On Algebraic Branching Programs of Small Width

KARL BRINGMANN and CHRISTIAN IKENMEYER, Max-Planck-Institut für Informatik
JEROEN ZUIDDAM, Centrum Wiskunde & Informatica

In 1979, Valiant showed that the complexity class $\mathbf{VP_e}$ of families with polynomially bounded formula size is contained in the class $\mathbf{VP_s}$ of families that have algebraic branching programs (ABPs) of polynomially bounded size. Motivated by the problem of separating these classes, we study the topological closure $\overline{\mathbf{VP_e}}$, i.e., the class of polynomials that can be approximated arbitrarily closely by polynomials in $\mathbf{VP_e}$. We describe $\overline{\mathbf{VP_e}}$ using the well-known continuant polynomial (in characteristic different from 2). Further understanding this polynomial seems to be a promising route to new formula size lower bounds.

Our methods are rooted in the study of ABPs of small constant width. In 1992, Ben-Or and Cleve showed that formula size is polynomially equivalent to width-3 ABP size. We extend their result (in characteristic different from 2) by showing that approximate formula size is polynomially equivalent to approximate width-2 ABP size. This is surprising because in 2011 Allender and Wang gave explicit polynomials that cannot be computed by width-2 ABPs at all! The details of our construction lead to the aforementioned characterization of $\overline{\mathbf{VP_e}}$.

As a natural continuation of this work, we prove that the class **VNP** can be described as the class of families that admit a hypercube summation of polynomially bounded dimension over a product of polynomially many affine linear forms. This gives the first separations of algebraic complexity classes from their nondeterministic analogs.

CCS Concepts: • **Theory of computation → Algebraic complexity theory**;

Additional Key Words and Phrases: Algebraic branching programs, algebraic complexity theory, border complexity, formula size, iterated matrix multiplication

ACM Reference format:
Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. 2018. On Algebraic Branching Programs of Small Width. *J. ACM* 65, 5, Article 32 (August 2018), 29 pages.
https://doi.org/10.1145/3209663

1 INTRODUCTION

32

Separating complexity classes and more generally proving computational complexity lower bounds is the infamous key objective in complexity theory. In an approach to prove complexity lower bounds, in 1979 Valiant [40] proposed an algebraic alternative to the classical Boolean circuit model of computation: Arithmetic formulas.

Arithmetic Formulas and the Class $\mathbf{VP_e}$. Fix a field $\mathbb{F}$. An arithmetic formula is defined as a rooted binary tree whose leaves are each labeled with a variable or a field constant and whose root and

Authors' addresses: K. Bringmann; C. Ikenmeyer, Max-Planck-Institut für Informatik, Saarland Informatics Campus, Campus E1 4, 66123, Saarbrücken, Germany; emails: {kbringma, cikenmey}@mpi-inf.mpg.de; J. Zuiddam, Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, Netherlands; email: j.zuiddam@cwi.nl.

https://doi.org/10.1145/3209663

intermediate vertices (called *gates*) are labeled with either "+" (addition) or "×" (multiplication). In the natural way, via induction over the tree structure, an arithmetic formula computes a multivariate polynomial f. The *formula size* of a multivariate polynomial f is defined as the smallest number of gates required for a formula to compute f.

A sequence (m_n) of natural numbers is called *polynomially bounded* if there exists a univariate polynomial q such that $m_n \le q(n)$ for all n. A sequence of multivariate polynomials (f_n) is called a *family*. Valiant [40] introduced the complexity class $\mathbf{VP_e}$ that is defined as the set of all families whose formula size is polynomially bounded. For example, the family $((x_1)^n + (x_2)^n + \cdots (x_n)^n) \in \mathbf{VP_e}$, because its formula size grows at most quadratically.

The smallest known formulas for the determinant family $\det_n := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n x_{i,\sigma(i)}$ have size $n^{O(\log n)}$. This follows from Berkowitz' algorithm [4], which gives an algebraic circuit of depth $O(\log^2 n)$, and thus by expanding we get an algebraic formula of depth $O(\log^2 n)$ whose size is then trivially bounded by $2^{O(\log^2 n)} = n^{O(\log n)}$. It is a major open question in algebraic complexity theory whether formulas of polynomially bounded size exist for $\det_n$. This question can be phrased in terms of complexity classes as asking whether or not the inclusion $\mathbf{VP_e} \subseteq \mathbf{VP_s}$ is strict.

Motivated by this question, we study the closure class $\overline{\mathbf{VP_e}}$ of families of polynomials that can be approximated arbitrarily closely by families in $\mathbf{VP_e}$ (see Section 2 for a formal definition). Over the field $\mathbb{R}$ or $\mathbb{C}$, one can think of $\overline{\mathbf{VP_e}}$ as the set of families whose *border formula size* is polynomially bounded, where the border formula size of a polynomial f is defined as the smallest c such that there exists a sequence g_i of polynomials with formula size at most c that satisfy $\lim_{i\to\infty} g_i = f$. In this article, we present a simple description of $\overline{\mathbf{VP_e}}$ and show that the *continuant polynomial* F_n is $\overline{\mathbf{VP_e}}$-complete, given the characteristic is not 2, see Theorem 3.12 below. The continuant has rich algebraic properties, which are expected to be useful in the future to prove complexity lower bounds.

The Continuant. The *continuant* F_n can be succinctly defined via $F_0 := 1$, $F_1 := x_1$, $F_n := x_n F_{n-1} + F_{n-2}$; see Section 3. We prove that F_n is $\overline{\mathbf{VP_e}}$-complete under p-degenerations: This means that every family (f_n) in $\overline{\mathbf{VP_e}}$ can be obtained as the pointwise limit of a sequence $f_n = \lim_{j\to\infty} F_{t(n)}(\ell_1(j), \ldots, \ell_{t(n)}(j))$, where each $\ell_i(j)$ is a variable or constant and $t(n)$ is a polynomially bounded function. The continuant is arguably the simplest $\overline{\mathbf{VP_e}}$-complete polynomial known today. Prior to our work, the simplest $\overline{\mathbf{VP_e}}$-complete (and $\mathbf{VP_e}$-complete) polynomial was the iterated 3×3 matrix multiplication polynomial [3]. This simple new polynomial immediately motivates the definition of the *border continuant complexity* $\underline{L}_{\mathrm{Con}}(f)$ of a polynomial f, which is the smallest number c such that f can be obtained as $\lim_{j\to\infty}(F_c(\ell_1(j), \ldots, \ell_c(j)))_j$. To make the situation more geometric, we allow the $\ell_i(j)$ to be arbitrary affine linear forms (i.e., polynomials of degree 1). Our results show that border continuant complexity is polynomially equivalent to border formula size. This insight is quite striking because a result of Allender and Wang [1] implies that the continuant complexity *without allowing approximations* can be infinite!

Continuous Lower Bounds. In algebraic complexity theory, the way of showing a complexity lower bound for a problem $f \in V$ for some $\mathbb{F}$-vector space V most often goes by (implicitly or explicitly) finding a function $\mathcal{F} : V \to \mathbb{F}$ that is zero on all problems of low complexity while at the same time $\mathcal{F}(f) \neq 0$. Grochow [17] gives a long list (see, e.g., References [13, 20, 23, 25, 32, 34]) of settings where complexity lower bounds are obtained in this way. Moreover, he points out that over the complex numbers these functions $\mathcal{F}$ can be assumed to be continuous (and even to be so-called highest-weight vector polynomials). If $\mathbf{C}$ and $\mathbf{D}$ are algebraic complexity classes with $\mathbf{C} \subseteq \mathbf{D}$ (for example, $\mathbf{C} = \mathbf{VP_e}$ and $\mathbf{D} = \mathbf{VP_s}$), then any separation of algebraic complexity classes $\mathbf{C} \neq \mathbf{D}$ in this continuous manner would automatically imply the stronger statement $\mathbf{D} \not\subseteq \overline{\mathbf{C}}$. It is therefore natural to try to prove the separation $\mathbf{VP}_s \not\subseteq \overline{\mathbf{VP}_e}$ instead of the slightly weaker $\mathbf{VP}_e \neq \mathbf{VP}_s$, which provides further motivation for studying $\overline{\mathbf{VP}_e}$. This is exactly analogous to Mulmuley and Sohoni's geometric complexity approach (see, e.g., References [29, 30] and the exposition Reference [14, Section 9]) where one tries to prove the separation $\mathbf{VNP} \not\subseteq \overline{\mathbf{VP}_s}$ to attack Valiant's famous $\mathbf{VP}_s \neq \mathbf{VNP}$ conjecture [40]. Here, **VNP** is the class of p-definable families; see Section 2 for a precise definition.

A Remark on Algebraic Geometry and Group Actions. A promising path toward proving formula lower bounds, for example, for the determinant or the permanent $\text{per}_n := \sum_{\sigma \in S_n} \prod_{i=1}^n x_{i,\sigma(i)}$, is to apply to our setting the following standard geometric ideas. If we take our field to be the complex numbers and fix the number of variables n and the degree d, then the set of homogeneous degree d polynomials $\mathbb{C}[x_1, \ldots, x_n]_d$ contains the set

$$X_c := \{f \in \mathbb{C}[x_1, \ldots, x_n]_d \mid \underline{L}_{\text{Con}}(f) \leq c\}$$

as an affine subvariety (X_c is the closure of the set of affine projections of F_c intersected with $\mathbb{C}[x_1, \ldots, x_n]_d$). Moreover, since we allowed the $\ell_i(j)$ to be affine linear forms, the group $\text{GL}(\mathbb{C}^n)$ acts canonically on X_c, making X_c an affine $\text{GL}(\mathbb{C}^n)$-variety. If we find a polynomial $\mathcal{F}$ that vanishes identically on X_c, then a nonzero evaluation $\mathcal{F}(f) \neq 0$ implies that $\underline{L}_{\text{Con}}(f) > c$. This approach looks feasible given the very simple structure of the continuant polynomial. This is emphasized by the fact that the action of $\text{GL}(\mathbb{C}^n)$ puts a lot of structure on the coordinate ring of X_c (see, e.g., References [2, 12, 13, 19, 21, 25, 33]), where the action of the general linear group on the coordinate ring of a variety is used to classify some of its defining equations.

1.1 Main Results

Algebraic Branching Programs (ABPs) of Width 2. Our main objects of study are the following classes of families of polynomials: the class of families of polynomials with polynomially bounded formula size $\mathbf{VP}_e$, its closure $\overline{\mathbf{VP}_e}$, and the nondeterministic variant **VNP** (see Section 2). We do so by studying algebraic branching programs of small width. These are defined as follows. An *algebraic branching program* (ABP) is a directed acyclic graph with a source vertex s and a sink vertex t that has affine linear forms over the base field $\mathbb{F}$ as edge labels. Moreover, we require that each vertex is labeled with an integer (its *layer*) and that edges in the ABP only point from vertices in layer i to vertices in layer $i + 1$. The *width* of an ABP is the cardinality of its largest layer. The *size* of an ABP is the number of its vertices. The *value* of an ABP is the sum of the values of all s-t-paths, where the value of an s-t-path is the product of its edge labels. We say that an ABP *computes* its value. The class $\mathbf{VP}_s$ coincides with the class of families of polynomials that can be computed by ABPs of polynomially bounded size (see, e.g., Reference [37].

For this article, we introduce the class $\mathbf{VP}_k$, $k \in \mathbb{N}$, which is defined as the class of families of polynomials computable by width-k ABPs of polynomially bounded size. It is a well-known simple exercise (see, e.g., Reference [8, Proposition 7.1] for a proof with all details) that for every $k \geq 1$,

$$\mathbf{VP}_k \subseteq \mathbf{VP}_e. \tag{1}$$

In 1992, Ben-Or and Cleve [3] showed that $\mathbf{VP}_k = \mathbf{VP}_e$ for all $k \geq 3$ (we review the proof, see Theorem B.1). In 2011, Allender and Wang [1] showed that width-2 ABPs cannot compute every polynomial, so in particular we have a strict inclusion $\mathbf{VP}_2 \subsetneq \mathbf{VP}_3$. Let the characteristic of the base field $\mathbb{F}$ be different from 2. Our first main result (Theorem 3.1 and Corollary 3.9) is that the closure of $\mathbf{VP}_2$ and the closure of $\mathbf{VP}_e$ are equal,

$$\overline{\mathbf{VP}_2} = \overline{\mathbf{VP}_e}. \tag{2}$$

Interestingly, as a direct corollary of Equation (2) and the result of Allender and Wang, the inclusion $\mathbf{VP}_2 \subsetneq \overline{\mathbf{VP}_2}$ is strict. It is easy to see that $\mathbf{VP}_1$ equals $\overline{\mathbf{VP}_1}$ (Proposition A.12), so $\mathbf{VP}_1$ and $\mathbf{VP}_2$ are examples of quite similar algebraic complexity classes that behave differently under closure. Most importantly, from the proof of Equation (2) we obtain our results about the continuant polynomial that we mentioned before.

VNP via Affine Linear Forms. To every algebraic complexity class there exists a natural nondeterministic analogoue (see Section 2 for the formal definition). Classically, the nondeterministic analogue to **VP** is called **VNP**, and the analogue to $\mathbf{VP}_\mathrm{e}$ is called $\mathbf{VNP}_\mathrm{e}$. We define the classes $\mathbf{VNP}_\mathrm{e}$ and **VNP** in the natural way. In 1980, Valiant [41] showed that $\mathbf{VNP}_\mathrm{e} = \mathbf{VNP}$ and in this article we will always view **VNP** as the nondeterministic analog of $\mathbf{VP}_\mathrm{e}$. To $\mathbf{VP}_1$ and $\mathbf{VP}_2$ we analogously associate nondeterministic analogs $\mathbf{VNP}_1$ and $\mathbf{VNP}_2$. Using interpolation techniques it is possible to deduce $\mathbf{VNP}_2 = \mathbf{VNP}$ from Equation (2), provided the field is infinite. Using more sophisticated techniques, we strengthen this result to get our second main result (Theorem 4.2):

$$\mathbf{VNP}_1 = \mathbf{VNP}. \tag{3}$$

This can be succinctly stated as: a family (f_n) is contained in **VNP** iff f_n can be written as a hypercube summation of polynomially bounded dimension over a product of polynomially many affine linear forms. Using Equation (3) it is then easy to verify that $\mathbf{VP}_1 \subsetneq \mathbf{VNP}_1$ and using Reference [1] yields $\mathbf{VP}_2 \subsetneq \mathbf{VNP}_2$, which separates complexity classes from their nondeterministic analogs. Interestingly, $\mathbf{VNP}_1 \subsetneq \mathbf{VNP}$ over the field with two elements; see Section 5.

Restricted ABP Edge Labels. Several more results on small-width ABPs, approximation closures, and hypercube summations are proved throughout this article. For example, in Appendix A we investigate the subtleties of what happens if we restrict the ABP edge labels to simple affine linear forms, or to variables and constants. The precise relations between complexity classes that we obtain are listed in Figure 8. As another example, we strengthen Equation (3) as follows (Theorem B.3): A family (f_n) is contained in **VNP** iff f_n can be written as a hypercube summation of polynomially bounded dimension over a product of polynomially many affine linear forms that use *at most two variables* each.

1.2 Further Related Work

An excellent exposition on the history of small-width computation can be found in Reference [1], along with an explicit polynomial that cannot be computed by width-2 ABPs: $x_1x_2 + x_3x_4 + \cdots + x_{15}x_{16}$. Saha, Saptharishi, and Saxena [36, Corollary 14] showed that $x_1x_2 + x_3x_4 + x_5x_6$ cannot be computed by width-2 ABPs that correspond to the iterated matrix multiplication of upper triangular matrices.

