UvA-DARE (Digital Academic Repository)

On algebraic branching programs of small width

Bringmann, K.; Ikenmeyer, C.; Zuiddam, J.

On Algebraic Branching Programs of Small Width[*]

Karl Bringmann[1], Christian Ikenmeyer[2], and Jeroen Zuiddam[3]

1   **Max-Planck-Institut für Informatik, Saarland Informatics Campus, Saarbrücken, Germany**
    `kbringma@mpi-inf.mpg.de`
2   **Max-Planck-Institut für Informatik, Saarland Informatics Campus, Saarbrücken, Germany**
    `cikenmey@mpi-inf.mpg.de`
3   **Centrum Wiskunde & Informatica, Amsterdam, The Netherlands**
    `j.zuiddam@cwi.nl`

—— **Abstract** ——

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 $\overline{\mathbf{VP}_e}$ with a strikingly simple complete polynomial (in characteristic different from 2) whose recursive definition is similar to the Fibonacci numbers. Further understanding this polynomial seems to be a promising route to new formula 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 $\mathbf{VNP}$ 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.

1   Introduction

Let $\mathbf{VP}_e$ denote the class of families of polynomials with polynomially bounded formula size and let $\mathbf{VP}_s$ denote the class of families of polynomials that can be written as determinants of matrices of polynomially bounded size whose entries are affine linear forms. In 1979 Valiant [53] proved his famous result $\mathbf{VP}_e \subseteq \mathbf{VP}_s$. The question whether this inclusion is

---

strict is a long-standing open question in algebraic complexity theory: Can the determinant polynomial $\det_n := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^n x_{i,\sigma(i)}$ be computed by formulas of polynomially bounded size? Motivated by this question we study the 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). We present a simple description of the closure $\overline{\mathbf{VP}_e}$ and of a $\overline{\mathbf{VP}_e}$-complete polynomial whose recursive definition is similar to the Fibonacci numbers, given the characteristic is not 2, see Theorem 3.11.

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 [20] gives a long list (e.g., [41, 44, 34, 23, 32, 13]) 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. [38, 39] and the exposition [15, Sec. 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 [53]. Here $\mathbf{VNP}$ is the class of p-definable families, see Section 2 for a precise definition.

The generalized Fibonacci polynomial

We prove that the *generalized Fibonacci polynomial* F_n is $\overline{\mathbf{VP}_e}$-complete under p-degenerations, where F_n is defined via $F_0 := 1$, $F_1 := x_1$, $F_n := x_n F_{n-1} + F_{n-2}$, see Section 3. This means that every family (f_n) in $\overline{\mathbf{VP}_e}$ can be obtained as the 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. This 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 [6]. This immediately motivates the definition of *border Fibonacci complexity* $\underline{L}_{\mathrm{Fib}}(f)$ of a polynomial f, which is the smallest number m such that f can be obtained as $\lim_{j \to \infty} (F_m(\ell_1(j), \ldots, \ell_m(j)))_j$. To make the situation more geometric we allow the $\ell_i(j)$ to be arbitrary affine linear forms. Our results show that border Fibonacci complexity is polynomially equivalent to border formula size. This insight is quite striking because a result of Allender and Wang [2] implies that the Fibonacci complexity *without allowing approximations* can be infinite!

A promising path towards proving formula lower bounds, for example for the determinant or the permanent, 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

$$X_m := \{f \in \mathbb{C}[x_1, \ldots, x_n]_d \mid \underline{L}_{\mathrm{Fib}}(f) \leq m\}$$

as an affine subvariety (X_m is the closure of the set of affine projections of F_m 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 $\mathrm{GL}(\mathbb{C}^n)$ acts canonically on X_m, making X_m an affine $\mathrm{GL}(\mathbb{C}^n)$-variety. If we find a polynomial $\mathcal{F}$ that vanishes identically on X_m, then a nonzero evaluation $\mathcal{F}(f) \neq 0$ implies that $\underline{L}_{\mathrm{Fib}}(f) > m$. This approach looks feasible given the very simple structure of the

generalized Fibonacci polynomial. This is emphasized by the fact that the action of $\mathrm{GL}(\mathbb{C}^n)$ puts a lot of structure on the coordinate ring of X_m, see for example [12, 5, 34, 13, 26, 22, 42] 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 $\mathbf{VP}_e$ (fan-in 2 arithmetic formulas that use additions and multiplications as their operations), its closure $\overline{\mathbf{VP}_e}$, and the nondeterministic variant $\mathbf{VNP}$. 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. [47].

For this paper 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 well-known that $\mathbf{VP}_k \subseteq \mathbf{VP}_e$ for every $k \geq 1$ (see Proposition 7.1). In 1992, Ben-Or and Cleve [6] showed that $\mathbf{VP}_k = \mathbf{VP}_e$ for all $k \geq 3$ (we review the proof, see Theorem 6.1). In 2011 Allender and Wang [2] 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 3.8) is that the closure of $\mathbf{VP}_2$ and the closure of $\mathbf{VP}_e$ are equal,

$$\overline{\mathbf{VP}_2} = \overline{\mathbf{VP}_e}. \tag{1}$$

Interestingly, as a direct corollary of (1) 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 5.10), 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 (1) we obtain our results about the generalized Fibonacci polynomial that we mentioned before.

VNP via affine linear forms

We define the classes $\mathbf{VNP}_e$ and $\mathbf{VNP}$ in the natural way. In 1980, Valiant [54] showed that $\mathbf{VNP}_e = \mathbf{VNP}$ and in this paper we will always view $\mathbf{VNP}$ as the nondeterministic analog of $\mathbf{VP}_e$. To $\mathbf{VP}_1$ and $\mathbf{VP}_2$ we similarly associate nondeterministic analogs $\mathbf{VNP}_1$ and $\mathbf{VNP}_2$ (see Section 2). Using interpolation techniques it is possible to deduce $\mathbf{VNP}_2 = \mathbf{VNP}$ from (1), provided the field is infinite. Using more sophisticated techniques we strengthen this result to get our second main result (Theorem 4.2):

$$\mathbf{VNP}_1 = \mathbf{VNP}. \tag{2}$$

That is, a family (f_n) is contained in $\mathbf{VNP}$ iff f_n can be written as a hypercube summation of polynomially bounded dimension over a product of polynomially many affine linear

forms. Using (2) it is then easy to verify that $\mathbf{VP}_1 \subsetneq \mathbf{VNP}_1$ and using [2] 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 2 elements, see Section 9.

Restricted ABP edge labels

Several more results on small-width ABPs, approximation closures, and hypercube summations are proved throughout this paper. For example, in Section 5 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 A in Appendix A. As another example, we strengthen (2) as follows (Theorem 6.2): A family (f_n) is contained in $\mathbf{VNP}$ 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 Related work

In the boolean setting as well as in the algebraic setting finding lower bounds for the formula size of explicit problems is considered a major open problem. For the boolean setting we refer the reader to the line of papers [49, 4, 28, 43, 25, 50], which results in an explicit function with formula size $\Omega(n^3/(\log^2 n \log \log n))$.

In the algebraic setting the smallest formula for the determinant has size $\mathcal{O}(n^{\log n})$, which can be deduced from e.g. [27]. The best known lower bound on the formula size of $\det_n$ is $\Omega(n^3)$ by [29]. That paper also gives a quadratic lower bound for an explicit polynomial (note that the lower bound for the determinant is not quadratic in the number of variables).

Toda [52] proved that several definitions for the class $\mathbf{VP}_s$ are equivalent, see also [36]. In particular $\mathbf{VP}_s$ is the class of polynomials that can be written as determinants of matrices of polynomially bounded size whose entries are affine linear forms. Due to its pure mathematical formulation, lower bounds for this *determinantal complexity* attracted the attention of geometers [37, 32, 3]. Moreover, Mulmuley and Sohoni's geometric complexity approach [38, 39] is also mainly focused on lower bounds for the determinantal complexity and the symmetries of the determinant polynomial play a key role in their work. Recently [14] showed that it is not possible to prove superpolynomial lower bounds on the determinantal complexity using only information about the occurrences/non-occurrences of irreducible representations in the coordinate rings of the orbit closures of the determinant and the (padded) permanent. This disproves a major conjecture in geometric complexity theory. The proof in [14] is fairly general and also holds for lower bounds on the formula size. Only very recently the formula size analog to determinantal complexity, the *iterated matrix multiplication complexity* was studied from a geometric perspective [19].

