Electronic Colloquium on Computational Complexity, Report No. 163 (2014)

Homomorphism Polynomials complete for VP*

Arnaud Durand[1], Meena Mahajan[2], Guillaume Malod[1], Nicolas de Rugy-Altherre [1], and Nitin Saurabh[2]

1 Univ Paris Diderot, Sorbonne Paris Cité, IMJ-PRG, UMR 7586 CNRS, Sorbonne Université, UPMC Univ Paris 06, F-75013, Paris, France.
{durand,malod,nderugy}@math.univ-paris-diderot.fr

2 The Institute of Mathematical Sciences, CIT Campus, Chennai 600113, India.
{meena,nitin}@imsc.res.in

Abstract

The VP versus VNP question, introduced by Valiant, is probably the most important open question in algebraic complexity theory. Thanks to completeness results, a variant of this question, VBP versus VNP, can be succinctly restated as asking whether the permanent of a generic matrix can be written as a determinant of a matrix of polynomially bounded size. Strikingly, this restatement does not mention any notion of computational model. To get a similar restatement for the original and more fundamental question, and also to better understand the class itself, we need a complete polynomial for VP. Ad hoc constructions yielding complete polynomials were known, but not natural examples in the vein of the determinant. We give here several variants of natural complete polynomials for VP, based on the notion of graph homomorphism polynomials.

1998 ACM Subject Classification F.1.1 Models of Computation, F.1.3 Complexity Measures and Classes

Keywords and phrases algebraic complexity, graph homomorphism, polynomials, VP, VNP, completeness

1 Introduction

One of the most important open questions in algebraic complexity theory is to decide whether the classes VP and VNP are distinct. These classes, first defined by Valiant in [13, 12], are the algebraic analogues of the Boolean complexity classes P and NP, and separating them is essential for separating P from NP (at least non-uniformly and assuming the generalised Riemann Hypothesis, over the field $\mathbb{C}$, [3]). Valiant established that the family of polynomials computing the permanent is complete for VNP under a suitable notion of reduction which can be thought of as a very strong form of polynomial-size reduction. The leading open question of VP versus VNP is often phrased as the permanent versus the determinant, as the determinant is complete for VP. However, the hardness of the determinant for VP is under the more powerful quasi-polynomial-size reductions. Under polynomial reductions, the determinant is complete for the possibly smaller class VBP. This naturally raises the question of finding polynomials which are complete for VP under polynomial-size reductions. Ad hoc families of generic polynomials can be constructed that are VP-complete, but, surprisingly, there are no known natural polynomial families that are VP-complete. Since complete problems characterise complexity classes, the existence of natural complete problems lends added legitimacy to the study of a class. The determinant and the permanent make the classes VBP, VNP interesting; analogously, what characterises VP?

* This work was supported by IFCPAR/CEFIPRA Project 4702-1(A).

ISSN 1433-8092

Our results and techniques

In this paper, we provide the first instance of natural families of polynomials that (1) are defined independently of the circuit definition of VP, and (2) are VP-complete. The families we consider are families of homomorphism polynomials. Formal definitions appear in Section 2, but here is a brief description. For graphs G and H, a homomorphism from "source graph" G to "target graph" H is a map from $V(G)$ to $V(H)$ that preserves edges. If G and H are directed, a directed homomorphism must preserve directed edges. Additionally, if the vertices of G and H are coloured, a coloured homomorphism must also preserve colours. Placing distinct variables on the vertices (X variables) and edges (Y variables) of the target graph H, we can associate with each homomorphism from G to H a monomial built using these variables. The homomorphism polynomial associated with G and H is the sum of all such monomials. Various variants can be obtained by (1) summing only over homomorphisms of a certain type $\mathcal{H}$, e.g., directed, coloured, injective,... (2) setting non-negative weights α on the vertices of G and using these weights while defining the monomial associated with a homomorphism. Thus the general form of a homomorphism polynomial is $f_{G,H,\alpha,\mathcal{H}}(X,Y)$. We show that over fields of characteristic zero, with respect to constant-depth oracle reductions, the following natural settings, in order of increasing generality, give rise to VP-complete families (Theorem 20):

1. G is a balanced alternately-binary-unary tree with n leaves, with a marker gadget added to the root, and with edge directions chosen in a specific way; H is the complete directed graph on n^6 nodes; α is 1 everywhere; $\mathcal{H}$ is the set of directed homomorphisms.
2. G is an undirected balanced alternately-binary-unary tree with n leaves; H is the complete undirected graph on n^6 nodes; α is 1 everywhere; the vertices are coloured with 5 colours in a specific way; $\mathcal{H}$ is the set of coloured homomorphisms.
3. G is a balanced binary tree with n leaves; H is a complete graph on n^6 nodes; α is 1 for every right child in G and 0 elsewhere; $\mathcal{H}$ is the set of all homomorphisms from G to H.