Bürgisser [10] studied approximations in the model of general algebraic circuits, finding general upper bounds on the error degree. For most specific algebraic complexity classes **C**, the relation between **C** and $\overline{\mathbf{C}}$ has not been an active object of study. As pointed out recently by Forbes [16], Nisan's result [31] implies that $\mathbf{C} = \overline{\mathbf{C}}$ for **C** being the class of size-k algebraic branching programs on noncommuting variables. Recently, a structured study of $\overline{\mathbf{VP}}$ and $\overline{\mathbf{VP}_\mathrm{s}}$ has been started; see Reference [18]. By far the most work in lower bounds for topological approximation algorithms has been done in the area of bilinear complexity, dating back to References [6, 26, 38] and more recently, e.g., References [21, 22, 24, 25, 42].

1.3 Paper Outline

In Section 2, we introduce in more detail the approximation closure and the nondeterminism closure of a complexity class. In Section 3, we prove the first main result: border formula size is polynomially equivalent to border width-2 ABP size and the continuant is $\overline{\mathbf{VP}_e}$-complete under p-degenerations. In Section 4, we prove the second main result: a new description of **VNP** as the nondeterminism closure of families that have polynomial-size width-1 ABPs. The later sections contain details on how to strengthen the result from Section 4 and results on the power of ABPs with restricted edge labels.

2 NONDETERMINISM AND APPROXIMATION CLOSURE

In this section, we introduce the approximation closure and the nondeterministic analog of a class. A *family* is a sequence of polynomials $(f_n)_{n\in\mathbb{N}}$. A *class* is a set of families and will be written in boldface, **C**. For an introduction to the algebraic complexity classes $\mathbf{VP}_e$, **VP**, and **VNP**, we refer the reader to Reference [11]. We denote by $\mathrm{poly}(n)$ the set of polynomially bounded functions $\mathbb{N} \to \mathbb{N}$. We define the norm of a complex multivariate polynomial as the sum of the absolute values of its coefficients. This defines a topology on the polynomial ring $\mathbb{C}[x_1, \ldots, x_m]$. Given a complexity measure L, say ABP size or formula size, there is a natural notion of approximate complexity that is called *border complexity*. Namely, a polynomial $f \in \mathbb{C}[\mathbf{x}]$ has *border complexity* $\underline{L}^{\mathrm{top}}$ at most c if there is a sequence of polynomials $g_1, g_2, \ldots$ in $\mathbb{C}[\mathbf{x}]$ converging to f such that each g_i satisfies $L(g_i) \leq c$. It turns out that for reasonable classes over the field of complex numbers $\mathbb{C}$, this *topological* notion of approximation is equivalent to what we call *algebraic* approximation (see, e.g., Reference [10]). Namely, a polynomial $f \in \mathbb{C}[\mathbf{x}]$ satisfies $\underline{L}(f)^{\mathrm{alg}} \leq c$ iff there are polynomials $f_1, \ldots, f_e \in \mathbb{C}[\mathbf{x}]$ such that the polynomial

$$h := f + \varepsilon f_1 + \varepsilon^2 f_2 + \cdots + \varepsilon^e f_e \in \mathbb{C}[\varepsilon, \mathbf{x}]$$

has complexity $L_{\mathbb{C}(\varepsilon)}(h) \leq c$, where ε is a formal variable and $L_{\mathbb{C}(\varepsilon)}(h)$ denotes the complexity of h over the field extension $\mathbb{C}(\varepsilon)$. This algebraic notion of approximation makes sense over any base field and we will use it in the statements and proofs of this article.

Definition 2.1. Let $\mathbf{C}(\mathbb{F})$ be a class over the field $\mathbb{F}$. We define the *approximation closure* $\overline{\mathbf{C}}(\mathbb{F})$ as follows: a family (f_n) over $\mathbb{F}$ is in $\overline{\mathbf{C}}(\mathbb{F})$ if there are polynomials $f_{n;i}(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ and a function $e : \mathbb{N} \to \mathbb{N}$ such that the family (g_n) defined by

$$g_n(\mathbf{x}) := f_n(\mathbf{x}) + \varepsilon f_{n;1}(\mathbf{x}) + \varepsilon^2 f_{n;2}(\mathbf{x}) + \cdots + \varepsilon^{e(n)} f_{n;e(n)}(\mathbf{x})$$

is in $\mathbf{C}(\mathbb{F}(\varepsilon))$. We define the *poly-approximation closure* $\overline{\mathbf{C}}^{\mathrm{poly}}(\mathbb{F})$ similarly, but with the additional requirement that $e(n) \in \mathrm{poly}(n)$. We call $e(n)$ the *error degree*.

Remark 2.2. In the algebraic complexity theory literature, error degree was already studied by Bini [5]. An alternative "quality measure" for approximative computation used in the literature is the *order of approximation*. To define order of approximation one replaces the field $\mathbb{F}(\varepsilon)$ in Definition 2.1 with the ring of power series $\mathbb{F}[\![\varepsilon]\!]$ and one aims for an equality

$$g_n(\mathbf{x}) = \varepsilon^{q_n} f_n(\mathbf{x}) + \varepsilon^{q_n+1} F_n(\mathbf{x}),$$

with $F_n(\mathbf{x}) \in \mathbb{F}[\![\varepsilon]\!][\mathbf{x}]$ and $(g_n) \in \mathbf{C}(\mathbb{F}[\![\varepsilon]\!])$. (Note that the coefficients of $F_n(\mathbf{x})$ are power series in ε.) Then we have an approximation of (f_n) with order of approximation equal to q_n; see, e.g., References [10, 14]. In this article, we work with *error degree*, since the interpolation technique that we use to transform approximate formulas into formulas relies directly on error degree (see Corollary 3.10 and Reference [8, Proposition 8.1]).

For all the complexity classes **C** considered in this article, the approximation closure operator $\mathbf{C} \mapsto \overline{\mathbf{C}}$ is idempotent. For these classes, it is a Kuratowski closure operator, i.e., $\overline{\emptyset} = \emptyset$, $\mathbf{C} \subseteq \overline{\mathbf{C}}$, $\overline{\mathbf{C} \cup \mathbf{D}} = \overline{\mathbf{C}} \cup \overline{\mathbf{D}}$, and $\overline{\overline{\mathbf{C}}} = \overline{\mathbf{C}}$.

One can think of **VNP** as a "nondeterminism closure" of **VP**. We want to use the nondeterminism closure for general classes.

Definition 2.3. Let **C** be a class. The class N(**C**) consists of families (f_n) with the following property: There is a family $(g_n) \in \mathbf{C}$ and $p(n), q(n) \in \text{poly}(n)$ such that

$$f_n(\mathbf{x}) = \sum_{\mathbf{b} \in \{0,1\}^{p(n)}} g_{q(n)}(\mathbf{b}, \mathbf{x}),$$

where $\mathbf{x}$ and $\mathbf{b}$ denote sequences of variables $x_1, x_2, \ldots$ and $b_1, b_2, \ldots, b_{p(n)}$. We will sometimes say that $f(\mathbf{x})$ is a *hypercube sum over* g and that $b_1, b_2, \ldots, b_{p(n)}$ are the *hypercube variables*. For any s, t, we will use the standard notation $\mathbf{VNP}_s^t$ to denote $\mathrm{N}(\mathbf{VP}_s^t)$, where the superscript t will become relevant in Appendix A. We remark that the map $\mathbf{C} \mapsto \mathrm{N}(\mathbf{C})$ trivially satisfies all properties of being a Kuratowski closure operator, i.e., $\mathrm{N}(\emptyset) = \emptyset$, $\mathbf{C} \subseteq \mathrm{N}(\mathbf{C})$, $\mathrm{N}(\mathbf{C} \cup \mathbf{D}) = \mathrm{N}(\mathbf{C}) \cup \mathrm{N}(\mathbf{D})$, and $\mathrm{N}(\mathrm{N}(\mathbf{C})) = \mathrm{N}(\mathbf{C})$.

3 APPROXIMATE WIDTH-2 ABPS AND FORMULA SIZE

As mentioned in the Introduction, Allender and Wang [1] showed that there exist polynomials that cannot be computed by any width-2 ABP, for example, the polynomial $x_1x_2 + x_3x_4 + \cdots + x_{15}x_{16}$. Therefore, we have a separation $\mathbf{VP}_2 \subsetneq \mathbf{VP}_3 = \mathbf{VP}_e$. We show that allowing approximation changes the situation completely: every polynomial can be approximated by a width-2 ABP. In fact, every polynomial can be approximated by a width-2 ABP of size polynomial in the formula size and with error degree polynomial in the formula size. This is the main result of this section.

Theorem 3.1. $\mathbf{VP}_e \subseteq \overline{\mathbf{VP}_2}^{\text{poly}}$ *when* $\text{char}(\mathbb{F}) \neq 2$.

We leave as an open question what happens in characteristic 2.

To understand the following proofs and the corresponding figures it is advisable to recall that an ABP corresponds naturally to an iterated product of matrices if we number the vertices in each layer consecutively, starting with 1. Namely, consider two consecutive layers i and $i+1$ and let M_i be the matrix whose entry at position (v, w) is the label of the edge from vertex v in layer i to vertex w in layer $i+1$ (or 0 if there is no edge between these vertices). Then the ABP's value equals the product $M_k \cdots M_2 M_1$.

On a high level, our proof uses the following identities for addition:

$$\begin{pmatrix} 1 & 0 \\ g & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ f & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ f+g & 1 \end{pmatrix},$$

and negative squaring:

$$\begin{pmatrix} 1 & 0 \\ -\varepsilon^{-1}f & 1 \end{pmatrix}\begin{pmatrix} 1 & \varepsilon^2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ \varepsilon^{-1}f & 1 \end{pmatrix} = \begin{pmatrix} 1+\varepsilon f & \varepsilon^2 \\ -f^2 & 1-\varepsilon f \end{pmatrix} \xrightarrow{\varepsilon \to 0} \begin{pmatrix} 1 & 0 \\ -f^2 & 1 \end{pmatrix},$$

as well as rescaling:

$$\begin{pmatrix} \alpha^{-1} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ f & 1 \end{pmatrix}\begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \alpha f & 1 \end{pmatrix}.$$

Multiplication can now be simulated using the identity $fg = \frac{1}{2}((f+g)^2 - f^2 - g^2)$. This essentially proves Theorem 3.1, except that we need to ensure that the error terms cannot build up.

In the following, we aim for a more precise result about the continuant that requires a slightly more complicated construction. Hence, we note that each matrix on the left-hand sides of the addition and negative squaring identities is either upper or lower triangular with 1s on the main diagonal. Such matrices are always products of two *primitive Q-matrices*, defined as follows. For

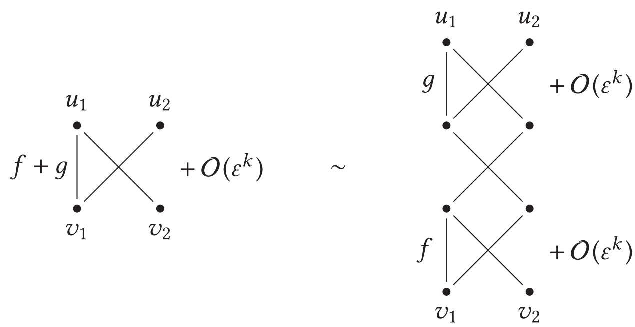


Fig. 1. Addition construction for Lemma 3.2.

a polynomial f over $\mathbb{F}(\varepsilon)$ define the matrix $Q(f) := \left(\begin{smallmatrix} f & 1 \\ 1 & 0 \end{smallmatrix}\right)$. A *parametrized affine linear form* is an affine linear form over the field $\mathbb{F}(\varepsilon)$. A *primitive Q-matrix* is any matrix $Q(\ell)$, where ℓ is a parametrized linear form.

For a 2×2 matrix M with entries in $\mathbb{F}(\varepsilon)[\mathbf{x}]$, we use the shorthand notation $M + \mathcal{O}(\varepsilon^k)$ for $M + \left(\begin{smallmatrix} \mathcal{O}(\varepsilon^k) & \mathcal{O}(\varepsilon^k) \\ \mathcal{O}(\varepsilon^k) & \mathcal{O}(\varepsilon^k) \end{smallmatrix}\right)$, where $\mathcal{O}(\varepsilon^k)$ denotes the set $\varepsilon^k\, \mathbb{F}[\varepsilon, \mathbf{x}]$. As a product of matrices, the ABP construction in our proof of Theorem 3.1 will be of the form $(1\ 0) M_\ell \cdots M_2 M_1 \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$, where the M_i are primitive Q-matrices $Q(f)$ for which f is either a constant from $\mathbb{F}(\varepsilon)$ or a variable. We are thus proving a slightly stronger statement than the statement of Theorem 3.1.

Lemma 3.2 (Addition). *Let $k \geq 1$. Let $f, g \in \mathbb{F}[\mathbf{x}]$ be polynomials such that some $F \in Q(f) + \mathcal{O}(\varepsilon^k)$ and some $G \in Q(g) + \mathcal{O}(\varepsilon^k)$ can be written as a product of n and m primitive Q-matrices, respectively. Then some matrix $H \in Q(f+g) + \mathcal{O}(\varepsilon^k)$ can be written as the product of $n+m+1$ primitive Q-matrices. Moreover, if the error degrees of F, G are e_f, e_g, respectively, then the error degree of H is at most $e_f + e_g$.*

Proof. Note that $(Q(f) + \mathcal{O}(\varepsilon^k)) \cdot Q(0) \cdot (Q(g) + \mathcal{O}(\varepsilon^k)) = Q(f+g) + \mathcal{O}(\varepsilon^k)$, so we have $H := F \cdot Q(0) \cdot G \in Q(f+g) + \mathcal{O}(\varepsilon^k)$. Moreover, the largest power of ε occurring in H is $\varepsilon^{e_f + e_g}$; see Figure 1. □

Lemma 3.3 (Squaring). *Let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial such that some $F \in Q(f) + \mathcal{O}(\varepsilon^3)$ can be written as the product of n primitive Q-matrices. Then some matrix $H \in Q(f^2) + \mathcal{O}(\varepsilon)$ and some matrix $H' \in Q(-f^2) + \mathcal{O}(\varepsilon)$ can be written as the product of $2n + 11$ primitive Q-matrices. Moreover, if the error degree of F is e_f then the error degree of H and H' is at most $2 \cdot e_f + 4$.*

Proof. We set

$$A := \begin{pmatrix} -\varepsilon^{-1} & 0 \\ 0 & \varepsilon \end{pmatrix} = Q(-\varepsilon^{-1}) \cdot Q(\varepsilon) \cdot Q(-\varepsilon^{-1}),$$

$$B := \begin{pmatrix} \varepsilon^2 & 1 \\ -1 & 0 \end{pmatrix} = Q(1) \cdot Q(-1) \cdot Q(1) \cdot Q(\varepsilon^2),$$

$$C := \begin{pmatrix} \varepsilon^{-1} & 0 \\ 0 & \varepsilon \end{pmatrix} = Q(-\varepsilon^{-1}) \cdot Q(\varepsilon - 1) \cdot Q(1) \cdot Q(\varepsilon^{-1} - 1).$$

Then one can check that

$$H \ := \ A \cdot F \cdot B \cdot F \cdot C \ \in \ A \cdot (Q(f) + \mathcal{O}(\varepsilon^3)) \cdot B \cdot (Q(f) + \mathcal{O}(\varepsilon^3)) \cdot C \ \in \ Q(-f^2) + \mathcal{O}(\varepsilon).$$

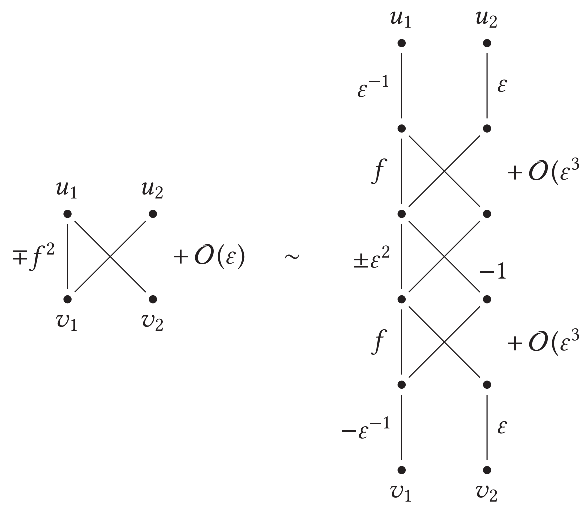


Fig. 2. Squaring construction for Lemma 3.3.