There is a large number of publications on lower bounds for constant *depth* circuits and formulas (with superconstant fan-in), see e.g. [1, 30, 24, 51], which recently led to the celebrated result [23] that the permanent does not admit size $2^{o(\sqrt{m})}$ homogeneous $\Sigma\Pi\Sigma\Pi$ circuits in which the bottom fan-in is bounded by $\sqrt{m}$. In the light of the previous depth-reduction results this seemed very close to separating $\mathbf{VP}$ from $\mathbf{VNP}$. Several very recent results [16, 18] indicate that new ideas are needed to separate $\mathbf{VP}$ from $\mathbf{VNP}$.

Ben-Or and Cleve [6] proved that a family of polynomials has polynomially bounded formula size if and only if it is computable by width-3 ABPs of polynomial size. An excellent exposition on the history of small-width computation can be found in [2], along with an explicit polynomial that cannot be computed by width-2 ABPs: $x_1 x_2 + x_3 x_4 + \cdots + x_{15} x_{16}$.

Saha, Saptharishi and Saxena [46, Cor. 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 $\mathbf{C}$ the relation between $\mathbf{C}$ and $\overline{\mathbf{C}}$ has not been an active object of study. As pointed out recently by Forbes [17], Nisan's result [40] implies that $\mathbf{C} = \overline{\mathbf{C}}$ for $\mathbf{C}$ being the class of size-k algebraic branching programs on noncommuting variables. Recently, a structured study of $\overline{\mathbf{VP}}$ and $\overline{\mathbf{VP}_s}$ has been started, see [21]. By far the most work in lower bounds for topological approximation algorithms has been done in the area of bilinear complexity, dating back to [7, 48, 35] and more recently [31, 34, 26, 55, 33], to list a few.

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 generalized Fibonacci polynomial is $\overline{\mathbf{VP}_e}$-complete under p-degenerations. In Section 4 we prove the second main result: a new description of $\mathbf{VNP}$ 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 nondeterminism 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, $\mathbf{C}$. For an introduction to the algebraic complexity classes $\mathbf{VP}_e$, $\mathbf{VP}$, and $\mathbf{VNP}$ we refer the reader to [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 $L(g_i) \le c$. 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. [10]). Namely, a polynomial $f \in \mathbb{C}[\mathbf{x}]$ satisfies $\underline{L}(f)^{\mathrm{alg}} \le c$ 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 $L_{\mathbb{C}(\varepsilon)}(h) \le c$, 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 paper.

▶ **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*.

Interestingly, for subtle reasons, taking the approximation closure $\mathbf{C} \mapsto \overline{\mathbf{C}}$ is *not* idempotent in general and hence not a closure operator, but for reasonable classes (like $\mathbf{VP}_k$, $\mathbf{VP}_e$, and $\mathbf{VP}$) it is.

One can think of $\mathbf{VNP}$ as a "nondeterminism closure" of $\mathbf{VP}$. We want to use the nondeterminism closure for general classes.

▶ **Definition 2.2.** Let $\mathbf{C}$ be a class. The class $\mathrm{N}(\mathbf{C})$ consists of families (f_n) with the following property: there is a family $(g_n) \in \mathbf{C}$ and $p(n), q(n) \in \mathrm{poly}(n)$ 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 Section 5. We remark that the map $\mathbf{C} \mapsto \mathrm{N}(\mathbf{C})$ trivially satisfies all properties of being a closure operator.

3   Approximate width-2 ABPs and formula size

As mentioned in the introduction, Allender and Wang [2] showed that there exist polynomials that cannot be computed by any width-2 ABP, for example the polynomial $x_1 x_2 + x_3 x_4 + \cdots + x_{15} x_{16}$. 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* $\mathrm{char}(\mathbb{F}) \neq 2$.

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

In order 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 $M_k \cdots M_2 M_1$.

For 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 $\left(\begin{smallmatrix} 1 & 0 \end{smallmatrix} \right) M_\ell \cdots M_2 M_1 \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix} \right)$ 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 in F, G are e_f, e_g, respectively, then the error degree of H is at most $e_f + e_g$.*
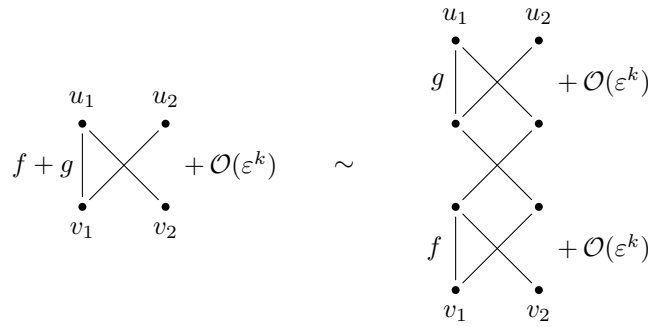
Figure 1 Addition construction for Lemma 3.2.

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 Fig. 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 in 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

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

To obtain $H' \in Q(f^2) + \mathcal{O}(\varepsilon)$, 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 Fig. 2 and Fig. 3 for a pictorial description. ◀

▶ **Lemma 3.4** (Multiplication). *Let $f, g \in \mathbb{F}[\mathbf{x}]$ be polynomials such that some $F \in Q(f/2) + \mathcal{O}(\varepsilon^3)$ and some $G \in Q(g) + \mathcal{O}(\varepsilon^3)$ can be written as the product of n and m primitive Q-matrices respectively. Then some $H \in Q(f \cdot g) + \mathcal{O}(\varepsilon)$ can be written as the product of $4n + 4m + 37$ primitive Q-matrices. Moreover, if the error degrees in F, G are e_f, e_g, respectively, then the error degree of H is at most $4 \cdot e_f + 4 \cdot e_g + 12$.*
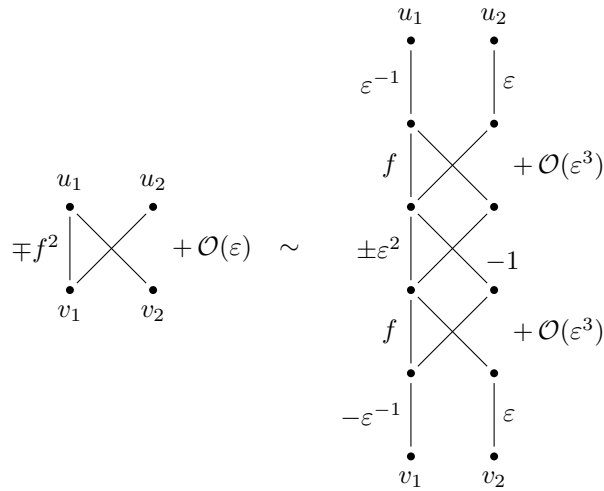
Figure 2 Squaring construction for Lemma 3.3.
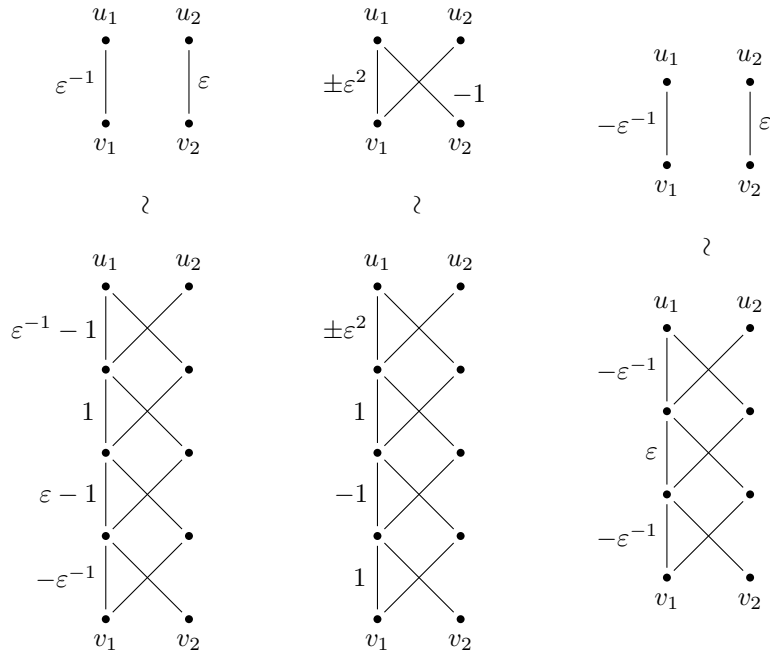
Figure 3 Squaring construction subroutines for C, B, and A for Lemma 3.3.