There seems to be a trade-off between the ease of describing the source and target graphs and the use of weights α. The first family above does not use weights (α is 1 everywhere), but G needs a marker gadget on a naturally defined graph. The second family also does not use weights (α is 1 everywhere), but the colouring of H is described with reference to previously known universal circuits. The third family has very natural source and target graphs, but requires non-trivial α. Ideally, we should be able to show VP-completeness with G and H as in the third family and with trivial weights as in the first two families; our hardness proofs fall short of this. Note however that the weights we use are 0-1 valued. Such 0-1 weights are commonly used in the literature, see, e.g., [2].

A crucial ingredient in our hardness proofs is the fact that VP circuits can be depth-reduced [14] and made multiplicatively disjoint [8] so that all parse trees are isomorphic to balanced binary trees. Another crucial ingredient is that homogeneous components of a polynomial p can be computed in constant depth and polynomial size with oracle gates for p. The hardness proofs illustrate how the monomials in the generic VP-complete polynomial can be put in correspondence with a carefully chosen homogeneous component of the homomorphism polynomial (equivalently, with monomials associated with homomorphisms and satisfying some degree constraints in certain variables). Extracting the homogeneous component is what necessitates an oracle-reduction (constant depth suffices) for hardness. The coloured homomorphism polynomial is however hard even with respect to projections, the stricter form of polynomial-size reductions which is more common in this setting.

For all the above families, membership in VP is shown in a uniform way by showing that a more general homomorphism polynomial, where we additionally have a set of variables Z for each pair of nodes $V(G) \times V(H)$, is in VP, and that the above variants can be obtained from this general polynomial through projections. The generalisation allows us to partition the terms corresponding to $\mathcal{H}$ into groups based on where the root of G is mapped, factorise the sums within each group, and recurse. A crucial ingredient here is the powerful Baur-Strassen Lemma 3 ([1]) which says that for a polynomial p computed by a size s circuit, p and all its first-order derivatives can be simultaneously computed in size $O(s)$.

We also show that when G is a cycle or a path (instead of a balanced binary tree), the homomorphism polynomial family is complete for VBP. Depending on whether G is directed or undirected, we get completeness under projections or c-reductions. On the other hand, using the generalised version with Z variables, and letting G, H be complete graphs, we get completeness for VNP.

Previous related results

As mentioned earlier, very little was previously known about VP-completeness. In [3], Bürgisser showed that a generic polynomial family constructed recursively while controlling the degree is complete for VP. (Bürgisser showed something even more general; completeness for relativised VP.) The construction directly follows a topological sort of a generic VP circuit. In [10] (see also [11]), Raz used the depth-reduction of [14] to show that a family of "universal circuits" is VP-complete; any VP computation can be embedded into it by appropriately setting the variables. Both these VP-complete families are thus directly obtained using the circuit definition / characterization of VP. In [9], Mengel described a way of associating polynomials with constraint satisfaction programs CSPs, and showed that for CSPs where all constraints are binary and the underlying constraint graph is a tree, these polynomials are in VP. Further, for each VP-polynomial, there is such a CSP giving rise to the same polynomial. This means that for the CSP corresponding to the generic VP polynomial or universal circuit, the associated polynomial is VP-complete. The unsatisfactory element here is that to describe the complete polynomial, one again has to fall back to the circuit definition of VP. Similarly, in [4], it is shown that tensor formulas can be computed in VP and can compute all polynomials in VP. Again, to put our hands on a specific VP-complete tensor formula, we need to fall back to the circuit characterisation of VP.

For VBP, on the other hand, there are natural known complete problems, most notably the determinant and iterated matrix multiplication.

A somewhat different homomorphism polynomial was studied in [5]; for a graph H, the monomials of the polynomial f_n^H encode the distinct graphs of size n that are homomorphic to H. The dichotomy result established there gives completeness for VNP or membership in Valiant's analogue of AC^0; it does not capture VP.

Finally, a considerable number of works have been done during the last years on the related subject of counting graph homomorphisms (but mostly in the non uniform settings — i.e., when the target graph is fixed — see [7]) or counting models of CSP and conjunctive queries with connections to VP-completeness (see [6]).