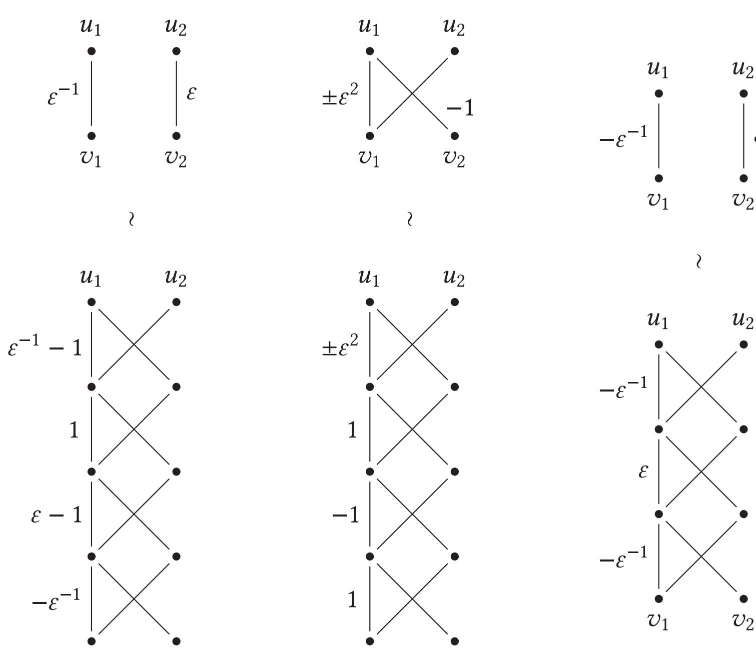


Fig. 3. Squaring construction subroutines for C, B, and A for Lemma 3.3.

To obtain $H' \in Q(f^2) + O(\varepsilon)$, we replace B by

$$B' := \begin{pmatrix} -\varepsilon^2 & 1 \\ -1 & 0 \end{pmatrix} = Q(1) \cdot Q(-1) \cdot Q(1) \cdot Q(-\varepsilon^2).$$

One checks that the highest power of ε appearing in H and H' is at most $2 \cdot e_f + 4$; see Figures 2 and 3 for a pictorial description. □

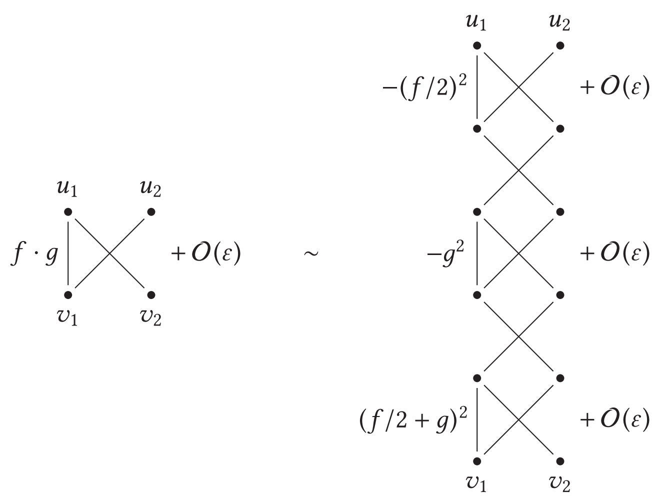


Fig. 4. Multiplication construction for Lemma 3.4.

Lemma 3.4 (Multiplication). *Let $f, g \in \mathbb{F}[\mathbf{x}]$ be polynomials such that some $F \in Q(f/2) + O(\varepsilon^3)$ and some $G \in Q(g) + O(\varepsilon^3)$ can be written as the product of n and m primitive Q-matrices, respectively. Then some $H \in Q(f \cdot g) + O(\varepsilon)$ can be written as the product of $4n + 4m + 37$ primitive Q-matrices. Moreover, if the error degrees of F, G are e_f, e_g, respectively, then the error degree of H is at most $4 \cdot e_f + 4 \cdot e_g + 12$.*

Proof. We make use of the identity $(-(f/2)^2) + (-g^2) + (f/2 + g)^2 = f \cdot g$. By the addition lemma (Lemma 3.2), $(f/2 + g) + O(\varepsilon^3)$ can be written as the product of $n + m + 1$ primitive Q-matrices with error degree at most $e_f + e_g$. By the squaring lemma (Lemma 3.3), $Q(-(f/2)^2) + O(\varepsilon)$, $Q(-g^2) + O(\varepsilon)$, and $Q((f/2 + g)^2) + O(\varepsilon)$ can be written as the product of $2n + 11$, $2m + 11$, and $2(n + m + 1) + 11$ primitive Q-matrices, respectively. The corresponding error degrees are at most $2 \cdot e_f + 4$, $2 \cdot e_g + 4$, and $2(e_f + e_g) + 4$. Finally, by the addition lemma again, $Q(f \cdot g) + O(\varepsilon) = Q(-(f/2)^2 + (-g^2) + (f/2 + g)^2) + O(\varepsilon)$ can be written as the product of $(2n + 11) + 1 + (2m + 11) + 1 + (2(n + m + 1) + 11) = 4n + 4m + 37$ primitive Q-matrices. The corresponding error degree is at most $(2 \cdot e_f + 4) + (2 \cdot e_g + 4) + (2(e_f + e_g) + 4) = 4 \cdot e_f + 4 \cdot e_g + 12$; see Figure 4 for a pictorial description. □

Proposition 3.5. *Let f be a polynomial computed by a formula of depth d. For every constant $\alpha \in \mathbb{F}$, some matrix in $F \in Q(\alpha f) + O(\varepsilon)$ can be written as a product of at most $45 \cdot 9^d$ primitive Q-matrices. Moreover, F has error degree at most $12 \cdot 25^d$.*

Proof. The proof is by induction on d. For $d = 0$, that is, f is a constant $\beta \in \mathbb{F}$ or a variable x, note that $Q(f)$ can be written directly as a primitive Q-matrix (with error degree 0). Since $Q(\alpha/2)$ can also be written directly (also with error degree 0), we can use the multiplication lemma (Lemma 3.4) to write $Q(\alpha f) + O(\varepsilon)$ as a product of $4 + 4 + 37 = 45$ primitive Q-matrices (with error degree at most 12).

For $d \geq 1$, fix a constant α. We know that either $f = g + h$ or $f = g \cdot h$ with formulas g, h of depth $< d$. By the induction hypothesis, for any constant β, γ, we can write $Q(\beta g) + O(\varepsilon)$ and $Q(\gamma h) + O(\varepsilon)$ as a product of $n_g, n_h \leq 45 \cdot 9^{d-1}$ primitive Q-matrices, with error degrees $e_g, e_h \leq 12 \cdot 25^{d-1}$.

Case $f = g + h$. We set $\beta = \gamma = \alpha$ and use the addition lemma (Lemma 3.2) to obtain $Q(\alpha f) + O(\varepsilon) = Q(\alpha g + \alpha h) + O(\varepsilon)$ as a product of $n_g + n_h + 1 \leq 2 \cdot 45 \cdot 9^{d-1} + 1 \leq 45 \cdot 9^d$ primitive Q-matrices, with error degree at most $e_g + e_h \leq 2 \cdot 12 \cdot 25^{d-1} \leq 12 \cdot 25^d$.

Case $f = g \cdot h$. By replacing ε by ε^3 in all primitive Q-matrices, we obtain matrices in $Q(\beta g) + O(\varepsilon^3)$ and $Q(\gamma h) + O(\varepsilon^3)$ as a product of n_g and n_h primitive Q-matrices with error degree at most $3 \cdot e_g$ and $3 \cdot e_h$, respectively. Now, we set $\beta = \alpha/2$ and $\gamma = 1$ and use the multiplication lemma (Lemma 3.4) to obtain $Q(\alpha f) + O(\varepsilon) = Q((\alpha \cdot g) \cdot h) + O(\varepsilon)$ as a product of $4n_g + 4n_h + 37 \leq 8 \cdot 45 \cdot 9^{d-1} + 37 \leq 45 \cdot 9^d$ primitive Q-matrices. The error degree is at most $4(3 \cdot e_g) + 4(3 \cdot e_h) + 12 = 12(e_g + e_h + 1) \leq 24 \cdot 12 \cdot 25^{d-1} + 12 \leq 12 \cdot 25^d$. □

The following proposition will be used to prove Theorem 3.1 (as a direct corollary) and to prove Theorem 3.12 on the continuant.

Proposition 3.6. *If $(f_n) \in \mathbf{VP}_\mathrm{e}$, then for each n a matrix in $F \in Q(f_n) + O(\varepsilon)$ can be written as a product of* poly(n) *many primitive Q-matrices. Moreover, F has error degree at most* poly(n).

Proof. The construction uses the classical depth-reduction theorem for formulas by Brent [7], for which a modern proof can be found in the survey of Saptharishi [37, Lemma 5.5]: If a family (f_n) has polynomially bounded formula size, then there are formulas computing f_n that have size poly(n) and depth $O(\log n)$. Applying Proposition 3.5 now yields the result. □

Proof of Theorem 3.1. This follows directly from Proposition 3.6. Namely, let $(f_n) \in \mathbf{VP}_e$. By Proposition 3.6, there is an $F \in Q(f_n) + O(\varepsilon)$ that is a product of polynomially many primitive Q-matrices such that F has polynomially bounded error degree. The width-2 ABP computing $f_n + O(\varepsilon)$ is given by $\begin{pmatrix} 1 & 0 \end{pmatrix} F \begin{pmatrix} 1 \\ 0 \end{pmatrix}$. □

Remark 3.7. The element $F \in Q(f_n) + O(\varepsilon)$ in the proof of Theorem 3.1, besides having polynomially bounded error degree, has the stronger property that the highest negative epsilon-power appearing with nonzero coefficient is polynomially bounded.

Example 3.8. Following the construction in Theorem 3.1, we get the following ABP for approximating the polynomial $x_1x_2 + x_3x_4 + \cdots + x_{15}x_{16}$, which cannot be computed by any width-2 ABP. Let

$$F(x,y) = \begin{pmatrix} \frac{1}{\varepsilon} - \frac{\varepsilon x}{2} & -\frac{x}{2\varepsilon} \\ \varepsilon^3 & \varepsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2}(x-2y)\varepsilon^2 + 1 & \frac{1}{2}(x-2y) \\ \varepsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x\varepsilon^2}{2} + 1 & -\frac{x}{2} \\ -\varepsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x+2y}{2\varepsilon} & \varepsilon \\ \varepsilon^{-1} & 0 \end{pmatrix}.$$

Then,

$$F(x,y) = \begin{pmatrix} xy & 1 \\ 1 & 0 \end{pmatrix} + O(\varepsilon).$$

Using the addition lemma Lemma 3.2, we get

$$\begin{pmatrix} 1 & 0 \end{pmatrix} F(x_1,x_2) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} F(x_3,x_4) \cdots \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} F(x_{15},x_{16}) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = x_1x_2 + x_3x_4 + \cdots + x_{15}x_{16} + O(\varepsilon).$$

Corollary 3.9. $\overline{\mathbf{VP}_2} = \overline{\mathbf{VP}_\mathrm{e}}$ *and* $\overline{\mathbf{VP}_2}$poly $= \overline{\mathbf{VP}_\mathrm{e}}^{\mathrm{poly}}$ *when* char$(\mathbb{F}) \neq 2$.

Proof. The inclusion $\mathbf{VP}_2 \subseteq \mathbf{VP}_\mathrm{e}$ is standard, see Equation (1). Taking closures on both sides, we obtain $\overline{\mathbf{VP}_2} \subseteq \overline{\mathbf{VP}_\mathrm{e}}$ and $\overline{\mathbf{VP}_2}^{\mathrm{poly}} \subseteq \overline{\mathbf{VP}_\mathrm{e}}^{\mathrm{poly}}$.

However, when char$(\mathbb{F}) \neq 2$, we have the inclusion $\mathbf{VP}_\mathrm{e} \subseteq \overline{\mathbf{VP}_2}^{\mathrm{poly}}$ (Theorem 3.1). By taking closures this implies $\overline{\mathbf{VP}_\mathrm{e}} \subseteq \overline{\mathbf{VP}_2}$ and $\overline{\mathbf{VP}_\mathrm{e}}^{\mathrm{poly}} \subseteq \overline{\mathbf{VP}_2}^{\mathrm{poly}}$. □

Corollary 3.10. $\overline{\mathbf{VP}_2}^{\mathrm{poly}} = \mathbf{VP}_\mathrm{e}$ *when* char$(\mathbb{F}) \neq 2$ *and* $\mathbb{F}$ *is infinite.*

Proof. By Corollary 3.9, we have $\overline{\mathbf{VP}_2}^{\text{poly}} = \overline{\mathbf{VP}_\text{e}}^{\text{poly}}$. The equality $\overline{\mathbf{VP}_\text{e}}^{\text{poly}} = \mathbf{VP}_\text{e}$ follows from a standard interpolation argument; see, e.g., Reference [8, Proposition 8.1] for details. (The interpolation argument in the context of approximative complexity goes back to Bini [5], see also Strassen [39], Bürgisser et al. [11], or Reference [18, Lemma 17] and the references therein.) □

As a consequence of Proposition 3.5, we obtain a new description of $\overline{\mathbf{VP}_\text{e}}$ as follows. The *continuant* $F_n(x_1, \ldots, x_n)$ is defined via $F_0 := 1$, $F_1 := x_1$, and $F_n := x_n F_{n-1} + F_{n-2}$ for all $n \geq 2$. Among the well-known algebraic properties of F_n is the fact that $F_n(1, 1, \ldots, 1)$ is the nth Fibonacci number and that F_n is the upper left entry of a product of Q-matrices $Q(x_i)$, that is, $F_n(x_1, \ldots, x_n) = (Q(x_n)Q(x_{n-1}) \cdots Q(x_1))_{1,1}$.

Definition 3.11. A polynomial f is a *projection* of F_m if there exist affine linear forms $\ell_1, \ldots, \ell_m$ such that $f = F_m(\ell_1, \ldots, \ell_m)$. The smallest m such that f is a projection of F_m we call the *continuant complexity of* f. A polynomial f is a *degeneration of* F_m if there exist parametrized affine linear forms $\ell_1(\varepsilon), \ldots, \ell_m(\varepsilon)$ such that $F_m(\ell_1(\varepsilon), \ldots, \ell_m(\varepsilon)) \in f + O(\varepsilon)$. The smallest m such that f is a degeneration of F_m we call the *border continuant complexity* of f, and denote it by $\underline{L}_{\text{Con}}(f)$. A family (h_n) of polynomials is called $\overline{\mathbf{VP}_\text{e}}$*-complete under p-degenerations* if $(h_n) \in \overline{\mathbf{VP}_\text{e}}$ and for every $(f_n) \in \overline{\mathbf{VP}_\text{e}}$ there exists a polynomially bounded function t such that f_n is a degeneration of $h_{t(n)}$.

The continuant complexity is not always finite [1], but Proposition 3.6 shows that the border continuant complexity $\underline{L}_{\text{Con}}(f)$ is always finite and that $\overline{\mathbf{VP}_\text{e}}$ can be characterized as the class of families with polynomially bounded border continuant complexity:

Theorem 3.12. $\overline{\mathbf{VP}_\text{e}} = \{(f_n) \mid \underline{L}_{\text{Con}}(f_n) \in \text{poly}(n)\}$.