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), $(f/2 + g) + \mathcal{O}(\varepsilon^3)$ 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), $Q(-(f/2)^2) + \mathcal{O}(\varepsilon)$, $Q(-g^2) + \mathcal{O}(\varepsilon)$, and $Q((f/2+g)^2) + \mathcal{O}(\varepsilon)$ 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, $Q(f \cdot g) + \mathcal{O}(\varepsilon) = Q(-(f/2)^2 + (-g^2) + (f/2+g)^2) + \mathcal{O}(\varepsilon)$ 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 Fig. 4 for a pictorial description. ◀
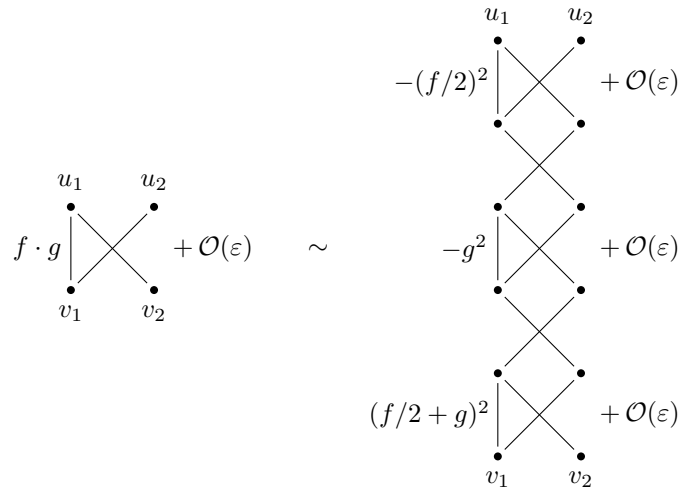
Figure 4 Multiplication construction for Lemma 3.4.

▶ **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 $F \in Q(\alpha f) + \mathcal{O}(\varepsilon)$ 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 also $Q(\alpha/2)$ can be written directly (also with error degree 0), we can use the multiplication lemma (Lemma 3.4), to write $Q(\alpha f) + \mathcal{O}(\varepsilon)$ 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 $Q(\beta g) + \mathcal{O}(\varepsilon)$ and $Q(\gamma h) + \mathcal{O}(\varepsilon)$ 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 $Q(\alpha f) + \mathcal{O}(\varepsilon) = Q(\alpha g + \alpha h) + \mathcal{O}(\varepsilon)$ 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 $Q(\beta g) + \mathcal{O}(\varepsilon^3)$ and $Q(\gamma h) + \mathcal{O}(\varepsilon^3)$ 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 $Q(\alpha f) + \mathcal{O}(\varepsilon) = Q((\alpha \cdot g) \cdot h) + \mathcal{O}(\varepsilon)$ 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 $4(3 \cdot e_g) + 4(3 \cdot e_h) + 12 = 12(e_g + e_h + 1) \leq 24 \cdot 12 \cdot 25^{d-1} + 12 \leq 12 \cdot 25^d$. ◀

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

Proof. The construction uses the classical depth-reduction theorem for formulas by Brent [8], for which a modern proof can be found in the survey of Saptharishi [47, Lemma 5.5]: If a family (f_n) has polynomially bounded formula size, then there are formulas computing f_n that have size $\mathrm{poly}(n)$ and depth $\mathcal{O}(\log n)$. 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 $F \in Q(f_n) + \mathcal{O}(\varepsilon)$ which is a product of polynomially many primitive Q-matrices such that F has polynomially bounded error degree. The width-2 ABP computing $f_n + \mathcal{O}(\varepsilon)$ is given by $(\begin{smallmatrix} 1 & 0 \end{smallmatrix}) F (\begin{smallmatrix} 1 \\ 0 \end{smallmatrix})$. ◀

▶ **Example 3.7.** Following the construction in Theorem 3.1 we get the following ABP for approximating the polynomial $x_1 x_2 + x_3 x_4 + \cdots + x_{15} x_{16}$, which cannot be computed by any width-2 ABP. Let

$$
F(x_1, x_2) = \begin{pmatrix} \frac{1}{\varepsilon} - \frac{\varepsilon x_1}{2} & -\frac{x_1}{2\varepsilon} \\ \varepsilon^3 & \varepsilon \end{pmatrix} \begin{pmatrix} \frac{1}{2}(x_1 - 2x_2)\varepsilon^2 + 1 & \frac{1}{2}(x_1 - 2x_2) \\ \varepsilon^2 & 1 \end{pmatrix}
$$

$$
\cdot \begin{pmatrix} \frac{x_1 \varepsilon^2}{2} + 1 & -\frac{x_1}{2} \\ -\varepsilon^2 & 1 \end{pmatrix} \begin{pmatrix} \frac{x_1 + 2x_2}{2\varepsilon} & \varepsilon \\ \varepsilon^{-1} & 0 \end{pmatrix}.
$$

Then

$$
F(x_1, x_2) = \begin{pmatrix} x_1 x_2 & 1 \\ 1 & 0 \end{pmatrix} + \mathcal{O}(\varepsilon).
$$

Using the addition lemma Lemma 3.2 we get

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

▶ **Corollary 3.8.** $\overline{\mathbf{VP}_2} = \overline{\mathbf{VP}_e}$ *and* $\overline{\mathbf{VP}_2}^{\text{poly}} = \overline{\mathbf{VP}_e}^{\text{poly}}$ *when* $\text{char}(\mathbb{F}) \neq 2$.

Proof. The inclusion $\mathbf{VP}_2 \subseteq \mathbf{VP}_e$ is standard (see Proposition 7.1). Taking closures on both sides, we obtain $\overline{\mathbf{VP}_2} \subseteq \overline{\mathbf{VP}_e}$ and $\overline{\mathbf{VP}_2}^{\text{poly}} \subseteq \overline{\mathbf{VP}_e}^{\text{poly}}$.

On the other hand, when $\text{char}(\mathbb{F}) \neq 2$, we have the inclusion $\mathbf{VP}_e \subseteq \overline{\mathbf{VP}_2}^{\text{poly}}$ (Theorem 3.1). By taking closures this implies $\overline{\mathbf{VP}_e} \subseteq \overline{\mathbf{VP}_2}$ and $\overline{\mathbf{VP}_e}^{\text{poly}} \subseteq \overline{\mathbf{VP}_2}^{\text{poly}}$. ◀

▶ **Corollary 3.9.** $\overline{\mathbf{VP}_2}^{\text{poly}} = \mathbf{VP}_e$ *when* $\text{char}(\mathbb{F}) \neq 2$ *and* $\mathbb{F}$ *is infinite.*

Proof. By Corollary 3.8 we have $\overline{\mathbf{VP}_2}^{\text{poly}} = \overline{\mathbf{VP}_e}^{\text{poly}}$. It remains to show the equality $\overline{\mathbf{VP}_e}^{\text{poly}} = \mathbf{VP}_e$. We give a proof of this via a standard interpolation argument in Section 8. ◀

As a consequence of Proposition 3.5, we obtain a new description of $\overline{\mathbf{VP}_e}$ as follows. We define the *generalized Fibonacci polynomial* $F_n(x_1, \ldots, x_n)$ by $F_0 := 1$, $F_1 := x_1$, and $F_n := x_n F_{n-1} + F_{n-2}$ for all $n \geq 2$. The name comes from the fact that $F_n(1, 1, \ldots, 1)$ is the nth Fibonacci number and $F_n(x, x, \ldots, x)$ is the nth Fibonacci polynomial. Another description of the polynomial F_n is that it 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 3.10.** 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 *Fibonacci complexity of f*. A polynomial is a *degeneration of F_m* if there exist parametrized affine linear forms $\ell_1(\varepsilon), \ldots, \ell_m(\varepsilon)$ such that $f = F_m(\ell_1(\varepsilon), \ldots, \ell_m(\varepsilon))$. The smallest m such that $f + \mathcal{O}(\varepsilon)$ is a degeneration of F_m we call the *border Fibonacci complexity* of f, and is denoted by $\underline{L}_{\text{Fib}}(f)$. A family (h_n) of polynomials is called $\overline{\mathbf{VP}_e}$-*complete under p-degenerations* if $(h_n) \in \overline{\mathbf{VP}_e}$ and for every $(f_n) \in \overline{\mathbf{VP}_e}$ there exists a polynomially bounded function t such that some polynomial in $f_n + \mathcal{O}(\varepsilon)$ is a degeneration of $F_{t(n)}$.

The Fibonacci complexity is not always finite ([2]), but Proposition 3.6 shows that the border Fibonacci complexity $\underline{L}_{\mathrm{Fib}}(f)$ is always finite and that $\overline{\mathbf{VP}_{\mathrm{e}}}$ can be characterized as the class of families with polynomially bounded border Fibonacci complexity:

▶ **Theorem 3.11.** $\overline{\mathbf{VP}_{\mathrm{e}}} = \{(f_n) \mid \underline{L}_{\mathrm{Fib}}(f_n) \in \mathrm{poly}(n)\}.$

Proof. Clearly the right-hand side is contained in the left-hand side. $\mathbf{VP}_{\mathrm{e}}$ 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. ◀

Theorem 3.11 says that (F_n) is $\overline{\mathbf{VP}_{\mathrm{e}}}$-complete under p-degenerations. From the proof of Proposition 3.5 it follows that also (F_{2n+1}) is $\overline{\mathbf{VP}_{\mathrm{e}}}$-complete under p-degenerations, that is, we only need the F_m with odd index m (this follows from $\det(Q(f)) = -1$).