Organization of this paper

In Section 2, basic definitions and notations and previous results used are stated. In Section 3 we describe the hardness of various homomorphism polynomials for VP. Membership in VP is established in Section 4. Completeness for VBP and VNP is discussed in Section 5.

2 Preliminaries and Notation

An arithmetic circuit is a directed acyclic graph with leaves labeled by variables or field elements, internal nodes (called gates) labeled by one of the field operations $+$ and $\times$, and designated output gates at which specific polynomials are computed in the obvious way. If every node has fan-out at most 1 (only one successor), then the circuit is a formula (the underlying graph is a tree). If at every node labeled $\times$, the subcircuits rooted at the children of the node are disjoint, then the circuit is said to be multiplicatively disjoint. For more details about arithmetic circuits, see for instance [11].

A family of polynomials $\{f_n(x_1, \ldots, x_{t(n)})\}$ is p-bounded if f_n has degree $d(n)$, and both $t(n)$, $d(n)$ are $n^{O(1)}$. A p-bounded family $\{f_n\}$ is in VP if a circuit family $\{C_n\}$ of size $s(n) \in n^{O(1)}$ computes it.

▶ **Proposition 1** ([14, 8]). *If $\{f_n\}$ is in* VP*, then $\{f_n\}$ can be computed by polynomial-size circuits of depth $O(\log n)$ where each $\times$ gate has fan-in at most 2. Furthermore, the circuits are multiplicatively disjoint.*

We say that $\{f_n\}$ is a p-projection of $\{g_n\}$ if there is an $m(n) \in n^{O(1)}$ such that each f_n can be obtained from $g_{m(n)}$ by setting each of the variables in $g_{m(n)}$ to a variable of f_n or to a field element.

A **constant-depth c-reduction** from $\{f_n\}$ to $\{g_n\}$, denoted $f \leq_c g$, is a polynomial-size constant-depth circuit family with $+$ and $\times$ gates and oracle gates for g, that computes f. (This is akin to AC^0-Turing reductions in the Boolean world.)

A family $\{D_n\}$ of **universal circuits** computing a polynomial family $\{p_n\}$ is described in [10, 11]. These circuits are universal in the sense that that every polynomial $f(X_1, \ldots, X_n)$ of degree d, computed by a circuit of size s, can be computed by a circuit Ψ such that the underlying graph of Ψ is the same as the graph of D_m, for $m \in \mathrm{poly}(n, s, d)$. (In fact, f_n can be obtained as a projection of p_m.) With minor modifications to $\{D_n\}$ (simple padding with dummy gates, followed by the multiplicative disjointness transformation from [8]), we can show that there is a universal circuit family $\{C_n\}$ in the normal form described below:

▶ **Definition 2** (Normal Form Universal Circuits). A universal circuit $\{C_n\}$ in normal form is a circuit with the following structure:

- It is a layered and semi-unbounded circuit, where $\times$ gates have fan-in 2, whereas $+$ gates are unbounded.
- Gates are alternating, namely every child of a $\times$ gate is a $+$ gate and vice versa. Without loss of generality, the root is a $\times$ gate.
- All the input gates have fan-out 1 and they are at the same level, i.e., all paths from the root of the circuit to an input gate have the same length.
- C_n is a multiplicatively disjoint circuit.
- Input gates are labeled by distinct variables. In particular, there are no input gates labeled by a constant.
- Depth $(C_n) = 2k(n) = 2c\lceil \log n \rceil$; number of variables $(\bar{x}) = v_n$; and size $(C_n) = s_n$, which is polynomial in n.
- The degree of the polynomial computed by the universal circuit is n.

We will identify the directed graph of the circuit, where each edge e is labeled by a new variable X_e, by the circuit itself. Let $(\mathsf{f}_{C_n}(\bar{x}))_n$ be the polynomial family computed by the universal circuit family in normal form.

The Baur-Strassen Lemma says that first-order derivatives can be simultaneously computed efficiently:

▶ **Lemma 3** ([1]). *Let $L(p_1, p_2, \ldots, p_k)$ denote the size of a smallest circuit computing the polynomials p_i at k of its nodes. For any $f \in \mathbb{F}[\bar{x}]$,*

$$L\left(f, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right) \leq 3L(f).$$

The coefficient of a particular monomial in a polynomial can be extracted as described by the following lemma. It appears to be folklore, and was pointed out in [3]; a version appears in Lemma 2 of [5].