Proof. Clearly the right-hand side is contained in the left-hand side. $\mathbf{VP}_\text{e}$ is contained in the right-hand side by Proposition 3.6. A moment's thought reveals that the right-hand side is closed under the approximation closure in the sense of Definition 2.1. Thus, taking the closure on both sides yields the result. □

Remark 3.13. Theorem 3.12 shows that (F_n) is $\overline{\mathbf{VP}_\text{e}}$-complete under p-degenerations. From the proof of Proposition 3.5 it follows that also (F_{2n+1}) is $\overline{\mathbf{VP}_\text{e}}$-complete under p-degenerations, that is, we only need the F_m with odd index m (this follows from $\det(Q(f)) = -1$).

Remark 3.14 (Symmetry). Define the polynomial $C_n(x_1, \ldots, x_n)$ as

$$C_n(x_1, \ldots, x_n) := \text{trace}(Q(x_n) \cdot Q(x_{n-1}) \cdots Q(x_1)).$$

Since the trace of a matrix product is invariant under cyclic shifts of the matrices, the polynomial $C_n(x_1, \ldots, x_n)$ is invariant under cyclic shifts of the variables $x_1, \ldots, x_n$. Thus, C_n can be viewed as a cyclically symmetric version of F_n. (Note that C_n and F_n are also both invariant under reversing the order of the variables $x_1, \ldots, x_n$, that is, mapping $(x_1, \ldots, x_n)$ to $(x_n, \ldots, x_1)$.) Define the *border cyclic continuant complexity* analogously to the border continuant complexity by replacing F_n by C_n in Definition 3.11. Analogously to Theorem 3.12, we now see that the families (C_n) and (C_{2n+1}) are both $\overline{\mathbf{VP}_\text{e}}$-complete under p-degenerations. The polynomial C_n is called *rotundus* in Reference [15].

Remark 3.15 (A closed form for F_n and C_n). We describe another way to write F_n and C_n. An *adjacent pair* is a set of two numbers $\{i, i+1\}$ with $1 \leq i < n$. A *supporting set* is the set $\{1, 2, \ldots, n\}$ after removing a disjoint (possibly empty) union of adjacent pairs. For a supporting set S define $x_S := \prod_{i \in S} x_i$. Then $F_n(x_1, \ldots, x_n) = \sum_S x_S$, where the sum is over all supporting sets.

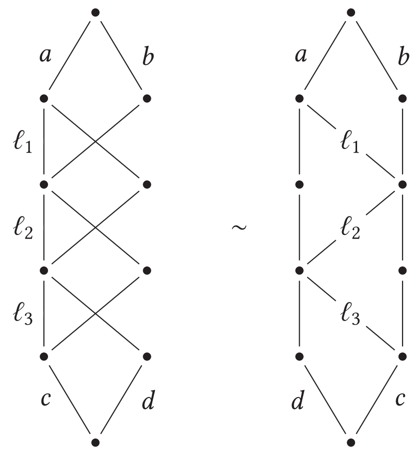


Fig. 5. Making an ABP consisting of three primitive Q-matrices planar.

We define a *cyclicly adjacent pair* as a set that is either an adjacent pair or the set $\{1, n\}$, if $1 \neq n$. We define a *cyclic supporting set* as the set $\{1, 2, \ldots, n\}$ after removing a disjoint (possibly empty) union of cyclicly adjacent pairs. Then $C_n(x_1, \ldots, x_n) = \sum_S x_S$, where the sum is over all cyclic supporting sets.

Remark 3.16 (Planarity). We remark that the product of two Q-matrices $Q(x)Q(y)$ can be rewritten as $Q(x)Q(y) = (Q(x)\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right))(\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)Q(y))$. We also have $Q(x)\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) = (Q(x)\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right))\left(\begin{smallmatrix} b \\ a \end{smallmatrix}\right)$. Consider a width-2 ABP that is a product of primitive Q-matrices,

$$(a \quad b)Q(\ell_1)Q(\ell_2)\cdots Q(\ell_k)\begin{pmatrix} c \\ d \end{pmatrix}.$$

By pairing up the ith Q-matrix with the $(i+1)$th Q-matrix for each odd i, and using the above equations, we can rewrite this ABP into a width-2 ABP whose underlying graph has no crossing edges, that is, a *planar* with-2 ABP; see Figure 5 for an example with three Q-matrices. Planarity has been studied in the context of ABPs over $\{0, 1\}$ in, e.g., Reference [27].

4 VNP VIA PRODUCTS OF AFFINE LINEAR FORMS

Valiant proved the following characterization of **VNP** in his seminal work [41]; see also References [11, Theorem 21.26], [9, Theorem 2.13], and [28, Therorem 2].

THEOREM 4.1 (VALIANT [41]). $\mathbf{VNP}_\mathrm{e} = \mathbf{VNP}$.

We strengthen Valiant's characterization of **VNP** from $\mathbf{VNP}_\mathrm{e}$ to $\mathbf{VNP}_1$.

THEOREM 4.2. $\mathbf{VNP}_1 = \mathbf{VNP}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

We give two proofs. The idea of the first proof is to show that the **VNP**-complete permanent family $\mathrm{per}_n := \sum_{\sigma \in S_n} \prod_{i \in [n]} x_{i,\sigma(i)}$ is in $\mathbf{VNP}_1$. The idea of the second proof is to simulate in $\mathbf{VNP}_1$ the primitives that are used in the proof of $\mathbf{VP}_\mathrm{e} = \mathbf{VP}_3$ by Reference [3]. We present the second proof in Appendix B. The advantage of the second proof is that we can restrict the ABP edge labels to affine linear forms that have at most two variables; see Theorem B.3. Both proofs use the following lemma to write expressions of the form $1 + xy$ as a hypercube sum of a product of affine linear forms.

Lemma 4.3. $\frac{1}{2}\sum_{b\in\{0,1\}}(x+1-2b)(y+1-2b) = 1+xy$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof. Expanding the left side gives the right side. □

Proof of Theorem 4.2. The permanent family (per_n) is well-known to be **VNP**-complete under p-projections, see, for example, Reference [9, Theorem 2.10]. Therefore, to show that $\mathbf{VNP} \subseteq \mathbf{VNP}_1$, it suffices to show that $(\mathrm{per}_n) \in \mathbf{VNP}_1$. We begin by writing per_n as an inclusion-exclusion-type expression due to Ryser [35, Theorem 4.1],

$$\mathrm{per}_n = (-1)^n \sum_{S\subseteq[n]} (-1)^{|S|} \prod_{j\in[n]} \sum_{i\in S} x_{i,j}.$$

Encoding every subset $S \subseteq [n]$ by a bit string $b = (b[1],\dots,b[n]) \in \{0,1\}^n$, we can rewrite the above as

$$\begin{aligned}\mathrm{per}_n &= (-1)^n \sum_{b\in\{0,1\}^n} \left(\prod_{k\in[n]} (1-2b[k])\right) \prod_{j\in[n]} \sum_{i\in[n]} b[i]\, x_{i,j} \\ &= (-1)^n \sum_{b\in\{0,1\}^n} \left(\prod_{k\in[n]} (1-2b[k])\right) \sum_{i_1,\dots,i_n\in[n]} \prod_{j\in[n]} b[i_j]\, x_{i_j,j}.\end{aligned}$$

For notational convenience, we use square brackets not only to refer to sets ($[n] := \{1,\dots,n\}$) but also to entries in a list ($b[k] := b_k$). We now introduce new Boolean variables $a[i,j]$, $1 \le i \le n-1$, $1 \le j \le n$, and we fix the values $a[0,j] = 1$, $a[n,j] = 0$. (This gives an $(n+1)\times n$ matrix of variables and constants in which the first row consists of all 1s and the last row contains only 0s.) We claim that the above expression equals

$$\begin{aligned}\mathrm{per}_n = (-1)^n \sum_{b\in\{0,1\}^n} \Bigg(\prod_{k\in[n]} (1-2b[k]) \cdot \sum_a \prod_{i,j\in[n]} &(1+(x_{i,j}-1)(a[i-1,j]-a[i,j])) \\ \cdot \Big(1+(b[i]-1)(a[i-1,j]-a[i,j])\Big) \cdot \Big(1+(a[i-1,j]-1)a[i,j]\Big)\Bigg),&\end{aligned} \tag{4}$$

where the second sum is over all Boolean assignments of $a[i,j]$. The idea is to encode the indices $i_1,\dots,i_n$ in the boolean variables $a[i,j]$ in unary. For example, for $n=4$, if $i_1 = 4$, $i_2 = 3$, $i_3 = 1$, $i_4 = 4$, then the corresponding matrix a is

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

We prove the claim Equation (4) in three steps. Fix j.

— If $a[i-1,j] = 0$ and $a[i,j] = 1$, then $1+(a[i-1,j]-1)a[i,j] = 0$. Thus, if in the sequence $a[0,j],\dots,a[n,j]$ a 0 is followed by a 1, then $\prod_{i\in[n]}(1+(a[i-1,j]-1)a[i,j] = 0$. Conversely, if $(a[0,j],\dots,a[n,j]) = (1,\dots,1,0,\dots,0)$, then $\prod_{i\in[n]}(1+(a[i-1,j]-1)a[i,j]) = 1$. The assignments of $(a[0,j],\dots,a[n,j])$ that contribute to the sum are thus exactly of the form $(1,\dots,1,0,\dots,0)$ where the first 0 occurs at some index $1 \le z \le n$ (since we have set $a[0,j] = 1$ and $a[n,j] = 0$). Fix such an assignment with first 0 occurring at index z.

— If $i = z$, then $1 + (x_{i,j} - 1)(a[i-1,j] - a[i,j])$ equals $x_{i,j}$. If $i \neq z$, then it equals 1.
— If $i = z$, then $1 + (b[i] - 1)(a[i-1,j] - a[i,j])$ equals $b[i]$. If $i \neq z$, then it equals 1.

This proves Equation (4).

Next, we apply Lemma 4.3, introducing fresh hypercube variables $c_1[i,j]$, $c_2[i,j]$, and $c_3[i,j]$, for $1 \leq i,j \leq n$, to obtain

$$\begin{aligned}\mathrm{per}_n = (-1)^n \left(\tfrac{1}{2}\right)^{3n^2} \sum_b \left(\prod_{k \in [n]} (1 - 2\,b[k]) \right) \cdot \sum_a \Bigg(\prod_{i,j \in [n]} & \\ \sum_{c_1[i,j]} [(x_{i,j} - 2\,c_1[i,j]) \cdot (a[i-1,j] - a[i,j] + 1 - 2\,c_1[i,j])] & \\ \cdot \sum_{c_2[i,j]} [(b[i] - 2\,c_2[i,j]) \cdot (a[i-1,j] - a[i,j] + 1 - 2\,c_2[i,j])] & \\ \cdot \sum_{c_3[i,j]} [(a[i-1,j] - 2\,c_3[i,j]) \cdot (a[i,j] + 1 - 2\,c_3[i,j])] \Bigg), &\end{aligned}$$

where the sum goes over all Boolean assignments of $b[i]$, $a[i,j]$, $c_1[i,j]$, $c_2[i,j]$, $c_3[i,j]$, for all indices $1 \leq i,j \leq n$, except for $a[n,j] := 0$, and $a[0,j] := 1$. After a rearrangement, we obtain the expression

$$\begin{aligned}\mathrm{per}_n = \sum_{\substack{a,\,b \\ c_1,\,c_2,\,c_3}} \Bigg((-1)^n \left(\tfrac{1}{2}\right)^{3n^2} \left(\prod_{k \in [n]} (1 - 2\,b[k]) \right) \cdot \prod_{i,j \in [n]} & \\ (x_{i,j} - 2\,c_1[i,j]) \cdot (a[i-1,j] - a[i,j] + 1 - 2\,c_1[i,j]) & \\ \cdot (b[i] - 2\,c_2[i,j]) \cdot (a[i-1,j] - a[i,j] + 1 - 2\,c_2[i,j]) & \\ \cdot (a[i-1,j] - 2\,c_3[i,j]) \cdot (a[i,j] + 1 - 2\,c_3[i,j]) \Bigg), &\end{aligned}$$

where the sum goes over all Boolean assignments of $a[i,j]$, $b[i]$, $c_1[i,j]$, $c_2[i,j]$, $c_3[i,j]$ for all indices $1 \leq i,j \leq n$, again except for $a[n,j] := 0$, and $a[0,j] := 1$. This shows that $(\mathrm{per}_n) \in \mathbf{VNP}_1$. □

In Section 5, we will prove that the statement of Theorem 4.2 does not hold over $\mathbb{F}_2$, that is, $\mathbf{VNP}_1 \subsetneq \mathbf{VNP}$ when $\mathbb{F} = \mathbb{F}_2$. We leave the situation over other fields of characteristic 2 as an open problem.

5 $\mathbf{VNP}_1 \subsetneq \mathbf{VNP}$ WHEN $\mathbb{F} = \mathbb{F}_2$

In our proofs of $\mathbf{VNP}_1 = \mathbf{VNP}$ (Section 4 and Appendix B) the assumption $\mathrm{char}(\mathbb{F}) \neq 2$ played a crucial role. We can prove that over the finite field $\mathbb{F}_2$ the inclusion $\mathbf{VNP}_1 \subseteq \mathbf{VNP}$ is indeed strict.

Proposition 5.1. $\mathbf{VNP}_1 \subsetneq \mathbf{VNP}$ *when* $\mathbb{F} = \mathbb{F}_2$.

Proof. Let $\mathbb{F} = \mathbb{F}_2$. Clearly $(1 + xy) \in \mathbf{VNP}$. However, we will prove that $1 + xy$ cannot be written as a hypercube sum of affine linear forms. In fact, we will prove something stronger, namely that the *function* $(x,y) \mapsto 1 + xy$ cannot be written as a hypercube sum of a product of affine linear forms.