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

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

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 Fibonacci complexity* analogously to the border Fibonacci complexity by replacing F_n by C_n in Definition 3.10. Analogously to Theorem 3.11 we now see that the families (C_n) and (C_{2n+1}) are both $\overline{\mathbf{VP}_{\mathrm{e}}}$-complete under p-degenerations.

▶ Remark (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 $1 \le i < n$. 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 $x_S \coloneqq \prod_{i \in S} x_i$. Then $F_n(x_1, \ldots, x_n) = \sum_S x_S$, where the sum is over all supporting sets.

We define a *cyclicly adjacent pair* as a set that is either an adjacent pair or the set $\{1, n\}$, if $1 \ne n$. 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 (Planarity). We remark that the product of two Q-matrices $Q(x)Q(y)$ can be rewritten as $Q(x)Q(y) = \left(Q(x)\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)\right)\left(\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)Q(y)\right)$. We also have $Q(x)\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) = \left(Q(x)\left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)\right)\left(\begin{smallmatrix} b \\ a \end{smallmatrix}\right)$. Consider a width-2 ABP that is a product of primitive Q-matrices,

$$\left(\begin{smallmatrix} a & b \end{smallmatrix}\right)Q(\ell_1)Q(\ell_2)\cdots Q(\ell_k)\left(\begin{smallmatrix} c \\ d \end{smallmatrix}\right).$$

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 Fig. 5 for an example with three Q-matrices.

4 VNP via products of affine linear forms

Valiant proved the following characterization of **VNP** [54] (see also [11, Thm. 21.26], [9, Thm. 2.13] and [36, Thm. 2]).
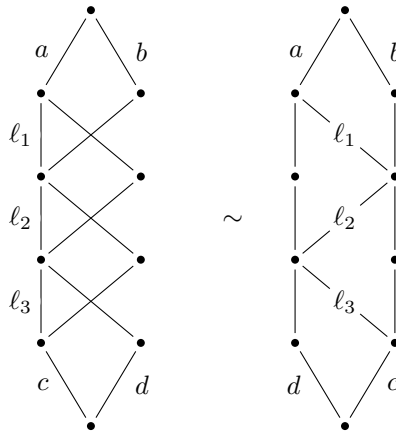
Figure 5 Making an ABP consisting of three primitive Q-matrices planar.

▶ **Theorem 4.1** (Valiant [54]). $\mathbf{VNP}_e = \mathbf{VNP}$.

We strengthen Valiant's characterization of $\mathbf{VNP}$ from $\mathbf{VNP}_e$ 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 $\mathbf{VNP}$-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 $\mathbf{VP}_e = \mathbf{VP}_3$ by [6]. We present the second proof in Section 6. The advantage of the second proof is that we can restrict the ABP edge labels to affine linear forms that have at most 2 variables, see Theorem 6.2. 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 $\mathbf{VNP}$-complete under p-projections, see for example [9, Thm. 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 [45, Thm. 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 $b = (b[1], \ldots, b[n]) \in \{0,1\}^n$, we can rewrite the above as

$$\mathrm{per}_n = (-1)^n \sum_{b \in \{0,1\}^n} \Big(\prod_{k \in [n]} (1 - 2b[k]) \Big) \prod_{j \in [n]} \sum_{i \in [n]} b[i]\, x_{i,j}$$

$$= (-1)^n \sum_{b \in \{0,1\}^n} \Big(\prod_{k \in [n]} (1 - 2b[k]) \Big) \sum_{i_1, \ldots, i_n \in [n]} \prod_{j \in [n]} b[i_j]\, x_{i_j, j}$$

For notational convenience we use square brackets not only to refer to sets ($[n] := \{1, \ldots, n\}$), but also to entries in a list ($b[k] := b_k$). We now introduce new Boolean variables $a[i,j]$, $1 \leq i \leq n-1$, $1 \leq j \leq n$, and 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

$$\mathrm{per}_n = (-1)^n \sum_{b \in \{0,1\}^n} \left(\prod_{k \in [n]} (1 - 2b[k]) \cdot \sum_a \prod_{i,j \in [n]} \big(1 + (x_{i,j} - 1)(a[i-1,j] - a[i,j])\big) \right. \tag{3}$$
$$\left. \cdot \big(1 + (b[i] - 1)(a[i-1,j] - a[i,j])\big) \cdot \big(1 + (a[i-1,j] - 1)a[i,j]\big) \right),$$

where the second sum is over all Boolean assignments of $a[i,j]$. The idea is to encode the indices $i_1, \ldots, i_n$ 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 (3) 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 $a[0,j], \ldots, a[n,j]$ a 0 is followed by a 1, then $\prod_{i \in [n]}(1 + (a[i-1,j]-1)a[i,j] = 0$. Conversely, if $(a[0,j], \ldots, a[n,j]) = (1, \ldots, 1, 0, \ldots, 0)$, then $\prod_{i \in [n]}(1 + (a[i-1,j]-1)a[i,j]) = 1$. The nontrivial assignments of $(a[0,j], \ldots, a[n,j])$ are thus exactly of the form $(1, \ldots, 1, 0, \ldots, 0)$ where the first 0 occurs at some index $1 \leq z \leq n$ (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 $i \neq z$, it equals 1.

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

This proves (3).

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

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

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

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

$$(x_{i,j} - 2\, c_1[i,j]) \cdot (a[i-1,j] - a[i,j] + 1 - 2\, c_1[i,j])$$

$$\cdot (b[i] - 2\, c_2[i,j]) \cdot (a[i-1,j] - a[i,j] + 1 - 2\, c_2[i,j])$$

$$\left. \cdot (a[i-1,j] - 2\, c_3[i,j]) \cdot (a[i,j] + 1 - 2\, c_3[i,j]) \right),$$

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 $1 \le i,j \le n$, again except for $a[n,j] \coloneqq 0$, and $a[0,j] \coloneqq 1$. This shows that $(\mathrm{per}_n) \in \mathbf{VNP}_1$.   ◀

In Section 9 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   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 [2].

▶ **Definition 5.1.** 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 $*$ equal to wst, w or g, the class $\mathbf{VP}_k^*$ consists of all families of polynomials over polynomially many variables that are computed by polynomial-size width-k $*$-ABPs. In the rest of this paper the star will act as a variable from $\{\mathrm{wst}, \mathrm{w}, \mathrm{g}\}$. We write $\mathbf{VP}_k$ if 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. 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 $\mathrm{char}(\mathbb{F}) \ne 2$. 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 A.

5.1   Comparing different types of edge labels in width-2 ABPs

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

▶ **Theorem 5.2.** $\mathbf{VP}_2^{\mathrm{w}} \subsetneq \mathbf{VP}_2^{\mathrm{g}}$.

In fact, we will show the following stronger statement.

▶ **Theorem 5.3.** *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 5.3 we will review and reuse the arguments used by Allender and Wang [2] to show that the polynomial $x_1 x_2 + \cdots + x_{15} x_{16}$ cannot be computed by any width-2 g-ABP.

For the proof of Theorem 5.3 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.

▶ **Definition 5.4.** A matrix $A \in \mathbb{L}^{2 \times 2}$ is called *inherently nondegenerate* (indg) when $\det(A) \in \mathbb{F} \setminus \{0\}$.

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 5.5** ([2, Thm. 3.9 and Lem. 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 affine 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 [2].

▶ **Lemma 5.6** ([2]). *Let p be a polynomial. If p is computed by a w-ABP that has an inner matrix containing 4 variables, then there is an assignment π of 4 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 nonzero and not-indg (that is, $\det(A)$ is either 0 or a nonconstant polynomial). 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 a matrix Γ_i with only constant entries and with rank 1. Then there is a 2×1 matrix $\Gamma_{i,2}$ and a 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 Γ_i *factors p into p_2p_1*. Recall that $\mathrm{H}(p)$ denotes the highest-degree homogeneous part of a polynomial p. The following is implicit in [2].