▶ **Lemma 4** (Folklore). *Let F be any field of characteristic zero.*

1. *Let p be a polynomial in $F(\bar{W})$, with total degree at most D. Let m be any monomial, with k distinct variables appearing in it. The coefficient of m in p can be computed by a $O(k)$-depth circuit of size $O(Dk)$ with oracle gates for p.*
2. *Let p be a polynomial in $F(\bar{X}, \bar{W})$, with $|\bar{W}| = n$ and total degree in $\bar{W}$ at most D. Let p_d denote the component of p of total degree in $\bar{W}$ exactly d. Then p_d can be computed by a constant depth circuit of size $O(Dn)$ with $O(D)$ oracle gates for p.*

We use (u, v) to denote an undirected edge between u and v, and $\langle u, v\rangle$ to denote a directed edge from u to v.

▶ **Definition 5** (Homomorphisms). Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be two undirected graphs. A homomorphism from G to H is a mapping $\phi : V(G) \to V(H)$ such that the image of an edge is an edge; i.e., for all $(u, v) \in E(G)$, $(\phi(u), \phi(v)) \in E(H)$.

If G, H are directed graphs, then a homomorphism only needs to satisfy for all $\langle u, v\rangle \in E(G)$, at least one of $\langle\phi(u), \phi(v)\rangle, \langle\phi(v), \phi(u)\rangle$ is in $E(H)$. But a directed homomorphism must satisfy for all $\langle u, v\rangle \in E(G)$, $\langle\phi(u), \phi(v)\rangle \in E(H)$.

If c_G, c_H are functions assigning colours to $V(G)$ and $V(H)$, then a coloured homomorphism must also satisfy, for all $u \in V(G)$, $c_G(u) = c_H(\phi(u))$.

▶ **Definition 6** (Homomorphism polynomials (see, e.g., [2])). Let G and H be undirected graphs; the definitions for the directed case are analogous. Consider the set of variables $X \cup Y$ where $X = \{X_u | u \in V(H)\}$ and $Y = \{Y_{uv} | (u, v) \in E(H)\}$. Let $\alpha : V(G) \mapsto \mathbb{N}$ be a labeling of vertices of G by non-negative integers. For each homomorphism ϕ from G to H we associate the monomial

$$mon(\phi) \triangleq \left(\prod_{u \in V(G)} X_{\phi(u)}^{\alpha(u)}\right)\left(\prod_{(u,v) \in E(G)} Y_{\phi(u),\phi(v)}\right)$$

Let $\mathcal{H}$ be a set of homomorphisms from G to H. The homomorphism polynomial $f_{G,H,\alpha,\mathcal{H}}$ is defined as follows:

$$f_{G,H,\alpha,\mathcal{H}}(X, Y) = \sum_{\phi \in \mathcal{H}} mon(\phi) = \sum_{\phi \in \mathcal{H}} \left(\prod_{u \in V(G)} X_{\phi(u)}^{\alpha(u)}\right)\left(\prod_{(u,v) \in E(G)} Y_{\phi(u),\phi(v)}\right)$$

Some sets of homomorphisms we consider are **InjDirHom**: injective directed homomorphisms, **InjHom**: injective homomorphisms, **DirHom**: directed homomorphisms, **ColHom**: coloured homomorphisms, **Hom**: all homomorphisms.

▶ **Definition 7** (Parse trees (see, e.g., [8])). The set of parse trees of a circuit C is defined by induction on its size:

- If C is of size 1, it has only one parse tree, itself.

- If the output gate of C is a $\times$ gate whose children are the gates α and β, the parse trees of C are obtained by taking a parse tree of C_α, a parse tree of a disjoint copy of C_β and the edges from α and β to the output gate.
- If the output of C is a $+$ gate, the parse trees of C are obtained by taking a parse tree of a subcircuit rooted at one of the children and the edge from the (chosen) child to the output gate.

Each parse tree T is associated with a monomial by computing the product of the values of the input gates. We denote this value by $mon(\mathsf{T})$.

▶ **Lemma 8** ([8]). *If C is a circuit computing a polynomial f, then $f(\bar{x}) = \sum_{\mathsf{T}} \mathrm{mon}(\mathsf{T})$, where the sum is over the set of parse trees, T, of C.*

▶ **Proposition 9** ([8]). *A circuit C is multiplicatively disjoint if and only if any parse tree of C is a subgraph of C. Furthermore, a subgraph T of C is a parse tree if the following conditions are met:*
- *T contains the output gate of C.*
- *If α is a multiplication gate in T having gates β and γ as children in C, then the edges $\langle\beta,\alpha\rangle$ and $\langle\gamma,\alpha\rangle$ also appear in T.*
- *If α is an addition gate in T, it has only one child in T.*
- *Only edges and gates obtained in this way belong to T.*

3 Lower Bounds: VP-hardness

Here we study the question of whether all families of polynomials in VP can be computed by homomorphism polynomials. Instantiating G, H and α to our liking we obtain a variety of homomorphism polynomials that are VP-hard. We describe them in increasing order of generalisation.