Assume the contrary: the function $(x, y) \mapsto 1 + xy$ can be written as a hypercube sum of a product of affine linear forms. We can thus write

$$1 + xy = \textstyle\sum_{\mathbf{b}} L_{\mathbf{b}} \quad \text{with} \quad L_{\mathbf{b}} := \prod_{i=1}^{\alpha}(x + A_i) \prod_{j=1}^{\beta}(y + B_j) \prod_{k=1}^{\gamma}(x + y + C_k) \tag{5}$$

for some affine linear forms $A_i(\mathbf{b})$, $B_j(\mathbf{b})$, $C_k(\mathbf{b})$ in the hypercube variables $\mathbf{b}$. On $\mathbb{F}_2$ the functions $x, x^2, x^3, \ldots$ coincide; the functions $y, y^2, y^3, \ldots$ coincide; and the functions $x + y, (x + y)^2$, $(x + y)^3, \ldots$ coincide, so

$$\textstyle\prod_i (x + A_i) = \prod_i A_i + x\Big(\prod_i (1 + A_i) + \prod_i A_i\Big),$$

$$\textstyle\prod_j (y + B_j) = \prod_j B_j + y\Big(\prod_j (1 + B_j) + \prod_j B_j\Big),$$

$$\textstyle\prod_k (x + y + C_k) = \prod_k C_k + (x + y)\Big(\prod_k (1 + C_k) + \prod_k C_k\Big).$$

We multiply the three expressions. To simplify the notation, we write $\mathcal{A} := \prod_i A_i$, $\overline{\mathcal{A}} := \prod_i (1 + A_i)$, $\mathcal{B} := \prod_j B_j$, $\overline{\mathcal{B}} := \prod_j (1 + B_j)$, $C := \prod_k C_k$, $\overline{C} := \prod_k (1 + C_k)$. In this notation, we have $L_{\mathbf{b}} = (\mathcal{A} + x(\mathcal{A} + \overline{\mathcal{A}}))(\mathcal{B} + y(\mathcal{B} + \overline{\mathcal{B}}))(C + (x + y)(C + \overline{C}))$. We expand

$$\begin{aligned} L_{\mathbf{b}} = {} & \mathcal{A}\mathcal{B}C \\ & + x((\mathcal{A} + \overline{\mathcal{A}})\mathcal{B}C + \mathcal{A}\mathcal{B}(C + \overline{C})) \quad + \quad x^2((\mathcal{A} + \overline{\mathcal{A}})\mathcal{B}(C + \overline{C})) \\ & + y(\mathcal{A}(\mathcal{B} + \overline{\mathcal{B}})C + \mathcal{A}\mathcal{B}(C + \overline{C})) \quad + \quad y^2(\mathcal{A}(\mathcal{B} + \overline{\mathcal{B}})(C + \overline{C})) \\ & + xy(\mathcal{A}(\mathcal{B} + \overline{\mathcal{B}})(C + \overline{C}) + (\mathcal{A} + \overline{\mathcal{A}})\mathcal{B}(C + \overline{C}) + (\mathcal{A} + \overline{\mathcal{A}})(\mathcal{B} + \overline{\mathcal{B}})C) \\ & + (x^2y + xy^2)(\mathcal{A} + \overline{\mathcal{A}})(\mathcal{B} + \overline{\mathcal{B}})(C + \overline{C}). \end{aligned}$$

Simplifying powers of x and y and using that the characteristic is 2, we obtain

$$L_{\mathbf{b}} = \mathcal{A}\mathcal{B}C + x(\mathcal{A}\mathcal{B}C + \overline{\mathcal{A}}\mathcal{B}\overline{C}) + y(\mathcal{A}\mathcal{B}C + \mathcal{A}\overline{\mathcal{B}}\overline{C}) + xy(\mathcal{A}\mathcal{B}C + \overline{\mathcal{A}}\overline{\mathcal{B}}C + \overline{\mathcal{A}}\mathcal{B}\overline{C} + \mathcal{A}\overline{\mathcal{B}}\overline{C}).$$

Plugging in the four possible assignments $(x, y) \in \mathbb{F}_2 \times \mathbb{F}_2$ into $1 + xy = \sum_{\mathbf{b}} L_{\mathbf{b}}$, we get the following system of equations:

$$\textstyle\sum_{\mathbf{b}} \prod_{i,j,k} A_i B_j C_k = 1, \tag{6}$$

$$\textstyle\sum_{\mathbf{b}} \prod_{i,j,k} (1 + A_i) B_j (1 + C_k) = 1, \tag{7}$$

$$\textstyle\sum_{\mathbf{b}} \prod_{i,j,k} A_i (1 + B_j)(1 + C_k) = 1, \tag{8}$$

$$\textstyle\sum_{\mathbf{b}} \prod_{i,j,k} (1 + A_i)(1 + B_j) C_k = 0. \tag{9}$$

We will show that the above system of equations is inconsistent. Note that Equation (6) asserts that an odd number of vectors $\mathbf{b}$ satisfy the system of equations

$$\begin{aligned} A_i &= 1\ \forall i, \\ B_j &= 1\ \forall j, \\ C_k &= 1\ \forall k. \end{aligned}$$

Recall that we defined α, β, γ as the number of factors $x + A_i$, $y + B_j$, $x + y + C_k$ in Equation (5), respectively. Let $m := \alpha + \beta + \gamma$. Recall that we defined n as the number of hypercube variables b_ℓ. As we work over $\mathbb{F}_2$, any affine linear form in $\mathbf{b}$ can be written as $\alpha_0 + \sum_{\ell=1}^{n} \alpha_\ell b_\ell$ with $\alpha_i \in \{0, 1\}$. Write the ith linear form in $(A_1, \ldots, A_\alpha, B_1, \ldots, B_\beta, C_1, \ldots, C_\gamma)$ as $v_{0,i} + \sum_{\ell=1}^{n} b_\ell v_{\ell,i}$, and let $v_\ell = (v_{\ell,1}, \ldots, v_{\ell,m})$ for $0 \leq \ell \leq n$. We define the linear map $M : \mathbb{F}_2^n \to \mathbb{F}_2^m$ by $M(\mathbf{b}) = \sum_{\ell=1}^{n} b_\ell v_\ell$.

We call a bit vector $\mathbf{b} \in \mathbb{F}_2^n$ a *solution* of Equation (6) if $M(b) = v_0 + 1^\alpha 1^\beta 1^\gamma$, where $1^\alpha 1^\beta 1^\gamma$ is the all-ones vector of length $m = \alpha + \beta + \gamma$. Observe that Equation (6) says that there is an odd number of solutions of Equation (6). Since the set of solutions of Equation (6) forms an affine linear subspace of $(\mathbb{F}_2)^n$, its cardinality is a power of two. The only odd power of two is 1, so there is exactly one solution of Equation (6). Let $b^{(1)}$ be this unique solution: $M(b^{(1)}) = v_0 + 1^\alpha 1^\beta 1^\gamma$. We do the same for Equations (7) and (8) and find unique solutions $M(b^{(2)}) = v_0 + 0^\alpha 1^\beta 0^\gamma$ and $M(b^{(3)}) = v_0 + 1^\alpha 0^\beta 0^\gamma$. Equation (9) asserts that the number of solutions of Equation (9) is even. One solution of Equation (9) is given by $M(b^{(1)} + b^{(2)} + b^{(3)}) = 3v_0 + 1^\alpha 1^\beta 1^\gamma + 0^\alpha 1^\beta 0^\gamma + 1^\alpha 0^\beta 0^\gamma = v_0 + 0^\alpha 0^\beta 1^\gamma$. Let $b^{(4')}$ and $b^{(4'')}$ be two distinct solutions of Equation (9) with $M(b^{(4')}) = M(b^{(4'')}) = v_0 + 0^\alpha 0^\beta 1^\gamma$. Then $M(b^{(2)} + b^{(3)} + b^{(4')}) = v_0 + 1^\alpha 1^\beta 1^\gamma = M(b^{(2)} + b^{(3)} + b^{(4'')})$, which contradicts the uniqueness of $b^{(1)}$. □

Remark 5.2. In the proof of Proposition 5.1, we considered a family (f_n) consisting of a single polynomial $f_n = 1 + xy$. We can immediately generalize this and find many more families in $\mathbf{VNP} \setminus \mathbf{VNP}_1$ over $\mathbb{F}_2$: For any family $g_n = g_n(x_1, \ldots, x_n)$ in $\mathbf{VNP}$, the family (h_n) with

$$h_n := g_n(x_1, \ldots, x_n) - g_n(0, \ldots, 0) + 1 + x_{n+1}x_{n+2}$$

is clearly in $\mathbf{VNP}$. However, if (h_n) would be in $\mathbf{VNP}_1$, then after setting $x_1 = \cdots = x_n = 0$, we would obtain a representation of $h_n(0, \ldots, 0, x, y) = 1 + xy = f_n$ as a hypercube sum of a product of affine linear forms, contradicting Proposition 5.1. Thus, (h_n) is in $\mathbf{VNP} \setminus \mathbf{VNP}_1$ over $\mathbb{F}_2$.

Remark 5.3. Our proof of Proposition 5.1 does not generalize to all fields $\mathbb{F}$ of characteristic 2, because the polynomial $1 + xy$ is in fact computable by a hypercube sum of a product of affine linear forms when $\mathbb{F} = \mathbb{F}_4$ (and thus when $\mathbb{F} = \mathbb{F}_{2^{2k}}$, $k \in \mathbb{N}$). Indeed, $\mathbb{F}_4 \cong \mathbb{F}_2[Z]/(Z^2 + Z + 1)$, so the element $Z \in \mathbb{F}_4$ is a third root of unity ($Z^3 = 1$) and satisfies $Z^2 + Z + 1 = 0$. It can be checked that, therefore, $\sum_{b=0}^{1}(x + Z^2 y + Zb) \cdot (x + Zy + Z^2 b) \cdot (x + y + b)$ equals $1 + xy$.

APPENDIXES

A ABPS WITH RESTRICTED EDGE LABELS

So far the edge labels of our ABPs were allowed to be arbitrary affine linear forms. This section is about ABPs in which the edge labels are restricted to be simple affine linear forms (“weak ABPs”), or variables and constants (“weakest ABPs”). These edge label types were also studied in Reference [1].

Definition A.1. A wst-ABP (weakest ABP) is an ABP with edges labeled by variables or constants. A w-ABP (weak ABP) is an ABP with edges labeled by simple affine linear forms $\alpha x_i + \beta$, $\alpha, \beta \in \mathbb{F}$. A g-ABP (general ABP) is an ABP with edges labeled by general affine linear forms $\sum_i \alpha_i x_i + \beta$, $\alpha_i, \beta \in \mathbb{F}$. For τ equal to wst, w, or g, the class $\mathbf{VP}_k^\tau$ consists of all families of polynomials over polynomially many variables that are computed by polynomial-size width-k τ-ABPs. In the rest of this article, τ will act as a variable from {wst, w, g}. By $\mathbf{VP}_k$, we mean $\mathbf{VP}_k^{\mathrm{g}}$.

From the above definition, it follows that $\mathbf{VP}_k^{\mathrm{wst}} \subseteq \mathbf{VP}_k^{\mathrm{w}} \subseteq \mathbf{VP}_k^{\mathrm{g}}$.

Remark A.2. One checks that the construction in the proof of Theorem 3.1 actually proves the inclusion $\mathbf{VP}_\mathrm{e} \subseteq \overline{\mathbf{VP}_2^{\mathrm{wst}}}^{\mathrm{poly}}$ when $\mathrm{char}(\mathbb{F}) \neq 2$. The inclusion $\mathbf{VP}_\mathrm{e} \subseteq \overline{\mathbf{VP}_2^{\mathrm{wst}}}^{\mathrm{poly}}$ implies the equalities $\overline{\mathbf{VP}_2^{\mathrm{wst}}} = \overline{\mathbf{VP}_\mathrm{e}}$ and $\overline{\mathbf{VP}_2^{\mathrm{wst}}}^{\mathrm{poly}} = \overline{\mathbf{VP}_\mathrm{e}}^{\mathrm{poly}}$.

In the following sections, we will prove all inclusions and separations that are listed in Figure 8.

A.1 Comparing Different Types of Edge Labels in Width-2 ABPs

The aim of this subsection is to prove the following separation.

THEOREM A.3. $\mathbf{VP}_2^{\mathrm{w}} \subsetneq \mathbf{VP}_2^{\mathrm{g}}$.

In fact, we will show the following stronger statement.

THEOREM A.4. *The polynomial*

$$p(\mathbf{x}) = (x_{11} + x_{12} + \cdots + x_{17})(x_{21} + x_{22} + \cdots + x_{27}) + (x_{31} + x_{32} + \cdots + x_{37})(x_{41} + x_{42} + \cdots + x_{47})$$

is computable by a width-2 g-*ABP, but not computable by any width-2* w-*ABP.*

We leave it as an open problem whether the inclusion $\mathbf{VP}_2^{\mathrm{wst}} \subseteq \mathbf{VP}_2^{\mathrm{w}}$ is strict.

To prove Theorem A.4, we will review and reuse the arguments used by Allender and Wang [1] to show that the polynomial $x_1x_2 + \cdots + x_{15}x_{16}$ cannot be computed by any width-2 g-ABP.

For the proof of Theorem A.4, we may without loss of generality assume that the base field $\mathbb{F}$ is algebraically closed, because for any field $\mathbb{F}$, if p is not computable over the algebraic closure of $\mathbb{F}$, then it is not computable over $\mathbb{F}$ itself. Let $\mathbb{H}$ be the affine linear forms that are single variables x_i or constants $\mathbb{F}$. Let $\mathbb{S}$ be the set of simple affine linear forms. Let $\mathbb{L}$ be the set of general affine linear forms. Let $\mathbb{H}^{2\times 2}$, $\mathbb{S}^{2\times 2}$, $\mathbb{L}^{2\times 2}$ be the sets of 2×2 matrices with entries in $\mathbb{H}$, $\mathbb{S}$, $\mathbb{L}$, respectively. In this subsection, all ABPs have width 2, and by a wst-, w-, or g-ABP Γ, we will mean a sequence $\Gamma_k, \ldots, \Gamma_1$ with $\Gamma_k \in \mathbb{F}^{1\times 2}$, $\Gamma_{k-1}, \ldots, \Gamma_2 \in X^{2\times 2}$, and $\Gamma_1 \in \mathbb{F}^{2\times 1}$ with X equal to $\mathbb{H}$, $\mathbb{S}$ or $\mathbb{L}$, respectively. We call $\Gamma_{k-1}, \ldots, \Gamma_2$ the *inner matrices of* Γ. It is important for technical reasons that Γ_1 and Γ_k have field entries only.

Definition A.5. A matrix $A \in \mathbb{L}^{2\times 2}$ is called *inherently nondegenerate* (indg) when $\det(A) \in \mathbb{F} \setminus \{0\}$. A matrix $A \in \mathbb{L}^{2\times 2}$ that is not inherently nondegenerate is called *possibly degenerate* (pdg).

Allender and Wang prove the following necessary condition for a polynomial to be computable by a wst-, w-, or g-ABP whose inner matrices are indg. Let $\mathrm{H}(p)$ denote the highest-degree homogeneous part of a polynomial p.

THEOREM A.6 ([1, THEOREM 3.9 AND LEMMA 4.7]). *Let p be a polynomial and Γ a* wst-, w- *or* g-*ABP computing p, whose inner matrices are indg. Then, $\mathrm{H}(p)$ is a product of homogeneous linear forms.*

Our next goal is to give a necessary condition for a polynomial p to be computable by a w-ABP. We begin with a simple lemma, which can essentially be found in Reference [1], but we include its brief proof here for completeness.

LEMMA A.7 ([1]). *Let p be a polynomial. If p is computed by a* w-*ABP that has an inner matrix containing four distinct variables, then there is an assignment π of these four variables with $\pi(p) = 0$.*

PROOF. Let M be such a matrix. Since the ABP is of type w, M is of the form

$$M = \begin{pmatrix} \alpha_{11}x_{11} + \beta_{11} & \alpha_{12}x_{12} + \beta_{12} \\ \alpha_{21}x_{21} + \beta_{21} & \alpha_{22}x_{22} + \beta_{22} \end{pmatrix}$$

for some constants $\alpha_{ij} \in \mathbb{F} \setminus \{0\}$, $\beta_{ij} \in \mathbb{F}$. Applying the four assignments $x_{ij} \mapsto -\beta_{ij}/\alpha_{ij}$ makes M zero and thus p zero. □

We need two more ideas before we will state and prove the necessary condition we are after. (1) Let $A \in \mathbb{L}^{2\times 2}$ be pdg. Then, there is an assignment π of the variables such that $\pi(A)$ has only constant entries and has rank ≤ 1. (2) Let p be a polynomial computed by an ABP Γ, that is,

$p = \Gamma_k \cdots \Gamma_1$. Suppose that Γ contains an inner matrix Γ_i, $1 < i < k$, with only constant entries and with rank ≤ 1. Then, there is a constant 2×1 matrix $\Gamma_{i,2}$ and a constant 1×2 matrix $\Gamma_{i,1}$ such that $\Gamma_i = \Gamma_{i,2}\Gamma_{i,1}$. Then, p is a product

$$p = p_2 p_1$$

of polynomials p_1, p_2, each computable by an ABP, namely

$$p_2 = \Gamma_k \cdots \Gamma_{i+1}\Gamma_{i,2},$$
$$p_1 = \Gamma_{i,1}\Gamma_{i-1} \cdots \Gamma_1.$$

We say that p *factors into* $p_2 p_1$. Recall that $\mathrm{H}(p)$ denotes the highest-degree homogeneous part of a polynomial p. The following is implicit in Reference [1].