▶ **Theorem 5.7** ([2]). *Let p be a polynomial computed by a w-ABP Γ. Then there is an assignment π of at most 6 variables such that one of the following is true:*
1. $\pi(p)$ *is affine linear (including constant), or*
2. $\mathrm{H}(\pi(p))$ *is a product of two polynomial of positive degree.*

Proof. Let $(\Gamma_k, \ldots, \Gamma_1)$ be the matrices of Γ, so that $p = \Gamma_k \cdots \Gamma_1$. If there is a Γ_i containing 4 variables, then there is an assignment π of these 4 variables with $\pi(p) = 0$ (Lemma 5.6), so we are in case 1. Otherwise, all Γ_i have at most 3 variables. If the inner Γ_i are all indg, then $\mathrm{H}(p)$ is a product of linear forms (Theorem 5.5), so we are in case 1 or 2. Otherwise, there is at least one not-indg inner matrix. Consider the nonempty subsequence $\mathcal{M} = (M_\ell, \ldots, M_1)$ of not-indg inner matrices. For each M_i there is an assignment π of at most 3 variables such that $\pi(M_i)$ has only constant entries and rank 1. We consider four possible situations.

1. *There is an $M \in \mathcal{M}$ and an assignment π of at most 3 variables such that $\pi(M)$ factors $\pi(p)$ into a product of two constants or a product of two polynomials with positive degree. Then we are in case 1 or 2.*
2. *There is an assignment π of at most 3 variables such that $\pi(M_1)$ factors $\pi(p)$ into p_2p_1 with p_2 a constant and p_1 not constant. Then p_1 is computed by an ABP consisting of indg inner matrices (since M_1 is the right-most not-indg inner matrix) and hence $\mathrm{H}(p_1)$ is a product of linear forms (Theorem 5.5), so we are in case 1 or 2.*
3. *There is an assignment π of at most 3 variables such that $\pi(M_\ell)$ factors $\pi(p)$ into p_2p_1 with p_2 not a constant and p_1 a constant. Then p_2 is computed by an ABP consisting of indg inner matrices (since M_2 is the left-most not-indg inner matrix) and one proceeds as in the previous situation.*
4. *Remaining situation. In the remaining situation we do the following. Let M_i be the left-most matrix in $\mathcal{M}$ such that there is an assignment π of at most 3 variables such that $\pi(M_i)$ factors $\pi(p)$ into p_2p_1 with p_2 not a constant and p_1 constant. Then there is an assignment σ of at most 3 variables such that $\sigma(\pi(M_{i+1}))$ factors $p_2 = p_3p_4$ with p_3 constant and p_4 not constant. Then p_4 is computed by an ABP consisting of indg matrices, and so $\mathrm{H}(p_4)$ is a product of homogeneous linear forms. Therefore we are in case 1 or 2.* ◀

▶ **Theorem 5.3** (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 5.7 there is an assignment π of at most 6 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 $x_ix_j + x_kx_\ell$ 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. ◀
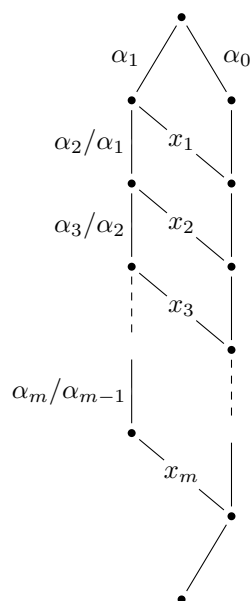
Figure 6 Width-2 wst-ABP computing $\ell(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_m x_m$.

5.2   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 $\mathbf{VP}_1^* \subseteq \mathbf{VP}_2^*$, but this does not give a complete description of all inclusions among these classes. The following two propositions realize a complete description among $\mathbf{VP}_1^*$ and $\mathbf{VP}_2^{\mathrm{wst}}$.

▶ **Proposition 5.8.** $\mathbf{VP}_1^{\mathrm{g}} \subseteq \mathbf{VP}_2^{\mathrm{wst}}$.

Proof. Let $(p_n) \in \mathbf{VP}_1^{\mathrm{g}}$. Then each p_n is a product of $\mathrm{poly}(n)$ affine linear forms in $\mathrm{poly}(n)$ 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 $\alpha_1, \ldots, \alpha_m \in \mathbb{F} \setminus \{0\}$. We can compute $\ell(\mathbf{x})$ with the width-2 wst-ABP in Fig. 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 $\mathrm{poly}(n)$ size. Thus, $(p_n) \in \mathbf{VP}_2^{\mathrm{wst}}$.   ◀

▶ **Proposition 5.9.** $\mathbf{VP}_1^{\mathrm{wst}} \subsetneq \mathbf{VP}_1^{\mathrm{w}} \subsetneq \mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VP}_2^{\mathrm{wst}}$.

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 5.8. 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 $\mathbf{VP}_1^{\mathrm{g}} \subsetneq \mathbf{VP}_2^{\mathrm{wst}}$.   ◀

5.3   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 5.10.** $\mathbf{VP}_1^* = \overline{\mathbf{VP}_1^*}$.

Proof. Trivially, $\mathbf{VP}_1^* \subseteq \overline{\mathbf{VP}_1^*}$. To prove the opposite inclusion, let $(f_n) \in \overline{\mathbf{VP}_1^*}$. 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 $\mathrm{poly}(n)$ affine linear forms in $\mathbb{F}(\varepsilon)[\mathbf{x}]$ in $\mathrm{poly}(n)$ 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 $(f_n) \in \mathbf{VP}_1^*$.　　◀

5.4　Nondeterminism in width-1 ABPs

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

▶ **Proposition 5.11.**
  ▬　$\mathbf{VP}_1^* = \mathbf{VNP}_1^*$ *for* $*$ *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, $\mathbf{VP}_1^* \subseteq \mathbf{VNP}_1^*$. 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 $2x_1 x_2 + x_1 + x_2 + 1 = (\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2)(\beta_0 + \beta_1 x_1 + \beta_2 x_3)$. 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 $(\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2)(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$ has degree 1) and since $\alpha_2 \beta_2 = 0$, we have $\beta_2 = 0$. Hence, $2x_1 x_2 + x_1 + x_2 + 1 = (\alpha_0 + \alpha_2 x_2)(\beta_0 + \beta_1 x_1)$. 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 5.12.** It follows directly from Proposition 5.11 and Proposition 5.9 that we have strict inclusions $\mathbf{VNP}_1^{\mathrm{wst}} \subsetneq \mathbf{VNP}_1^{\mathrm{w}} \subsetneq \mathbf{VNP}_1^{\mathrm{g}}$, when $\mathrm{char}(\mathbb{F}) \neq 2$.

6 Alternative proof of $\mathbf{VNP}_1 = \mathbf{VNP}$ via $\mathbf{VP}_3$

Recall that in Section 4 we proved that

$$\mathbf{VNP}_1^{\mathrm{g}} = \mathbf{VNP} \tag{4}$$

using the completeness of the permanent (Theorem 4.2). We will present an alternative proof of (4) inspired by the proof of the following theorem by Ben-Or and Cleve. The alternative proof of (4) has the benefit that it can be extended to show a slightly stronger result, see Theorem 6.2.

▶ **Theorem 6.1** (Ben-Or and Cleve, [6]). *For $k \geq 3$, $\mathbf{VP}_k^* = \mathbf{VP}_{\mathrm{e}}$.*

Proof. Proposition 7.1 says that $\mathbf{VP}_k^* \subseteq \mathbf{VP}_{\mathrm{e}}$. We will prove that $\mathbf{VP}_{\mathrm{e}} \subseteq \mathbf{VP}_3^{\mathrm{wst}}$, from which it follows that $\mathbf{VP}_{\mathrm{e}} \subseteq \mathbf{VP}_k^*$ and thus $\mathbf{VP}_k^* = \mathbf{VP}_{\mathrm{e}}$. 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
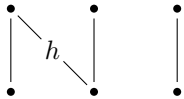
We call the following matrices *primitive*:

- $M(h)$ 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 5.1).

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 ([8]) f_n can then also be computed by a formula of size $\mathrm{poly}(n)$ and depth $d(n) \in \mathcal{O}(\log n)$.