▶ **Definition 10.** Let AT_k be a directed balanced alternately-binary-unary tree with k leaves. Vertices on an odd layer have exactly two incoming edges whereas vertices on an even layer have exactly one incoming edge. The first layer has only one vertex called root, and the edges are directed from leaves towards the root.

▶ **Lemma 11.** *The parse trees of C_n, the universal circuit in normal form, are subgraphs of C_n and are isomorphic to AT_n.*

This observation suggests a way to capture monomial computations of the universal circuit via homomorphisms from AT_k into C_n.

Injective Directed Homomorphism

▶ **Proposition 12.** *Consider the homomorphism polynomial where*
- *$G := \mathsf{AT}_m$.*
- *H is the directed graph corresponding to the universal circuit in normal form C_m.*
- *$\mathcal{H} :=$ set of injective directed homomorphisms from G to H.*
- *α is 1 everywhere.*

Then, the family $(f_{\mathsf{AT}_m,H,\alpha,\textbf{InjDirHom}}(\bar{X},\bar{Y}))_m$, where $m \in \mathbb{N}$, is VP-hard for projections.

We want to express the universal polynomial through a projection. The idea is to show that elements in **InjDirHom** are in bijection with parse trees of C_m, and compute the same monomials.

Proof. We claim $(\mathsf{f}_{C_n}(\bar{x}))_n \leq_p (f_{\mathsf{AT}_m,H,\alpha,\mathbf{InjDirHom}}(\bar{X},\bar{Y}))_m$. To prove our claim it suffices to show that $\mathsf{f}_{C_m}(\bar{x}) \leq f_{\mathsf{AT}_m,H,\alpha,\mathbf{InjDirHom}}(\bar{X},\bar{Y})$. Let $m = 2^{k(n)}$.

The $\bar{Y}$ variables are all set to 1. The $\bar{X}$ variables that correspond to input gates of C_m are set to corresponding values (in $\bar{x}$) of the input gates, otherwise they are set to 1.

By Lemma 8 and the definition of $f_{\mathsf{AT}_m,H,\alpha,\mathbf{InjDirHom}}$, it suffices to show that $\sum_{\phi\in\mathcal{H}} mon(\phi) = \sum_{\mathsf{T}} mon(\mathsf{T})$, where T is a parse tree of C_m.

Let us consider an injective directed homomorphism ϕ such that $\phi(\mathsf{AT}_{2^{k(n)}})$ is a parse tree of C_m. It is easy to observe that $mon(\phi) = mon(\phi(\mathsf{AT}_{2^{k(n)}}))$. Therefore, to complete the proof, it suffices to show that the set $\mathcal{I}$ of images of injective directed homomorphisms from $\mathsf{AT}_{2^{k(n)}}$ to C_m is equal to the set of parse trees of C_m.

Since the homomorphisms are injective and respect direction each element of the set $\mathcal{I}$ is isomorphic to $\mathsf{AT}_{2^{k(n)}}$. Hence, by Lemma 11, the set of parse trees of C_m are contained in $\mathcal{I}$.

We now show that every element of the set $\mathcal{I}$ is a parse tree of C_m. Let $\phi \in \mathbf{InjDirHom}$ and r be the root of $\mathsf{AT}_{2^{k(n)}}$. Let ℓ be a leaf of C_m in $\phi(\mathsf{AT}_{2^{k(n)}})$. As ϕ respects direction, there is a path in $\phi(\mathsf{AT}_{2^{k(n)}})$ of length $2k(n)$ from ℓ to $\phi(r)$. But the only gate in C_m at a distance $2k(n)$ from a leaf is the root of C_m. Therefore the root of $\mathsf{AT}_{2^{k(n)}}$ must be mapped to the root of C_m. Similarly, we can argue that the i-th layer of $\mathsf{AT}_{2^{k(n)}}$ must be mapped to the i-th layer of C_m. Hence, by Proposition 9, every element of the set $\mathcal{I}$ is a parse tree of C_m. ◀

▶ Remark. The hardness proof above will work even if H is the complete directed graph on poly(m) nodes. In the projection, we can set the $\bar{Y}$ variables to values in $\{0,1\}$ such that the edges with variables set to 1 together form the underlying graph of C_n.