Theorem A.8 ([1]). *Let p be a polynomial computed by a* w*-ABP* Γ. *Then, there is an assignment π of at most six variables such that $\mathrm{H}(\pi(p))$ is either a constant, a homogeneous linear form, or a product of two homogeneous polynomials of positive degree.*

Proof. Let $(\Gamma_k, \ldots, \Gamma_1)$ be the matrices of Γ, so that $p = \Gamma_k \cdots \Gamma_1$. Clearly, if some Γ_i is the zero matrix, then $p = 0$, and we are also done. If there is a Γ_i containing four distinct variables, then there is an assignment π of these four variables with $\pi(p) = 0$ (Lemma A.7), so we are done. Otherwise, all Γ_i are nonzero and have at most three distinct variables. If the inner Γ_i are all indg, then $\mathrm{H}(p)$ is a product of homogeneous linear forms (Theorem A.6), in which case we are done. Therefore, we are left to discuss the case where there is at least one nonzero pdg inner matrix. Consider the nonempty subsequence $\mathcal{M} = (M_\ell, \ldots, M_1)$ of all nonzero pdg inner matrices. For each M_i there is an assignment π of at most three distinct variables such that $\pi(M_i)$ has only constant entries and rank ≤ 1. To each M_i, we assign a type (several types might be possible for a single M_i, in which case, we choose and fix the type arbitrarily from the possible ones):

— If there is an assignment π of at most three variables of M_i such that $\pi(M_i)$ is constant of rank $\leq$1 and $\pi(p)$ factors into a product $p_2 p_1$ with p_2 and p_1 both constant, then M_i has type "C".
— If M_i does not have type "C" and if there is an assignment π of at most three variables of M_i such that $\pi(M_i)$ is constant of rank ≤ 1 and $\pi(p)$ factors into a product $p_2 p_1$ with p_2 and p_1 both polynomials of positive degree, then M_i has type "P."
— If M_i does not have type "C" or "P" and if there is an assignment π of at most three variables of M_i such that $\pi(M_i)$ is constant of rank ≤ 1 and $\pi(p)$ factors into a product $p_2 p_1$ with p_2 a polynomial of positive degree and p_1 constant, then M_i has type "L."
— If M_i does not have type "C" or "P" or "L" and if there is an assignment π of at most three variables of M_i such that $\pi(M_i)$ is constant of rank ≤ 1 and $\pi(p)$ factors into a product $p_2 p_1$ with p_2 constant and p_1 a polynomial of positive degree, then M_i has type "R."

The slight imbalance between type "L" and "R" will be relevant. In particular, in a type "R" matrix M every assignment of variables for which M becomes rank deficient results in a factorization of p with a constant left factor. We consider four possible situations.

(a) *There is an $M \in \mathcal{M}$ of type "C" or "P."* In this case, we are done.
(b) *M_1 has type "R."* Then p_1 is computed by an ABP whose inner matrices are all indg (since M_1 is the right-most pdg inner matrix) and hence $\mathrm{H}(p_1)$ is a product of homogeneous linear forms (Theorem A.6), so we are done.
(c) *M_ℓ has type "L."* Then p_2 is computed by an ABP whose inner matrices are all indg (since M_ℓ is the left-most pdg inner matrix) and hence $\mathrm{H}(p_2)$ is a product of homogeneous linear forms (Theorem A.6), so we are done.

(d) *Remaining situation.* Since we are not in situation (a), the types "C" and "P" do not appear. Since we are neither in situation (b) nor (c), both types "L" and "R" do appear. Let i be the largest number such that M_i has type "L." Since we are not in situation (c), $M_{i+1}, \dots, M_\ell$ all have type "R." With a assignment π to at most three variables of M_i, $\pi(p)$ factorizes as $p_2 p_1$. Consider the matrices $\pi(M_j)$, $i + 1 \leq j \leq \ell$. If those are all indg, then $\mathrm{H}(p_2)$ is a product of homogeneous linear forms (Theorem A.6), and so is $\mathrm{H}(\pi(p)) = \mathrm{H}(p_2)p_1$, which means that we are done. Otherwise, choose the smallest j, $i + 1 \leq j \leq \ell$, such that $\pi(M_j)$ is pdg. Since M_j has type "R", there is an assignment σ of at most three variables of $\pi(M_j)$, such that $\sigma(p_2)$ factors into $p_4 p_3$ with p_4 constant. Since p_3 is computed by an ABP whose inner matrices are all indg, $\mathrm{H}(p_3)$ is a product of homogeneous linear forms (Theorem A.6). Since $\sigma(\pi(p)) = p_4 p_3 \sigma(p_1)$ is a scalar multiple of p_3, the theorem follows. □

Theorem A.4 (repeated). *The polynomial*

$$p(\mathbf{x}) = (x_{11} + x_{12} + \cdots + x_{17})(x_{21} + x_{22} + \cdots + x_{27}) + (x_{31} + x_{32} + \cdots + x_{37})(x_{41} + x_{42} + \cdots + x_{47})$$

is computable by a width-2 g*-ABP but not computable by any width-2* w*-ABP.*

Proof. Clearly $p(\mathbf{x})$ is computable by a width-2 g-ABP. Suppose $p(\mathbf{x})$ is computable by a width-2 w-ABP. Then, by Theorem A.8 there is an assignment π of at most six variables such that either $\pi(p)$ is affine linear or $\mathrm{H}(\pi(p))$ is a product of two polynomials of positive degree. The first option is impossible, because distinct variables do not cancel. So, $\mathrm{H}(\pi(p))$ is a product of two polynomials of positive degree. With another assignment σ, we can achieve that $\mathrm{H}(\sigma(\pi(p))$ is of the form $x_i x_j + x_k x_\ell$ for some distinct variables x_i, x_j, x_k, x_ℓ. This is not a product of two polynomials of positive degree, so $\mathrm{H}(\pi(p))$ is not either. □

A.2 Comparing Different Types of Edge Labels in Width-1 ABPs

Clearly, $\mathbf{VP}_1^{\mathrm{wst}} \subseteq \mathbf{VP}_1^{\mathrm{w}} \subseteq \mathbf{VP}_1^{\mathrm{g}}$ and $\mathbf{VP}_1^{\tau} \subseteq \mathbf{VP}_2^{\tau}$ ($\tau \in \{\mathrm{wst}, \mathrm{w}, \mathrm{g}\}$), but this does not give a complete description of all inclusions among these classes. The following two propositions realize a complete description among $\mathbf{VP}_1^{\tau}$ and $\mathbf{VP}_2^{\mathrm{wst}}$.

Proposition A.10. $\mathbf{VP}_1^{\mathrm{g}} \subseteq \mathbf{VP}_2^{\mathrm{wst}}$.

Proof. Let $(p_n) \in \mathbf{VP}_1^{\mathrm{g}}$. Then, each p_n is a product of poly(n) affine linear forms in poly(n) variables. Let $\ell(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_m x_m$ be such an affine linear form with $\alpha_0 \in \mathbb{F}$ and $\alpha_1, \dots, \alpha_m \in \mathbb{F} \setminus \{0\}$. We can compute $\ell(\mathbf{x})$ with the width-2 wst-ABP in Figure 6. A product of affine linear forms can be computed by the width-2 wst-ABP that is the concatenation of the width-2 wst-ABPs computing the affine linear forms. For p_n the resulting ABP has poly(n) size. Thus, $(p_n) \in \mathbf{VP}_2^{\mathrm{wst}}$. □

Proposition A.11. $\mathbf{VP}_1^{\mathrm{wst}} \subsetneq \mathbf{VP}_1^{\mathrm{w}} \subsetneq \mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VP}_2^{\mathrm{wst}}$.

Proof. If $(p_n) \in \mathbf{VP}_1^{\mathrm{wst}}$, then p_n is a monomial. However, $(\alpha_0 + \alpha_1 x_1) \in \mathbf{VP}_1^{\mathrm{w}}$ and $\alpha_0 + \alpha_1 x_1$ is not a monomial, so $\mathbf{VP}_1^{\mathrm{wst}} \subsetneq \mathbf{VP}_1^{\mathrm{w}}$. If $(p_n) \in \mathbf{VP}_1^{\mathrm{w}}$ and p_n is homogeneous, then p_n is a monomial. However, $((x_1 + x_2)^2) \in \mathbf{VP}_1^{\mathrm{g}}$ and $(x_1 + x_2)^2$ is not a monomial, so $\mathbf{VP}_1^{\mathrm{w}} \subsetneq \mathbf{VP}_1^{\mathrm{g}}$. The last inclusion is Proposition A.10. To see the strictness, if $(p_n) \in \mathbf{VP}_1^{\mathrm{g}}$, then the highest-degree homogeneous part $\mathrm{H}(p_n)$ of p_n is a product of homogeneous linear forms. However, $(x_1 x_2 + x_3 x_4) \in \mathbf{VP}_2^{\mathrm{wst}}$ and $x_1 x_2 + x_3 x_4$ is not a product of homogeneous linear forms, so $\mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VP}_2^{\mathrm{wst}}$. □

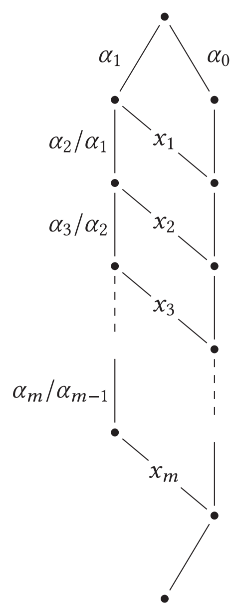


Fig. 6. Width-2 wst-ABP computing $\ell(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_m x_m$.

A.3 Approximation in Width-1 ABPs

The following proposition says that each of $\mathbf{VP}_1^{\mathrm{wst}}$, $\mathbf{VP}_1^{\mathrm{w}}$, and $\mathbf{VP}_1^{\mathrm{g}}$ is closed under approximation.

Proposition A.12. $\mathbf{VP}_1^\tau = \overline{\mathbf{VP}_1^\tau}$ *for* $\tau \in \{\mathrm{wst}, \mathrm{w}, \mathrm{g}\}$.

Proof. Trivially, $\mathbf{VP}_1^\tau \subseteq \overline{\mathbf{VP}_1^\tau}$. To prove the opposite inclusion, let $(f_n) \in \overline{\mathbf{VP}_1^\tau}$. There are polynomials $g_n(\varepsilon, \mathbf{x}) \in \mathbb{F}[\varepsilon, \mathbf{x}]$ such that $f_n + \varepsilon g_n(\varepsilon, \mathbf{x})$ can be written as a product of poly(n) affine linear forms in $\mathbb{F}(\varepsilon)[\mathbf{x}]$ in poly(n) variables (these affine linear forms have either wst-, w-, or g-type). That is (forgetting the subscript n for the moment), $f(\mathbf{x}) + \varepsilon g(\varepsilon, \mathbf{x})$ can be written as

$$f(\mathbf{x}) + \varepsilon g(\varepsilon, \mathbf{x}) = \prod_{i=1}^{m} \ell_i(\varepsilon, \mathbf{x})$$

with

$$\ell_i(\varepsilon, \mathbf{x}) = \sum_{j=d_i}^{e_i} \varepsilon^j k_{i,j}(\mathbf{x})$$

for some affine linear forms $k_{i,j} \in \mathbb{F}[\mathbf{x}]$, such that $k_{i,d_i}(\mathbf{x}) \neq 0$, and $d_i \leq e_i \in \mathbb{Z}$. By shifting ε-factors from $\ell_1, \ldots, \ell_{m-1}$ to ℓ_m, we can assume that $d_i = 0$ for $i < m$. We claim that $d_m \geq 0$. If $d_m < 0$, then expanding $\prod_i \ell_i(\mathbf{x})$ as a Laurent series in ε gives a term with a negative power of ε. This contradicts $f(\mathbf{x}) + \varepsilon g(\mathbf{x})$ having only nonnegative powers of ε. Therefore, the $\ell_i(\mathbf{x})$ do not contain any negative powers of ε, and we can safely substitute $\varepsilon \mapsto 0$ in each linear form ℓ_i to obtain f as a product of affine linear forms in $\mathbb{F}[\mathbf{x}]$ (either of wst-, w-, or g-type). Remembering our subscript n again, we have thus proven $(f_n) \in \mathbf{VP}_1^\tau$. □

A.4 Nondeterminism in Width-1 ABPs

In the following proposition, we compare $\mathbf{VP}_1^\tau$ to $\mathbf{VNP}_1^\tau$ for all three versions $\tau \in \{\mathrm{wst}, \mathrm{w}, \mathrm{g}\}$.

Proposition A.13.

—$\mathbf{VP}_1^\tau = \mathbf{VNP}_1^\tau$ *for* τ *equal to* wst *or* w.
—$\mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VNP}_1^{\mathrm{g}}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof. Trivially, $\mathbf{VP}_1^\tau \subseteq \mathbf{VNP}_1^\tau$. Let $(p_n) \in \mathbf{VNP}_1^{\mathrm{wst}}$. Then, p_n can be written as a hypercube-sum over a monomial,

$$p(\mathbf{x}) = \sum_{\mathbf{b}\in\{0,1\}^{\mathrm{poly}(n)}} m(\mathbf{b},\mathbf{x}),$$

with m a monomial (subscripts n are implied). For any $\mathbf{b}$-variable that does not occur in m, we remove that $\mathbf{b}$-variable form the summation and at the same time multiply the expression by 2, to again have an expression for $p(\mathbf{x})$. Assuming all $\mathbf{b}$-variables occur in m, only for $\mathbf{b} = (1,1,\ldots,1)$ can $m(\mathbf{b},\mathbf{x})$ be nonzero. So, $p(\mathbf{x}) = m((1,\ldots,1),\mathbf{x})$. Remembering the subscript n, we proved $(p_n) \in \mathbf{VP}_1^{\mathrm{wst}}$.

Let $(p_n) \in \mathbf{VNP}_1^{\mathrm{w}}$. Then (forgetting the subscript n),

$$p(\mathbf{x}) = \sum_{\mathbf{b}\in\{0,1\}^{\mathrm{poly}(n)}} \prod_i \ell_i(\mathbf{b}) \prod_j k_j(\mathbf{x})$$

for some simple affine linear forms ℓ_i in the variables $\mathbf{b}$ and some simple affine linear forms k_j in the variables $\mathbf{x}$. The product $\prod_j k_j(\mathbf{x})$ is independent of $\mathbf{b}$, while $\sum_{\mathbf{b}} \prod_i \ell_i(\mathbf{b})$ is a constant. We can thus write $p(\mathbf{x})$ as a constant times $\prod_j k_j(\mathbf{x})$. Therefore (remembering n), $p_n(\mathbf{x}) \in \mathbf{VP}_1^{\mathrm{w}}$. This proves the first line of the proposition.

To prove the second line, recall that if $(p_n) \in \mathbf{VP}_1^{\mathrm{g}}$, then p_n is a product of affine linear forms. However, let $p_n(x_1,x_2) = \sum_{b\in\{0,1\}}(x_1+b)(x_2+b) = 2x_1x_2 + x_1 + x_2 + 1$. Then, $(p_n) \in \mathbf{VNP}_1^{\mathrm{g}}$, but $p_n(x_1,x_2)$ is a not a product of affine linear forms, as we will now verify. Suppose $2x_1x_2 + x_1 + x_2 + 1 = (\alpha_0 + \alpha_1x_1 + \alpha_2x_2)(\beta_0 + \beta_1x_1 + \beta_2x_3)$. Then, $\alpha_1\beta_1 = 0$ and $\alpha_2\beta_2 = 0$. Since $\alpha_1\beta_1 = 0$, we may assume without loss of generality that $\alpha_1 = 0$. Since not both α_1 and α_2 can be 0 (otherwise $(\alpha_0 + \alpha_1x_1 + \alpha_2x_2)(\beta_0 + \beta_1x_1 + \beta_2x_2)$ has degree 1) and since $\alpha_2\beta_2 = 0$, we have $\beta_2 = 0$. Hence, $2x_1x_2 + x_1 + x_2 + 1 = (\alpha_0 + \alpha_2x_2)(\beta_0 + \beta_1x_1)$. Then, $\alpha_0\beta_0 = 1$, $\alpha_0\beta_1 = 1$, $\alpha_2\beta_0 = 1$, and $\alpha_2\beta_1 = 2$. The first two of these equations imply $\beta_0 = \beta_1$, which contradicts the last two of these equations. So $\mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VNP}_1^{\mathrm{g}}$. □

Remark A.14. It follows directly from Propositions A.13 and A.11 that we have strict inclusions $\mathbf{VNP}_1^{\mathrm{wst}} \subsetneq \mathbf{VNP}_1^{\mathrm{w}} \subsetneq \mathbf{VNP}_1^{\mathrm{g}}$, when $\mathrm{char}(\mathbb{F}) \neq 2$.