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

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

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

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

so $f_n(\mathbf{x})$ can be computed by a width-3 wst-ABP of size $\text{poly}(n)$, 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) primitives $A_1, \ldots, A_m$ such that

$$A_1 \cdots A_m = \begin{pmatrix} 1 & 0 & 0 \\ h & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{with } m \in \mathcal{O}(4^d). \tag{5}$$

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 primitives. Then $M(f + g)$ equals a product of primitives, 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:
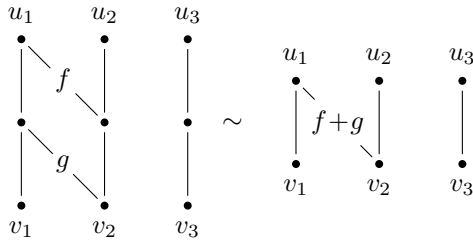

Suppose $h = fg$ is a product of two polynomials f, g and suppose $M(f)$ and $M(g)$ can be written as a product of primitives. Then $M(fg)$ equals a product of primitives, because

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

(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:
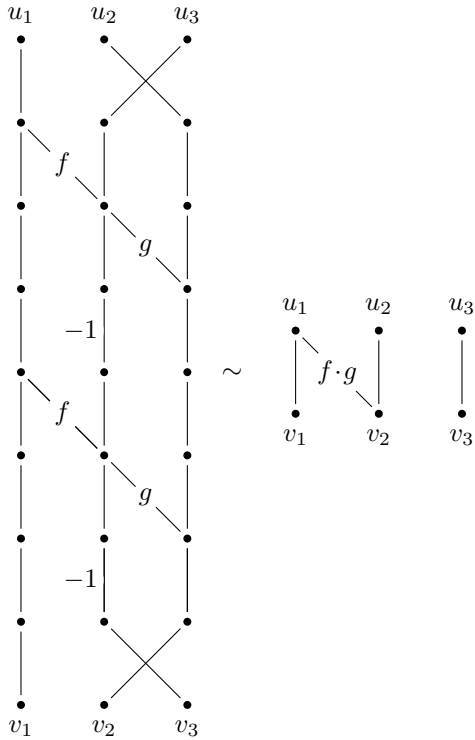
This completes the construction.

The length m of the construction is $m(h) = 1$ for h a variable or constant and recursively $m(f + g) = m(f) + m(g)$, $m(f \cdot g) = 2(m(f) + m(g)) + \mathcal{O}(1)$, so $m \in \mathcal{O}(4^d)$ where d is the formula depth of h. The construction thus satisfies (5), proving the theorem. ◀

We will now give an alternative proof of Theorem 4.2.

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

Proof. Clearly, $\mathbf{VNP}_1^{\mathrm{g}} \subseteq \mathbf{VNP}$ by Proposition 7.1 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 6.1), 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 primitives $A_{n,1}, \ldots, A_{n,t(n)}$ with $t(n) \in \mathrm{poly}(n)$ such that

$$f_n(\mathbf{x}) = (\begin{smallmatrix} 1 & 1 & 1 \end{smallmatrix}) M_{-1,1,0} A_1 \cdots A_m \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$
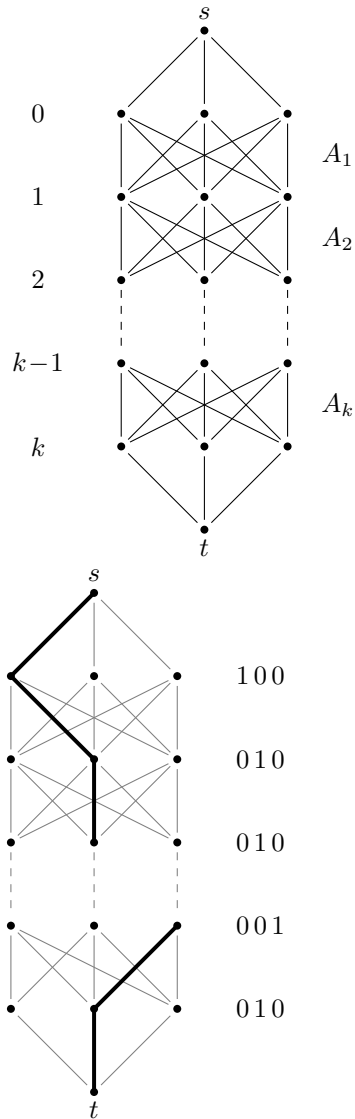
Figure 7 Illustration of layer labelling and path labelling in the proof of Theorem 4.2.

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

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

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

View this expression as a width-3 ABP G, with vertex layers labeled as shown in the left diagram of Fig. 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 $f(\mathbf{x})$,

$$f(\mathbf{x}) = \sum_{j \in [3]^k} A_k[j_k, j_{k-1}] \cdots A_1[j_2, j_1]. \tag{6}$$

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 Fig. 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

$$V(b_1, b_2, b_3) := \prod_{i \in [k]} (b_1[i] + b_2[i] + b_3[i]) \prod_{\substack{s,t \in [3]: \\ s \neq t}} (1 - b_s[i] b_t[i]). \tag{7}$$

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 (6) by a product of affine linear forms $S_i(A_i)$ with variables $\mathbf{b}$ and $\mathbf{x}$ as follows

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

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

$$S_i(M(x)) := \big(1 + (x - 1)(b_1[i] - b_1[i-1])\big)$$
$$\cdot \big(1 - (1 - b_2[i]) b_2[i-1]\big)$$
$$\cdot \mathrm{Eq}\big(b_3[i-1], b_3[i]\big).$$

- For any permutation $\pi \in S_3$ define

$$S_i(M_\pi) := \mathrm{Eq}\big(b_1[i-1], b_{\pi(1)}[i]\big)$$
$$\cdot \mathrm{Eq}\big(b_2[i-1], b_{\pi(2)}[i]\big)$$
$$\cdot \mathrm{Eq}\big(b_3[i-1], b_{\pi(3)}[i]\big).$$

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

$$S_i(M_{a,b,c}) := \big(a \cdot b_1[i-1] + b \cdot b_2[i-1] + c \cdot b_3[i-1]\big)$$
$$\cdot \mathrm{Eq}\big(b_1[i-1], b_1[i]\big)$$
$$\cdot \mathrm{Eq}\big(b_2[i-1], b_2[i]\big)$$
$$\cdot \mathrm{Eq}\big(b_3[i-1], b_3[i]\big).$$

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 5.12 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 2.2). Then we can adjust the construction in the above proof of Theorem 4.2 to show the following.

▶ **Theorem 6.2.** $\mathbf{VNP}_1^{\mathrm{w}} \subsetneq \mathbf{VNP}_1^{\mathrm{w}+} = \mathbf{VNP}$ *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 5.12. 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 $\mathrm{S}(M(x))$

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

$$\frac{1}{2}\sum_{b \in \{0,1\}}(x+1-2b)(y+1-2b)(z+1-2b) = x + y + z + xyz. \tag{8}$$

Indeed, expression 1 can be replaced by

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

since the unwanted term $b_1[i]b_2[i]b_3[i]$ will always vanish in our construction (because in (7) we multiply with $1 - b_s[i]b_t[i]$ for every $s \neq t$). Similarly 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 (8), by

$$\frac{1}{2}\sum_{c' \in \{0,1\}}\big(b_1[i]+1-2c'\big)\big(-b_1[i-1]+1+1-2c'\big)\big(-2c+1-2c'\big)$$
$$= b_1[i] - b_1[i-1] + 1 - 2c + b_1[i]\big(1 - b_1[i-1]\big)\big(-2c\big).$$

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.   ◀

7   Constant-width ABPs have small formulas

The following well-known proposition says that the iterated product of constant-size matrices can be efficiently computed by a formula.

▶ **Proposition 7.1.** *Let* $k \geq 1$. *Then* $\mathbf{VP}_k \subseteq \mathbf{VP}_\mathrm{e}$.

Proof. Let $(f_n) \in \mathbf{VP}_k$, so f_n has $v(n) \in \text{poly}(n)$ variables. There is a function $m(n) \in$ $\text{poly}(n)$ and there are $k \times k$ matrices $M_{n,1}, \ldots, M_{n,m(n)}$ with affine linear forms as entries, such that $\text{tr}(M_{n,1} \cdots M_{n,m(n)}) = f_n$. We may assume that each affine linear form occurring in $M_{n,i}$ has $v(n)$ variables, since f_n has $v(n)$ variables. We will recursively construct a multi-output formula computing the product $M_{n,1} \cdots M_{n,m(n)}$. From this one can efficiently compute the trace. Let $\text{size}(m, n)$ denote the size of the formula that we construct. Let $w(n) := 2v(n)$. A single matrix $M_{n,i}$ we can compute by a multi-output formula of size $k^2 w(n)$ (the $w(n)$ is needed to compute each affine linear form). So $\text{size}(1, n) = k^2 w(n)$. Suppose that matrices A and B can be computed by a multi-output formulas F and G respectively, each of size s. Then the product AB can be computed by a multi-output formula of size $2ks + c(k)$ with $c(k) \in \mathcal{O}(k^3)$, as follows: take k copies of the formula F for A and take k copies of the formula G for B and appropriately add $c(k)$ $\times$- and $+$-gates in order to perform the matrix multiplication. (The reason that we take k copies of the formulas F and G is that in the matrix multiplication, each input entry is used k times, and in formulas we cannot use intermediate results more than once.) Therefore, the recurrence relation $\text{size}(m, n) = 2k \, \text{size}(m/2, n) + c(k)$ holds. Working out the recurrence relation gives $\text{size}(m, n) = (2k)^{\log_2(m)} k^2 w(n) + \left[(2k)^{\log_2(m)} + (2k)^{\log_2(m)-1} + \cdots + 1 \right] c(k)$. Since $v(n), m(n) \in \text{poly}(n)$ and since k is constant in n, we have $\text{size}(m(n), n) \in \text{poly}(n)$. We can thus also compute the trace of $M_{n,1} \cdots M_{n,m(n)}$ with a $\text{poly}(n)$-size formula. This shows that $(f_n) \in \mathbf{VP}_e$. We thus have $\mathbf{VP}_k \subseteq \mathbf{VP}_e$. ◄

8   Poly-approximation in width-2 ABPs

We give the interpolation argument that completes the proof of Corollary 3.9, which says that the poly-approximation closure of $\mathbf{VP}_2$ equals $\mathbf{VP}_e$ when $\text{char}(\mathbb{F}) \neq 2$ and $\mathbb{F}$ is infinite.