If we follow the proof of the previous proposition and look at the image of a given homomorphism in layers, we notice that "direction"-respecting homomorphisms basically ensured that we never fold back (in the image). In particular, the mapping respect layers. Furthermore "injectivity" helped ensure that vertices within a layer are mapped distinctly. This raises an intriguing question: can we eliminate either assumption (*direction* or *injectivity*) and still prove VP-hardness? We answer this question positively, albeit under a stronger notion of reduction.

Injective Homomorphisms

Let AT^u_k be defined as the alternately-binary-unary tree AT_k, but with no directions on edges.

▶ **Proposition 13.** *Consider the homomorphism polynomial where*
- $G := \mathsf{AT}^u_m$.
- *H is a complete graph (undirected) on poly(m), say m^6, nodes.*
- *$\mathcal{H}$:= set of injective homomorphisms from G to H.*
- *α is 1 everywhere.*

Then, the family $(f_{\mathsf{AT}^u_m,H,\alpha,\textbf{\textit{InjHom}}}(\bar{X},\bar{Y}))_m$ is VP*-hard for constant-depth c-reductions.*

Again, we want to express the universal polynomial. To enforce directedness of the injective homomorphisms, we assign a special variable on the edges emerging from the root, and a special variable on edges reaching the leaves. The proof idea is to show that coefficient of certain monomial in f extracts exactly the contribution of injective directed homomorphisms, and this, by Proposition 12, is the universal polynomial. The desired coefficient can be extracted by a constant-depth c-reduction. We now give the proof in detail.

Proof. Let $m = 2^k$. The choice of poly(m), in defining H, is such that $s_n \leq \text{poly}(m)$. $\bar{Y}$ variables take values in $\{0, 1, r, \ell\}$ such that the ones set to non-zero together form the undirected underlying graph of C_n. Y variables corresponding to edges adjacent to the root of C_n are set to 'r'. Y variables corresponding to edges adjacent to an input gate in C_n are set to 'ℓ'. $\bar{X}$ variables (of H) that correspond to input gates in C_n are set to corresponding values (in $\bar{x}$) of the input gates, otherwise they are set to 1.

Let $\mathcal{I}$ be the set of images of injective homomorphisms from AT^u_m to C_n. By Lemma 11, we know that the set of parse trees of C_n is contained in $\mathcal{I}$. Let $\phi \in \mathcal{H}$ be such that $\phi(\mathsf{AT}^u_m)$ is a parse tree of C_n. Observe that in this case $mon(\phi)$ has degree 2^k in ℓ and 2 in r. We now claim that if $mon(\phi)$ has degree 2^k in ℓ and 2 in r, then the image of ϕ is a parse tree.

Since ϕ is injective, only degree 1 vertices of AT^u_m can be mapped to a leaf in C_n. Thus, due to the degree of ℓ in $mon(\phi)$, the 2^k leaves of AT^u_m must be mapped to different input gates in C_n. Also the degree constraint on r and injectivity together suggest that two edges adjacent to the root in C_n are in the image of ϕ. Hence there must be a vertex in AT^u_m that is mapped to the root in C_n. Let us call this vertex v. Note that the shortest distance between two vertices in AT^u_m is at least as large as the shortest distance between their homomorphic image in C_n. Hence v is a vertex of AT^u_m such that every leaf in AT^u_m is at least a distance of $2k$ from v. But this is true of only one vertex in AT^u_m, and that is the root of AT^u_m. Therefore the image of an injective homomorphism such that its monomial has degree 2^k in ℓ and 2 in r is a parse tree of C_n.

Now to compute the universal polynomial we do an interpolation over the oracle polynomial to extract the coefficient of $\ell^{2^k} r^2$, as described in Lemma 4. ◀

Directed Homomorphisms

Consider the directed alternately-binary-unary-tree AT_k. For every vertex in an odd layer there are two incoming edges. Flip the direction of the right edge for every such vertex. Note that the edges coming into the unary vertices at even layers are unchanged. Also connect a path $t_1 \to t_2 \to \cdots \to t_s$ to the root by adding an edge $\langle t_s, root\rangle$. The vertices $t_1, \ldots, t_s$ are new vertices. Denote this modified alternately-binary-unary-tree by $\mathsf{AT}^d_{k,s}$.