Remark A.15. For showing $\mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VNP}_1^{\mathrm{g}}$ we considered a family (p_n) consisting of a single polynomial $p_n = 2x_1x_2 + x_1 + x_2 + 1$. Similar to Remark 5.2, it is easy to find many more families $(f_n) \in \mathbf{VNP}_1^{\mathrm{g}} \setminus \mathbf{VP}_1^{\mathrm{g}}$. For example, for $f_n = f_n(x_1,\ldots,x_n) = p_n(x_1,x_2)\cdot(1+x_3)(1+x_4)\cdots(1+x_n)$, we clearly obtain $(f_n) \in \mathbf{VNP}_1^{\mathrm{g}}$, but since $p_n(x_1,x_2) = f_n(x_1,x_2,0,\ldots,0)$, we cannot have $(f_n) \in \mathbf{VP}_1^{\mathrm{g}}$.

B ALTERNATIVE PROOF OF $\mathbf{VNP}_1$ = VNP VIA $\mathbf{VP}_3$

Recall that in Section 4, we proved that

$$\mathbf{VNP}_1^{\mathrm{g}} = \mathbf{VNP}, \tag{10}$$

using the completeness of the permanent (Theorem 4.2). We will present an alternative proof of Equation (10) inspired by the proof of the following theorem by Ben-Or and Cleve. The alternative proof of Equation (10) has the benefit that it can be extended to show a slightly stronger result, see Theorem B.3.

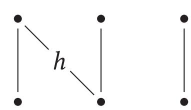
Theorem B.1 (Ben-Or and Cleve [3]). *For* $k \geq 3$, $\mathbf{VP}_k^\tau = \mathbf{VP}_\mathrm{e}$ *for* $\tau \in \{\mathrm{wst}, \mathrm{w}, \mathrm{g}\}$.

We include a proof of this theorem, since we will later adapt it to prove Equation (10).

Proof. It is well-known (see Equation (1)) that $\mathbf{VP}_k^\tau \subseteq \mathbf{VP}_\mathrm{e}$. We will prove that $\mathbf{VP}_\mathrm{e} \subseteq \mathbf{VP}_3^{\mathrm{wst}}$, from which it follows that $\mathbf{VP}_\mathrm{e} \subseteq \mathbf{VP}_k^\tau$ and thus $\mathbf{VP}_k^\tau = \mathbf{VP}_\mathrm{e}$. For a polynomial h, define the matrix

$$M(h) := \begin{pmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which, as part of an ABP, looks like



We call the following matrices *primitive*:

- $M(h)$ with h any variable or any constant in $\mathbb{F}$,
- every 3×3 permutation matrix M_π with $\pi \in S_3$ any permutation,
- every diagonal matrix $M_{a,b,c} := \mathrm{diag}(a, b, c)$ with a, b, c any constants in $\mathbb{F}$.

The entries of the primitives are variables or constants in $\mathbb{F}$, making them suitable to use in the construction of a width-3 wst-ABP (Definition A.1).

Let $(f_n) \in \mathbf{VP}_\mathrm{e}$. Then f_n can be computed by a formula of size $s(n) \in \mathrm{poly}(n)$. By Brent's depth-reduction theorem for formulas [7] f_n can then also be computed by a formula of size $\mathrm{poly}(n)$ and depth $d(n) \in O(\log n)$.

We will construct a sequence of primitive matrices $A_1, \ldots, A_{m(n)}$, such that

$$A_1 \cdots A_{m(n)} = \begin{pmatrix} 1 & 0 & 0 \\ f_n & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

with $m(n) \in O(4^{d(n)}) = \mathrm{poly}(n)$. Then,

$$f_n(\mathbf{x}) = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} M_{-1,1,0} A_1 \cdots A_m \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

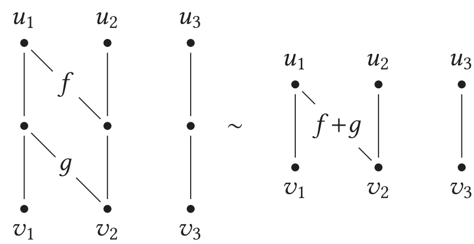
so $f_n(\mathbf{x})$ can be computed by a width-3 wst-ABP of size $\mathrm{poly}(n)$, proving the theorem.

To explain the construction, let h be a polynomial and consider a formula computing h of depth d. The goal is to construct (recursively on the formula structure) primitive matrices $A_1, \ldots, A_m$, such that

$$A_1 \cdots A_m = \begin{pmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{with } m \in O(4^d). \tag{11}$$

Suppose h is a variable or a constant. Then, $M(h)$ is itself a primitive matrix.

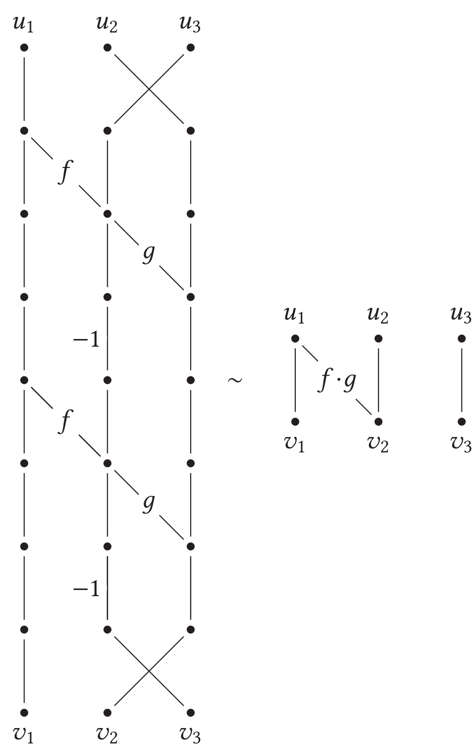
Suppose $h = f + g$ is a sum of two polynomials f, g and suppose $M(f)$ and $M(g)$ can be written as a product of primitive matrices. Then, $M(f + g)$ equals a product of primitive matrices, because $M(f + g) = M(f)M(g)$. This can easily be verified directly, or by noting that in the corresponding partial ABPs, the top-bottom paths (u_i-v_j paths) have the same value:



Suppose $h = fg$ is a product of two polynomials f, g and suppose $M(f)$ and $M(g)$ can be written as a product of primitive matrices. Then, $M(fg)$ equals a product of primitive matrices, because

$$M(f \cdot g) = M_{(23)} \left(M_{1,-1,1} M_{(123)} M(g) M_{(132)} M(f) \right)^2 M_{(23)}$$

(here, $(23) \in S_3$ denotes the transposition $1 \mapsto 1, 2 \mapsto 3, 3 \mapsto 2$ and $(123) \in S_3$ denotes the cyclic shift $1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 1$), as can be verified either directly or by checking that in the corresponding partial ABPs, the top-bottom paths (u_i-v_j paths) have the same value:



This completes the construction.

The length m of the construction is $m(h) = 1$ for h a variable or constant and recursively $m(f + g) = m(f) + m(g)$, $m(f \cdot g) = 2(m(f) + m(g)) + O(1)$, so $m \in O(4^d)$, where d is the formula depth of h. The construction thus satisfies Equation (11), proving the theorem. □

We will now give an alternative proof of Theorem 4.2.

Theorem 4.2 (repeated). **VNP**$_1$ = **VNP** *when* char($\mathbb{F}$) $\neq 2$.

Second proof of Theorem 4.2. Clearly, $\mathbf{VNP}_1^{\mathrm{g}} \subseteq \mathbf{VNP}$ by Equation (1) and taking the nondeterminism closure N. We will prove that $\mathbf{VNP} \subseteq \mathbf{VNP}_1^{\mathrm{g}}$.

Recall that in the proof of $\mathbf{VP}_{\mathrm{e}} \subseteq \mathbf{VP}_3^{\mathrm{wst}}$ (Theorem B.1), we defined for any polynomial h the matrix

$$M(h) := \begin{pmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and we called the following matrices *primitive*:

- $M(h)$ with h any variable or any constant in $\mathbb{F}$,
- every 3×3 permutation matrix M_π with $\pi \in S_3$ any permutation,
- every diagonal matrix $M_{a,b,c} := \mathrm{diag}(a, b, c)$ with a, b, c any constants.

In the proof of $\mathbf{VP}_{\mathrm{e}} \subseteq \mathbf{VP}_3^{\mathrm{wst}}$, we constructed, for any family $(f_n) \in \mathbf{VP}_{\mathrm{e}}$, a sequence of primitive matrices $A_{n,1}, \ldots, A_{n,t(n)}$ with $t(n) \in \mathrm{poly}(n)$, such that

$$f_n(\mathbf{x}) = (1 \quad 1 \quad 1) M_{-1,1,0} A_1 \cdots A_m \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

We will construct a hypercube sum over a width-1 g-ABP that evaluates the right-hand side, to show that $\mathbf{VP}_{\mathrm{e}} \subseteq \mathbf{VNP}_1^{\mathrm{g}}$. This implies $\mathbf{VNP}_{\mathrm{e}} \subseteq \mathbf{VNP}_1^{\mathrm{g}}$ by taking closures. Then by Valiant's Theorem 4.1, $\mathbf{VNP} \subseteq \mathbf{VNP}_1^{\mathrm{g}}$.

Let $f(\mathbf{x})$ be a polynomial and let $A_1, \ldots, A_k$ be primitive matrices, such that $f(\mathbf{x})$ is computed as

$$f(\mathbf{x}) = (1 \quad 1 \quad 1) A_k \cdots A_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

View this expression as a width-3 ABP G, with vertex layers labeled as shown in the left diagram of Figure 7. Assume for simplicity that all edges between layers are present, possibly with label 0. The sum of the values of every s-t path in G equals $f(\mathbf{x})$,

$$f(\mathbf{x}) = \sum_{j \in [3]^k} A_k[j_k, j_{k-1}] \cdots A_1[j_2, j_1]. \tag{12}$$

We now introduce some hypercube variables. To every vertex, except s and t, we associate a bit; the bits in the ith layer we call $b_1[i], b_2[i], b_3[i]$. To an s-t path in G, we associate an assignment of the $b_j[i]$ by setting the bits of vertices visited by the path to 1 and the others to 0. For example, in the right diagram in Figure 7 we show an s-t path with the corresponding assignment of the bits $b_1[i]$, $b_2[i]$, and $b_3[i]$. The assignments of $b_j[i]$ corresponding to s-t paths are the ones such that for every $i \in [k]$ exactly one of $b_1[i], b_2[i], b_3[i]$ equals 1. Let

$$V(b_1, b_2, b_3) := \prod_{i \in [k]} (b_1[i] + b_2[i] + b_3[i]) \prod_{\substack{s,\, t \,\in [3]: \\ s \neq t}} \big(1 - b_s[i] b_t[i]\big). \tag{13}$$

The assignments of $b_j[i]$ corresponding to s-t paths are thus the ones such that $V(b_1, b_2, b_3) = 1$. Otherwise, $V(b_1, b_2, b_3) = 0$.

We will now write $f(\mathbf{x})$ as a hypercube sum by replacing each $A_i[j_i, j_{i-1}]$ in Equation (12) by a product of affine linear forms $S_i(A_i)$ with variables $\mathbf{b}$ and $\mathbf{x}$ as follows:

$$\sum_{\mathbf{b}} V(b_1, b_2, b_3) S_k(A_k) \cdots S_1(A_1).$$

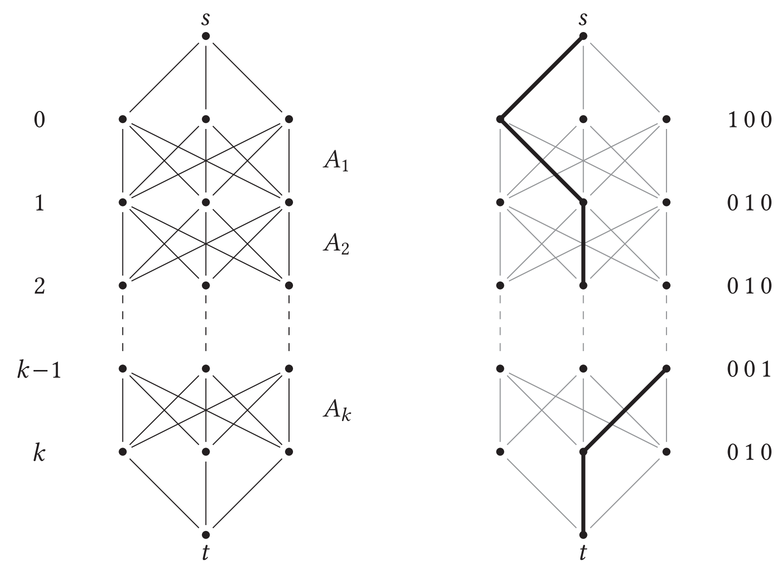


Fig. 7. Illustration of layer labelling and path labelling in the proof of Theorem 4.2.

Define $\mathrm{Eq}(\alpha, \beta) : \{0, 1\}^2 \to \{0, 1\}$ by $(1 - \alpha - \beta)(1 - \alpha - \beta)$. This function is 1 if $\alpha = \beta$ and 0 otherwise.

—For any variable or constant x, define

$$\begin{aligned} \mathrm{S}_i(M(x)) := &\Big(1 + (x - 1)(b_1[i] - b_1[i{-}1])\Big) \\ &\cdot \Big(1 - (1 - b_2[i])b_2[i{-}1]\Big) \\ &\cdot \mathrm{Eq}\Big(b_3[i{-}1], b_3[i]\Big). \end{aligned}$$

—For any permutation $\pi \in S_3$, define

$$\begin{aligned} \mathrm{S}_i(M_\pi) := &\,\mathrm{Eq}\Big(b_1[i{-}1], b_{\pi(1)}[i]\Big) \\ &\cdot \mathrm{Eq}\Big(b_2[i{-}1], b_{\pi(2)}[i]\Big) \\ &\cdot \mathrm{Eq}\Big(b_3[i{-}1], b_{\pi(3)}[i]\Big). \end{aligned}$$

—For any constants $a, b, c \in \mathbb{F}$, define

$$\begin{aligned} \mathrm{S}_i(M_{a,b,c}) := &\Big(a \cdot b_1[i{-}1] + b \cdot b_2[i{-}1] + c \cdot b_3[i - 1]\Big) \\ &\cdot \mathrm{Eq}\Big(b_1[i{-}1], b_1[i]\Big) \\ &\cdot \mathrm{Eq}\Big(b_2[i{-}1], b_2[i]\Big) \\ &\cdot \mathrm{Eq}\Big(b_3[i{-}1], b_3[i]\Big). \end{aligned}$$

One verifies that with these definitions, indeed,

$$f(\mathbf{x}) = \sum_{\mathbf{b}} V(b_1, b_2, b_3) S_k(A_k) \cdots S_1(A_1).$$

Some of the factors in the $S_i(A_i)$ are not affine linear. As a final step, we apply the equation $1 + xy = \frac{1}{2}\sum_{c\in\{0,1\}}(x + 1 - 2c)(y + 1 - 2c)$ (Lemma 4.3) to write these factors as products of affine linear forms, introducing new hypercube variables. □

Combining Theorem 4.2 and Remark A.14 gives the separation $\mathbf{VNP}_1^{\mathrm{w}} \subsetneq \mathbf{VNP}_1^{\mathrm{g}} = \mathbf{VNP}$. We can prove a slightly stronger separation by adjusting the construction in the above proof of Theorem 4.2. Namely, let $\mathbb{S}^+ := \{\alpha x_i + \beta x_j + \gamma \mid \alpha, \beta, \gamma \in \mathbb{F}\}$ be the set of affine linear forms in at most two variables and let $\mathbf{VP}_1^{\mathrm{w}+}$ be the class of families that can be computed by width-1 ABPs over $\mathbb{S}^+$ of polynomial size. Define $\mathbf{VNP}_1^{\mathrm{w}+}$ accordingly (Definition 2.3). Then, we can adjust the construction in the above proof of Theorem 4.2 to show the following.