▶ **Proposition 8.1.** $\overline{\mathbf{VP}_e}^{\text{poly}} = \mathbf{VP}_e$ when $\text{char}(\mathbb{F}) \neq 2$ and $\mathbb{F}$ is infinite.

Proof. The inclusion $\mathbf{VP}_e \subseteq \overline{\mathbf{VP}_e}^{\text{poly}}$ is clear. For the other direction, let $(f_n) \in \overline{\mathbf{VP}_e}^{\text{poly}}$. Then there are polynomials $f_{n;i}(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$, $e(n) \in \text{poly}(n)$ such that

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

is computed by a poly-size formula Γ over $\mathbb{F}(\varepsilon)$. Let $\alpha_0, \alpha_1, \ldots, \alpha_{e(n)}$ be distinct elements in $\mathbb{F}$ such that replacing ε by α_j in Γ is a valid substitution (these α_j exist since by assumption our field is infinite). View

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

as a polynomial in ε. The polynomial $g_n(\varepsilon)$ has degree at most $e(n)$ so we can write $g_n(\varepsilon)$ as follows (Lagrange interpolation on $e(n) + 1$ points)

$$g_n(\varepsilon) = \sum_{j=0}^{e(n)} g_n(\alpha_j) \prod_{\substack{0 \leq m \leq e(n): \\ m \neq j}} \frac{\varepsilon - \alpha_m}{\alpha_j - \alpha_m}. \tag{9}$$

Clearly, $f_n(\mathbf{x}) = g_n(0)$. From (9) we see directly how to write $g_n(0)$ as a linear combination of the values $g_n(\alpha_j)$, namely

$$g_n(0) = \sum_{j=0}^{e(n)} g_n(\alpha_j) \prod_{\substack{0 \leq m \leq e(n): \\ m \neq j}} \frac{-\alpha_m}{\alpha_j - \alpha_m},$$

that is,

$$g_n(0) = \sum_{j=0}^{e(n)} \beta_j \, g_n(\alpha_j) \quad \text{with} \quad \beta_j := \prod_{\substack{0 \leq m \leq e(n): \\ m \neq j}} \frac{\alpha_m}{\alpha_m - \alpha_j}.$$

The value $g_n(\alpha_j)$ is computed by the formula Γ with ε replaced by α_j, which we denote by $\Gamma|_{\varepsilon=\alpha_j}$. Thus $f_n(\mathbf{x})$ is computed by the poly-size formula $\sum_{j=0}^{e(n)} \beta_j \, \Gamma|_{\varepsilon=\alpha_j}$. Therefore we have $(f_n) \in \mathbf{VP_e}$. ◀

▶ **Remark 8.2.** Proposition 8.1 also holds with $\mathbf{VP_e}$ replaced by $\mathbf{VP_s}$ or $\mathbf{VP}$ by a similar proof.

9 $\mathbf{VNP_1} \subsetneq \mathbf{VNP}$ when $\mathbb{F} = \mathbb{F_2}$

In our proofs of $\mathbf{VNP_1} = \mathbf{VNP}$ (Section 4 and Section 6) 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 9.1.** $\mathbf{VNP_1} \subsetneq \mathbf{VNP}$ *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

$$1 + xy = \sum_{\mathbf{b}} L_{\mathbf{b}} \quad \text{with} \quad L_{\mathbf{b}} := \prod_{i=1}^{\alpha}(x + A_i) \prod_{j=1}^{\beta}(y + B_j) \prod_{k=1}^{\gamma}(x + y + C_k) \tag{10}$$

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

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

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

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

Multiplying the three expressions and simplifying powers of x and y gives

$$L_{\mathbf{b}} = \prod_{i,j,k} A_i B_j C_k + x\Big(\prod_{i,j,k}(1 + A_i)B_j(1 + C_k) + \prod_{i,j,k} A_i B_j C_k\Big)$$

$$+ y\Big(\prod_{i,j,k} A_i(1 + B_j)(1 + C_k) + \prod_{i,j,k} A_i B_j C_k\Big)$$

$$+ xy\Big(\prod_{i,j,k} A_i(1 + B_j)(1 + C_k) + \prod_{i,j,k}(1 + A_i)B_j(1 + C_k)$$

$$+ \prod_{i,j,k}(1 + A_i)(1 + B_j)C_k + \prod_{i,j,k} A_i B_j C_k\Big).$$

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

$$\sum_{\mathbf{b}} \prod_{i,j,k} A_i B_j C_k = 1, \tag{11}$$

$$\sum_{\mathbf{b}} \prod_{i,j,k} (1 + A_i) B_j (1 + C_k) = 1, \tag{12}$$

$$\sum_{\mathbf{b}} \prod_{i,j,k} A_i (1 + B_j)(1 + C_k) = 1, \tag{13}$$

$$\sum_{\mathbf{b}} \prod_{i,j,k} (1 + A_i)(1 + B_j) C_k = 0. \tag{14}$$

We will show that the above system of equations is inconsistent. Note that (11) asserts that an odd number of vectors $\mathbf{b}$ satisfy the system of equations

$$A_i = 1 \;\; \forall i$$
$$B_j = 1 \;\; \forall j$$
$$C_k = 1 \;\; \forall k.$$

Recall that we defined α, β, γ as the number of factors $x + A_i$, $y + B_j$, $x + y + C_k$ in (10), 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 $0 \le \ell \le n$. 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 (11) if $M(b) = v_0 + 1^\alpha 1^\beta 1^\gamma$, where $1^\alpha 1^\beta 1^\gamma$ is the all-ones vector. Observe that (11) says that there is an odd number of solutions of (11). Since the set of solutions of (11) 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 (11). Let $b^{(1)}$ be this unique solution: $M(b^{(1)}) = v_0 + 1^\alpha 1^\beta 1^\gamma$. We do the same for (12) and (13) 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 (14) asserts that the number of solutions of (14) is even. One solution of (14) 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 (14) 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.** Our proof of Proposition 9.1 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 $1 + xy$.

───── **References** ─────

1   Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 67–75, 2008. `doi:10.1109/FOCS.2008.32`.

2   Eric Allender and Fengming Wang. On the power of algebraic branching programs of width two. *Comput. Complexity*, 25(1):217–253, 2016. `doi:10.1007/s00037-015-0114-7`.

3   Jarod Alper, Tristram Bogart, and Mauricio Velasco. A lower bound for the determinantal complexity of a hypersurface. *Found. Comput. Math.*, pages 1–8, 2015. `arXiv:1505.02205`, `doi:10.1007/s10208-015-9300-x`.

4   Alexander E. Andreev. A method for obtaining more than quadratic effective lower estimates of complexity of π schemes. *Moscow Univ. Math. Bull.*, 42(1):63–66, 1987.

5 Daniel J. Bates and Luke Oeding. Toward a salmon conjecture. *Exp. Math.*, 20(3):358–370, 2011. `doi:10.1080/10586458.2011.576539`.

6 Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992. `doi:10.1137/0221006`.

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

8 Richard P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21(2):201–206, April 1974. `doi:10.1145/321812.321815`.

9 Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*, volume 7 of *Algorithms Comput. Math.* Springer-Verlag, Berlin, 2000. `doi:10.1007/978-3-662-04179-6`.

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

11 Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren Math. Wiss.* Springer-Verlag, Berlin, 1997. `doi:10.1007/978-3-662-03338-8`.

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

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

14 Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. *Proceedings IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 386–395, 2016. `doi:10.1109/FOCS.2016.49`.

