z:v].\end{aligned}$$

As for the other term, it can be written as

$$\begin{aligned}T_2 &:= \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u:w]\cdot[z:v]\\ &= \sum_{\substack{(w,z)\in\mathcal{F}^{+}_{m,v}\\ w\neq u}} [u:w]\cdot[z:v] \;+ \sum_{j:|\mathrm{Var}(u_j)|<m} [u:u]\cdot[u_j:v]\\ &= \sum_{\substack{(w,z)\in\mathcal{F}^{+}_{m,v}\\ w\neq u}} \left(\sum_i [u_i:w]\right)\cdot[z:v] \;+ \sum_{j:|\mathrm{Var}(u_j)|<m} [u_j:v]\\ &= \sum_{\substack{(w,z)\in\mathcal{F}^{+}_{m,v}\\ w\neq u}} \left(\sum_{i:|\mathrm{Var}(u_i)|\geq m} [u_i:w]\right)\cdot[z:v] \;+ \sum_{j:|\mathrm{Var}(u_j)|<m} [u_j:v]\\ &= \sum_{i:|\mathrm{Var}(u_i)|\geq m}\;\sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u_i:w]\cdot[z:v] \;+ \sum_{j:|\mathrm{Var}(u_j)|<m} [u_j:v].\end{aligned}$$

The last equality holds because $[u_i : u] = 0$. Putting it together,

$$\begin{aligned}\text{RHS} &= T_1 + T_2\\ &= \sum_{i:|\mathrm{Var}(u_i)|\geq m} \left(\sum_{(w,z)\in\mathcal{F}^{\times}_{m,v}} [u_i:w]\cdot[w_L]\cdot[z:v] + \sum_{(w,z)\in\mathcal{F}^{+}_{m,v}} [u_i:w]\cdot[z:v]\right)\end{aligned}$$

$$\begin{aligned}
&\quad + \sum_{j:|\mathrm{Var}(u_j)|<m} [u_j : v] \\
&= \sum_{i:|\mathrm{Var}(u_i)|\geq m} [u_i : v] + \sum_{j:|\mathrm{Var}(u_j)|<m} [u_j : v] \qquad \text{(induction)} \\
&= [u : v] = \mathrm{LHS}.
\end{aligned}$$

◀