▶ **Theorem 14.** *Consider the homomorphism polynomial where*

- $G := \mathsf{AT}^d_{m,s}$ *for sufficiently large s in poly(m), say $s = m^7$.*
- *H is a complete directed graph on poly(m), say m^6, nodes.*
- *$\mathcal{H}$:= set of directed homomorphisms from G to H.*
- *α is 1 everywhere.*

Then, the family $(f_{\mathsf{AT}^d_{m,s}, H, \alpha, \boldsymbol{DirHom}}(\bar{X}, \bar{Y}))_m$ is VP*-hard for constant-depth c-reductions.*

Proof. As before, to compute the universal polynomial we assign special variables on the edges of the graph. The idea is to show that homomorphism monomials with certain degrees in special variables are in bijection with parse trees of C_m (and compute the same corresponding monomials). We use the length of the tail, the degree constraints and multiplicative disjointness of C_m to establish a required bijection. We fill in the details now.

Let us set $m := 2^{k(n)}$ and $s := 2s_n$. The choice of poly(m) is such that $3s_n \leq \text{poly}(m)$. $\bar{Y}$ variables take values in $\{0, 1, t, r, \ell\}$ such that the ones set to non-zero together form the undirected underlying graph of C_n with a path $v_1 \to v_2 \to \cdots \to v_{2s_n} \to$ root, attached to the root of C_n. $Y_{\langle v_1, v_2\rangle}$ is set to t. $Y_{\langle v_{2s_n}, root\rangle}$ is set to r. Y variables corresponding to edges adjacent to an input gate in C_n are set to 'ℓ'. $\bar{X}$ variables (of H) that correspond to input gates in C_n are set to corresponding values (in $\bar{x}$) of the input gates, otherwise they are set to 1.

Let $\phi \in \mathcal{H}$ be such that $\phi(\mathsf{AT}^d_{m,s})$ is a parse tree of C_n. Observe that in this case $mon(\phi)$ has degree $2^{k(n)}$ in ℓ, 1 in r and 1 in t. We claim that if $mon(\phi)$ has degree $2^{k(n)}$ in ℓ, 1 in r and 1 in t, then the image of ϕ is a parse tree of C_n. First, note that any directed homomorphism from $\mathsf{AT}^d_{m,s}$ to C_n with degree 1 in r and 1 in t is well rooted, that is, root of $\mathsf{AT}^d_{m,s}$ is mapped to the root of C_n and the path of length $2s_n$ in $\mathsf{AT}^d_{m,s}$ is mapped isomorphically to a path of length $2s_n$ in C_n. Since it also has degree $2^{k(n)}$ in ℓ, it must be the case that $2^{k(n)}$ leaves of $\mathsf{AT}^d_{m,s}$ (except t_1) are mapped to the leaves of C_n. Hence layers of $\mathsf{AT}^d_{m,s}$ must be mapped to the corresponding layer in C_n. Now to prove that the image of ϕ is a parse tree, it suffices to show that ϕ is injective on each layer of $\mathsf{AT}^d_{m,s}$.

ϕ is injective on the first layer since it has only one vertex, the root. Inductively suppose ϕ is injective until the $(i-1)$-th layer. Now assume that there are two vertices α and β on the i-th layer of $\mathsf{AT}^d_{m,s}$ which are mapped to the same gate on the i-th layer. We argue that this violates the *multiplicative disjointness* of C_n. First notice that two children of a binary vertex γ of $\mathsf{AT}^d_{m,s}$ must be mapped to two distinct vertices, since the edges connecting them have different orientations and there are no 2-cycles in C_n. Let α' and β' be parents of α and β respectively. From the aforementioned observation it follows that α' must be different from β'. Let us consider the smallest common ancestor of $\phi(\alpha')$ and $\phi(\beta')$ in $\phi(\mathsf{AT}^d_{m,s})$. This must be a $\times$ gate, and hence we get a contradiction to the multiplicative disjointness of C_n. Now to compute the universal polynomial, as before, we use Lemma 4 to extract the coefficient of $\ell^{2^{k(n)}}rt$.

◀

Coloured Homomorphisms

In all the above hardness proofs we restricted the set of homomorphisms to be *direction-respecting*, or *injective*, or both. Here we show another restriction, called *colour-respecting*, that gives a VP-hard polynomial. Recall that a homomorphism from a coloured graph to another coloured graph is *colour-respecting* if it preserves the colour class of vertices.

Consider the following colouring of AT^u_k with colours *brown, left, right, white* and *green.* The root of AT^u_k is coloured *brown*, leaves are coloured *green.* For every gate on an even layer, if it is the left (resp. right) child of its parent then colour it *left* (resp. *right*). Every gate on an odd layer, except the root, is coloured *white.* Denote this coloured alternately-binary-unary-tree as AT^c_k.

We define a circuit to be *properly* coloured if the root is coloured *brown*, leaves are coloured *green*, all multiplication gates but the root are coloured *white* and all addition gates are coloured *left* or *right* depending on whether they are left or right child respectively.