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

16 Klim Efremenko, Joseph M. Landsberg, Hal Schenck, and Jerzy Weyman. The method of shifted partial derivatives cannot separate the permanent from the determinant. *ArXiv*, 2016. `arXiv:1609.02103`.

17 Michael Forbes. Some concrete questions on the border complexity of polynomials. Presentation given at the Workshop on Algebraic Complexity Theory WACT 2016 in Tel Aviv, `https://www.youtube.com/watch?v=1HMogQIHT6Q`, 2016.

18 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving algebraic circuits lower bounds. *ArXiv*, 2017. `arXiv:1701.05328`.

19 Fulvio Gesmundo. Geometric aspects of iterated matrix multiplication. *J. Algebra*, 461:42–64, 2016. `arXiv:1512.00766`, `doi:10.1016/j.jalgebra.2016.04.028`.

20 Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Comput. Complexity*, 24(2):393–475, 2015. `doi:10.1007/s00037-015-0103-x`.

21 Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. Boundaries of VP and VNP. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 34:1–34:14, 2016. `arXiv:1605.02815`, `doi:10.4230/LIPIcs.ICALP.2016.34`.

22 Yonghui Guan. Brill's equations as a GL(V)-module. *ArXiv*, 2015. `arXiv:1508.02293`.

23 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. In *2013 IEEE Conference on Computational Complexity – CCC 2013*, pages 65–73. IEEE Computer Soc., Los Alamitos, CA, 2013. `doi:10.1109/CCC.2013.16`.

24 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: a chasm at depth three. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science – FOCS 2013*, pages 578–587. IEEE Computer Soc., Los Alamitos, CA, 2013. `doi:10.1109/FOCS.2013.68`.

25 Johan Håstad. The shrinkage exponent of De Morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998. `doi:10.1137/S0097539794261556`.

26 Jonathan D. Hauenstein, Christian Ikenmeyer, and Joseph M. Landsberg. Equations for lower bounds on border rank. *Exp. Math.*, 22(4):372–383, 2013. `doi:10.1080/10586458.2013.825892`.

27 Pavel Hrubeš and Iddo Tzameret. Short proofs for the determinant identities. *SIAM J. Comput.*, 44(2):340–383, 2015. `doi:10.1137/130917788`.

28 Russell Impagliazzo and Noam Nisan. The effect of random restrictions on formula size. *Random Structures Algorithms*, 4(2):121–133, 1993. `doi:10.1002/rsa.3240040202`.

29 K. A. Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. `doi:10.1137/0214050`.

30 Pascal Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Theoret. Comput. Sci.*, 448:56–65, 2012. `doi:10.1016/j.tcs.2012.03.041`.

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

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

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

34 Joseph M. Landsberg and Giorgio Ottaviani. New lower bounds for the border rank of matrix multiplication. *Theory Comput.*, 11:285–298, 2015. `arXiv:1112.6007`, `doi:10.4086/toc.2015.v011a011`.

35 Thomas Lickteig. A note on border rank. *Inf. Process. Lett.*, 18(3):173–178, 1984. `doi:10.1016/0020-0190(84)90023-1`.

36 Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. *J. Complexity*, 24(1):16–38, 2008. `doi:10.1016/j.jco.2006.09.006`.

37 Thierry Mignon and Nicolas Ressayre. A quadratic bound for the determinant and permanent problem. *Int. Math. Res. Not.*, 2004(79):4241–4253, 2004. `doi:10.1155/S1073792804142566`.

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

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

40 Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 410–418. ACM, 1991. `doi:10.1145/103418.103462`.

41 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complexity*, 6(3):217–234, 1996/97. `doi:10.1007/BF01294256`.

42 Luke Oeding and Steven V. Sam. Equations for the fifth secant variety of segre products of projective spaces. *Exp. Math.*, 25(1):94–99, 2016. `doi:10.1080/10586458.2015.1037872`.

43 Michael S. Paterson and Uri Zwick. Shrinkage of De Morgan formulae under restriction. *Random Structures Algorithms*, 4(2):135–150, 1993. `doi:10.1002/rsa.3240040203`.

44 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):Art. 8, 17, 2009. `doi:10.1145/1502793.1502797`.

45 Herbert John Ryser. *Combinatorial mathematics*. The Carus Mathematical Monographs, No. 14. Published by The Mathematical Association of America; distributed by John Wiley and Sons, Inc., New York, 1963.

46 Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. The power of depth 2 circuits over algebras. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 4, pages 371–382, 2009. `arXiv:0904.2058`, `doi:10.4230/LIPIcs.FSTTCS.2009.2333`.

47 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity 3.0.2, 2016. Online survey, `https://github.com/dasarpmar/lowerbounds-survey`. URL: `https://github.com/dasarpmar/lowerbounds-survey`.

48 Volker Strassen. Rank and optimal computation of generic tensors. *Linear Algebra Appl.*, 52/53:645–685, 1983. `doi:10.1016/0024-3795(83)80041-X`.

49 Bella Abramovna Subbotovskaya. Realizations of linear functions by formulas using $+,\cdot,-$. *Doklady Akademii Nauk SSSR*, 136(3):553–555, 1961.

50 Avishay Tal. Shrinkage of De Morgan formulae by spectral techniques. In *55th Annual IEEE Symposium on Foundations of Computer Science – FOCS*, pages 551–560. IEEE Computer Soc., Los Alamitos, CA, 2014. `doi:10.1109/FOCS.2014.65`.

51 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inform. and Comput.*, 240:2–11, 2015. `doi:10.1016/j.ic.2014.09.004`.

52 Seinosuke Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Trans. Inf. & Syst.*, 75(1):116–124, 1992.

53 Leslie G. Valiant. Completeness classes in algebra. In *Conference Record of the Eleventh Annual ACM Symposium on Theory of Computing (Atlanta, Ga., 1979)*, pages 249–261. ACM, New York, 1979. `doi:10.1145/800135.804419`.

54 Leslie G. Valiant. *Reducibility by algebraic projections*. University of Edinburgh, Department of Computer Science, 1980. Internal Report.

55 Jeroen Zuiddam. A note on the gap between rank and border rank. *Linear Algebra Appl.*, 525:33–44, 2017. `arXiv:1504.05597`, `doi:10.1016/j.laa.2017.03.015`.

A   Overview figure

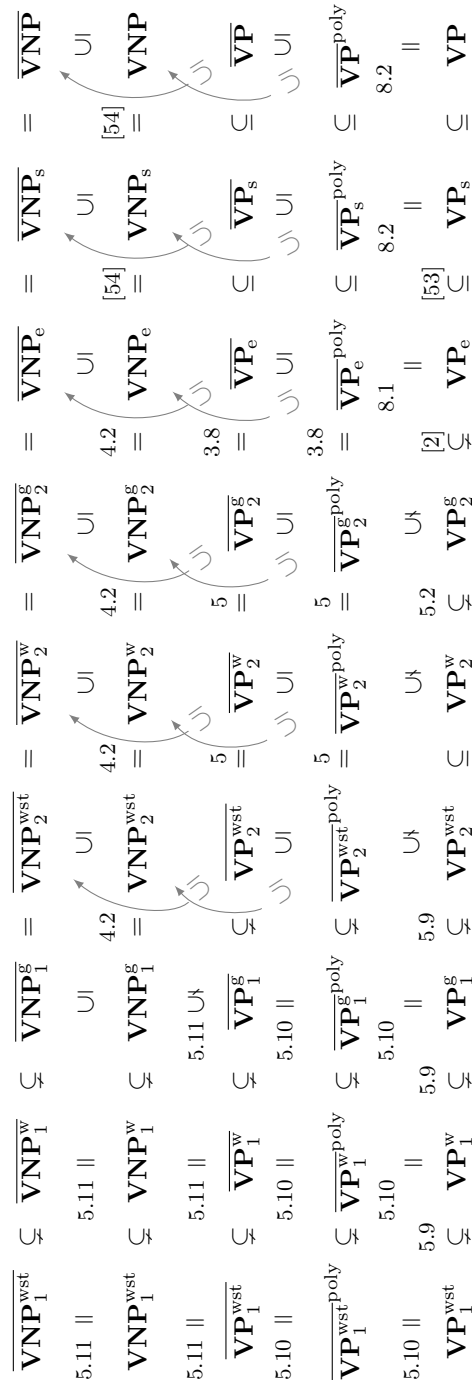The diagram in Fig. A gives an overview of inclusions and separations of complexity classes.

Figure A Overview of inclusions and separations among $\mathbf{VP}_k^*$, $\mathbf{VP}_e$, $\mathbf{VP}_s$, $\mathbf{VP}_e$ and their closures when $\mathrm{char}(\mathbb{F}) \neq 2$.