Theorem B.3. $\mathbf{VNP}_1^{\mathrm{w}} \subsetneq \mathbf{VNP}_1^{\mathrm{w}+} = \mathbf{VNP}$ *when* $\mathrm{char}(\mathbb{F}) \neq 2$.

Proof. We only need to show $\mathbf{VNP}_1^{\mathrm{w}+} = \mathbf{VNP}$, as $\mathbf{VNP}_1^{\mathrm{w}} \subsetneq \mathbf{VNP}$ was shown in Remark A.14. The adjustments we have to make to the construction in the proof of Theorem 4.2 are as follows. Most of the resulting polynomial of the construction is already of the correct form where each linear forms contains at most two variables, since the expression $\mathrm{Eq}(x, y) = (1 - x - y)^2$ and the expression $1 + xy = \frac{1}{2}\sum_{c\in\{0,1\}}(x + 1 - 2c)(y + 1 - 2c)$ are of this form. Three expressions occur that are not of the correct form:

(1) $b_1[i] + b_2[i] + b_3[i]$ in $V(b_1, b_2, b_3)$,
(2) $a \cdot b_1[i-1] + b \cdot b_2[i-1] + c \cdot b_3[i-1]$ in $\mathrm{S}(M_{a,b,c})$, and
(3) $1 + (x - 1)(b_1[i] - b_1[i-1])$ in $\mathrm{S}(M(x))$.

Expressions 1 and 2 we can write in the correct form using the identity

$$\frac{1}{2}\sum_{b\in\{0,1\}}(x + 1 - 2b)(y + 1 - 2b)(z + 1 - 2b) = x + y + z + xyz. \tag{14}$$

Indeed, expression 1 can be replaced by

$$\begin{aligned}\frac{1}{2}\sum_{c\in\{0,1\}}&(b_1[i] + 1 - 2c)(b_2[i] + 1 - 2c)(b_3[i] + 1 - 2c)\\ &= b_1[i] + b_2[i] + b_3[i] + b_1[i]b_2[i]b_3[i],\end{aligned}$$

since the unwanted term $b_1[i]b_2[i]b_3[i]$ will always vanish in our construction (because in Equation (13), we multiply with $1 - b_s[i]b_t[i]$ for every $s \neq t$). Similar for expression 2.

For expression 3, we first replace the expression $1 + (x - 1)(b_1[i] - b_1[i-1])$ by the expression $\frac{1}{2}\sum_{c\in\{0,1\}}(x - 1 + 1 - 2c)(b_1[i] - b_1[i-1] + 1 - 2c)$. The second factor has too many variables. We replace it, using identity Equation (14), by

$$\begin{aligned}\frac{1}{2}\sum_{c'\in\{0,1\}}&\Big(b_1[i] + 1 - 2c'\Big)\Big(-b_1[i-1] + 1 + 1 - 2c'\Big)\Big(-2c + 1 - 2c'\Big)\\ &= b_1[i] - b_1[i-1] + 1 - 2c + b_1[i](1 - b_1[i-1])(-2c).\end{aligned}$$

The first four summands in the right-hand side are as we want. The last summand is only nonzero if $b_1[i] = 1$ and $b_1[i-1] = 0$. However, since $\mathrm{S}_i(M(x))$ contains a factor $1 - (1 - b_2[i])b_2[i-1]$ and a factor $\mathrm{Eq}(b_3[i-1], b_3[i])$, it can be checked that this last summand will always vanish.

In the new construction thus obtained each linear form is in $\mathbb{S}^+$. This completes the necessary adjustments to the construction. □

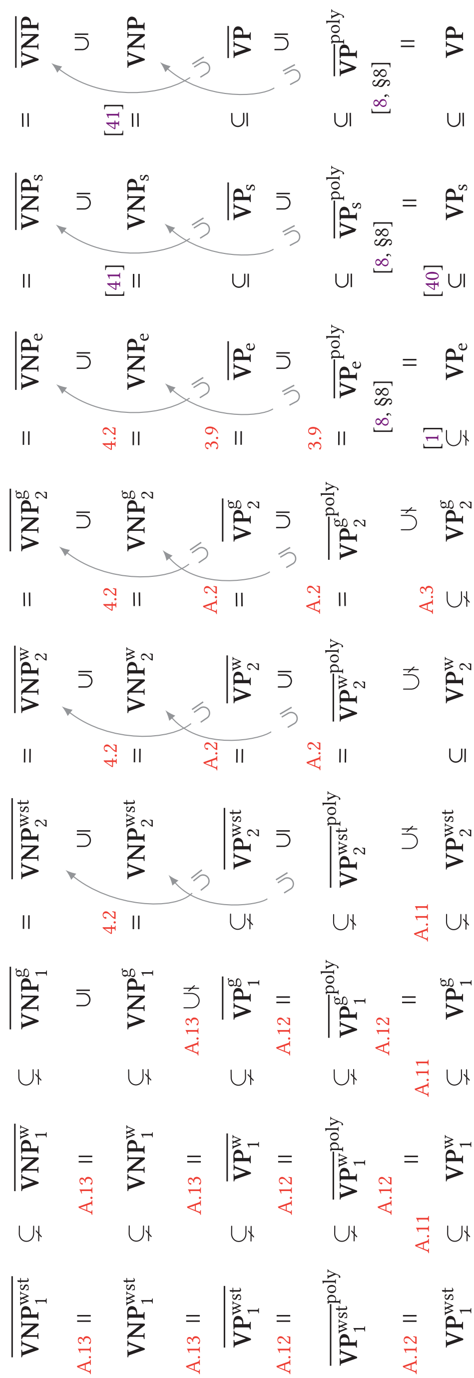


Fig. 8. Overview of inclusions and separations among $\mathbf{VP}_k^{\tau}$ ($\tau \in \{wst, w, g\}$), $\mathbf{VP}_e$, $\mathbf{VP}_s$, $\mathbf{VP}$ and their closures when $\text{char}(\mathbb{F}) \neq 2$.

REFERENCES

[1] Eric Allender and Fengming Wang. 2016. On the power of algebraic branching programs of width two. *Comput. Complex.* 25, 1 (2016), 217–253. DOI: http://dx.doi.org/10.1007/s00037-015-0114-7

[2] Daniel J. Bates and Luke Oeding. 2011. Toward a salmon conjecture. *Exp. Math.* 20, 3 (2011), 358–370. DOI: http://dx.doi.org/10.1080/10586458.2011.576539

[3] Michael Ben-Or and Richard Cleve. 1992. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.* 21, 1 (1992), 54–58. DOI: http://dx.doi.org/10.1137/0221006

[4] Stuart J. Berkowitz. 1984. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.* 18, 3 (1984), 147–150. DOI: http://dx.doi.org/10.1016/0020-0190(84)90018-8

[5] Dario Bini. 1980. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo* 17, 1 (1980), 87–97. https://doi.org/10.1007/BF02575865

[6] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. 1979. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inf. Process. Lett.* 8, 5 (1979), 234–235. DOI: http://dx.doi.org/10.1016/0020-0190(79)90113-3

[7] Richard P. Brent. 1974. The parallel evaluation of general arithmetic expressions. *J. ACM* 21, 2 (April 1974), 201–206. DOI: http://dx.doi.org/10.1145/321812.321815

[8] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. 2017. On algebraic branching programs of small width. In *Proceedings of the 32nd Computational Complexity Conference (CCC'17) (Leibniz International Proceedings in Informatics (LIPIcs))*, Ryan O'Donnell (Ed.), Vol. 79. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 20:1–20:31. DOI: http://dx.doi.org/10.4230/LIPIcs.CCC.2017.20

[9] Peter Bürgisser. 2000. *Completeness and Reduction in Algebraic Complexity Theory.* Algorithms Comput. Math., Vol. 7. Springer-Verlag, Berlin. xii+168 pages. DOI: http://dx.doi.org/10.1007/978-3-662-04179-6

[10] Peter Bürgisser. 2004. The complexity of factors of multivariate polynomials. *Found. Comput. Math.* 4, 4 (2004), 369–396. DOI: http://dx.doi.org/10.1007/s10208-002-0059-5

[11] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. 1997. *Algebraic Complexity Theory.* Grundlehren Math. Wiss., Vol. 315. Springer-Verlag, Berlin. xxiv+618 pages. DOI: http://dx.doi.org/10.1007/978-3-662-03338-8

[12] Peter Bürgisser and Christian Ikenmeyer. 2011. Geometric complexity theory and tensor rank. *Proceedings 43rd Annual ACM Symposium on Theory of Computing*, 509–518. DOI: http://dx.doi.org/10.1145/1993636.1993704

[13] Peter Bürgisser and Christian Ikenmeyer. 2013. Explicit lower bounds via geometric complexity theory. *Proceedings 45th Annual ACM Symposium on Theory of Computing*, 141–150. DOI: http://dx.doi.org/10.1145/2488608.2488627

[14] Peter Bürgisser, Joseph M. Landsberg, Laurent Manivel, and Jerzy Weyman. 2011. An overview of mathematical issues arising in the geometric complexity theory approach to VP ≠ VNP. *SIAM J. Comput.* 40, 4 (2011), 1179–1209. DOI: http://dx.doi.org/10.1137/090765328

[15] Charles Conley and Valentin Ovsienko. 2017. Rotundus: triangulations, Chebyshev polynomials, and Pfaffians. arXiv:1707.09106.

[16] Michael Forbes. 2016. Some Concrete Questions on the Border Complexity of Polynomials. *Proceedings of the Workshop on Algebraic Complexity Theory (WACT'16).* Retrieved from https://www.youtube.com/watch?v=1HMogQIHT6Q.

[17] Joshua A. Grochow. 2015. Unifying known lower bounds via geometric complexity theory. *Comput. Complex.* 24, 2 (2015), 393–475. DOI: http://dx.doi.org/10.1007/s00037-015-0103-x

[18] Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. 2016. Boundaries of VP and VNP. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, Vol. 55. 34:1–34:14. DOI: http://dx.doi.org/10.4230/LIPIcs.ICALP.2016.34

[19] Yonghui Guan. 2015. Brill's equations as a GL(V)-module. arXiv:1508.02293

[20] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. 2013. Approaching the chasm at depth four. In *Proceedings of the IEEE Conference on Computational Complexity (CCC'13).* IEEE, Los Alamitos, CA, 65–73. DOI: http://dx.doi.org/10.1109/CCC.2013.16

[21] Jonathan D. Hauenstein, Christian Ikenmeyer, and Joseph M. Landsberg. 2013. Equations for lower bounds on border rank. *Exp. Math.* 22, 4 (2013), 372–383. DOI: http://dx.doi.org/10.1080/10586458.2013.825892

[22] Joseph M. Landsberg. 2006. The border rank of the multiplication of 2×2 matrices is seven. *J. Amer. Math. Soc.* 19, 2 (2006), 447–459. DOI: http://dx.doi.org/10.1090/S0894-0347-05-00506-0

[23] Joseph M. Landsberg, Laurent Manivel, and Nicolas Ressayre. 2013. Hypersurfaces with degenerate duals and the geometric complexity theory program. *Comment. Math. Helv.* 88, 2 (2013), 469–484. DOI: http://dx.doi.org/10.4171/CMH/292

[24] Joseph M. Landsberg and Mateusz Michałek. 2016. A $2n^2 - \log(n) - 1$ lower bound for the border rank of matrix multiplication. arXiv:1608.07486

[25] Joseph M. Landsberg and Giorgio Ottaviani. 2015. New lower bounds for the border rank of matrix multiplication. *Theory Comput.* 11 (2015), 285–298. DOI: http://dx.doi.org/10.4086/toc.2015.v011a011 arXiv:1112.6007

[26] Thomas Lickteig. 1984. A note on border rank. *Inf. Process. Lett.* 18, 3 (1984), 173–178. DOI: http://dx.doi.org/10.1016/0020-0190(84)90023-1

[27] Meena Mahajan, Nitin Saurabh, and Karteek Sreenivasaiah. 2012. Counting paths in planar width 2 branching programs. In *Proceedings of the 18th Australasian Theory Symposium on Computing*. Australian Computer Society, Inc., 59–68. Retrieved from http://dl.acm.org/citation.cfm?id=2523693.2523701.

[28] Guillaume Malod and Natacha Portier. 2008. Characterizing Valiant's algebraic complexity classes. *J. Complex.* 24, 1 (2008), 16–38. DOI: http://dx.doi.org/10.1016/j.jco.2006.09.006

[29] Ketan D. Mulmuley and Milind Sohoni. 2001. Geometric complexity theory. I. An approach to the P vs. NP and related problems. *SIAM J. Comput.* 31, 2 (2001), 496–526. DOI: http://dx.doi.org/10.1137/S009753970038715X

[30] Ketan D. Mulmuley and Milind Sohoni. 2008. Geometric complexity theory II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.* 38, 3 (2008), 1175–1206. DOI: http://dx.doi.org/10.1137/080718115

[31] Noam Nisan. 1991. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. ACM, 410–418. DOI: http://dx.doi.org/10.1145/103418.103462

[32] Noam Nisan and Avi Wigderson. 1996/97. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complexity* 6, 3 (1996/97), 217–234. DOI: http://dx.doi.org/10.1007/BF01294256

[33] Luke Oeding and Steven V. Sam. 2016. Equations for the fifth secant variety of Segre products of projective spaces. *Exp. Math.* 25, 1 (2016), 94–99. DOI: http://dx.doi.org/10.1080/10586458.2015.1037872

[34] Ran Raz. 2009. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM* 56, 2 (2009), Art. 8, 17. DOI: http://dx.doi.org/10.1145/1502793.1502797

[35] Herbert John Ryser. 1963. *Combinatorial Mathematics*. Published by The Mathematical Association of America, distributed by John Wiley and Sons, Inc., New York. xiv+154 pages.

[36] Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. 2009. The power of depth 2 circuits over algebras. In *Proceedings of the IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Vol. 4. 371–382. DOI: http://dx.doi.org/10.4230/LIPIcs.FSTTCS.2009.2333 arXiv:0904.2058

[37] Ramprasad Saptharishi. 2016. A survey of lower bounds in arithmetic circuit complexity 3.0.2. Retrieved from https://github.com/dasarpmar/lowerbounds-survey.

[38] Volker Strassen. 1983. Rank and optimal computation of generic tensors. *Linear Algebra Appl.* 52/53 (1983), 645–685. DOI: http://dx.doi.org/10.1016/0024-3795(83)80041-X

[39] Volker Strassen. 1987. Relative bilinear complexity and matrix multiplication. *J. Reine Angew. Math.* 375/376 (1987), 406–443. DOI: http://dx.doi.org/10.1515/crll.1987.375-376.406

[40] Leslie G. Valiant. 1979. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*. ACM, New York, 249–261. DOI: http://dx.doi.org/10.1145/800135.804419

[41] Leslie G. Valiant. 1980. *Reducibility by Algebraic Projections*. University of Edinburgh, Department of Computer Science. Internal Report.

[42] Jeroen Zuiddam. 2017. A note on the gap between rank and border rank. *Linear Algebra Appl.* 525 (2017), 33–44. DOI: http://dx.doi.org/10.1016/j.laa.2017.03.015

Received May 2017; revised February 2018; accepted April 2018