We obtain a *properly* coloured circuit from the universal circuit C_n as follows. For all addition gates in C_n we make two coloured copies, one coloured *left* and the other coloured *right.* We add edge connections as follows: for a multiplication gate we add an incoming edge to it from the *left* (resp. *right*) coloured copy of the left (resp. right) child, and for an addition gate the coloured gates are connected as the original gate in the circuit C_n.

We say that an undirected complete graph H on M nodes is *properly* coloured if, for all $s_n \leq M/2$, there is an embedding of the graph that underlies an s_n-sized *properly* coloured universal circuit, into H.

▶ **Theorem 15.** *Consider the homomorphism polynomial where*
- $G := \mathsf{AT}^c_m$.
- *H is a* properly *coloured complete graph (undirected) on poly(m), say m^6, nodes.*
- *$\mathcal{H} :=$ set of coloured homomorphisms from G to H.*

- α *is 1 everywhere.*

Then, the family $(f_{\mathsf{AT}^c_m,H,\alpha,\boldsymbol{ColHom}}(\bar{X},\bar{Y}))_m$*, where* $m \in \mathbb{N}$*, is* VP*-hard for projections.*

Proof. Let us set $m := 2^{k(n)}$. The choice of poly(m) is such that $2s_n \leq$ poly(m). The $\bar{Y}$ variables are set to $\{0,1\}$ such that the ones set to 1 together form the underlying graph of *properly* coloured universal circuit C_n. The $\bar{X}$ variables that correspond to input gates of C_n are set to corresponding values (in $\bar{x}$) of the input gates (irrespective of their colour), otherwise they are set to 1.

Let us consider a coloured homomorphism ϕ such that $\phi(\mathsf{AT}^c_{2^{k(n)}})$ is a parse tree of the properly coloured C_n. It is easy to observe that $mon(\phi) = mon(\phi(\mathsf{AT}^c_{2^{k(n)}}))$. Therefore, to complete the proof it suffices to show that the set $\mathcal{I}$ of images of coloured homomorphisms from $\mathsf{AT}^c_{2^{k(n)}}$ to C_n is equal to the set of parse trees of C_n.

By Lemma 11, we know that the set of parse trees of C_n is contained in $\mathcal{I}$.

We now show that every element of the set $\mathcal{I}$ is a parse tree of C_n. Since ϕ is colour-respecting it maps the root to the root and leaves to leaves. Hence, ϕ also respects layers, that is, i-th layer of AT^c_m is mapped to i-th layer of C_n. Therefore, it suffices to show that ϕ is injective on each layer. This follows from a similar argument as in Theorem 14.

◀

The generic homomorphism polynomial gives us immense freedom in the choice of G, target graph H, weights α and the set of homomorphisms $\mathcal{H}$. Until now we used several modified graphs along with different restrictions on $\mathcal{H}$ to capture computations in the universal circuit. The question here is: can we get rid of restrictions on the set of homomorphisms? We provide a positive answer, using instead weights on the vertices of the source graph.

Homomorphism with weights

For k a power of 2, let T_k denote a complete (perfect) binary tree with k leaves.

▶ **Theorem 16.** *Consider the homomorphism polynomial where*

- $G := \mathsf{T}_m$.
- H *is a complete graph (undirected) on poly(m), say* m^6*, nodes.*
- $\mathcal{H} :=$ *set of all homomorphisms from* G *to* H.
- *Define* α *such that,*

$$\alpha(u) = \begin{cases} 0 & u = \textit{root} \\ 1 & \textit{if } u \textit{ is the right child of it's parent} \\ 0 & \textit{otherwise} \end{cases}$$

Then, the family $(f_{\mathsf{T}_m,H,\alpha,\boldsymbol{Hom}}(\bar{X},\bar{Y}))_m$*, where* $m \in \mathbb{N}$*, is* VP*-hard for constant-depth c-reductions.*

Since the proof is long with several case analysis, we would like to discuss the proof outline before presenting the proof.

Note that the source graphs are complete binary trees. Therefore, we first need to compact parse trees and get rid of the unary nodes (corresponding to + gates). We construct from the universal circuit C_n a graph J_n that allows us to get rid of the alternating binary-unary parse tree structure while maintaining the property that the compacted "parse trees" are subgraphs of J_n. The graph J_n has two copies g_L and g_R of each $\times$ gate and input gate of C_n. It also has two children attached to each leaf node. The edges of J_n essentially shortcut the + edges of C_n.

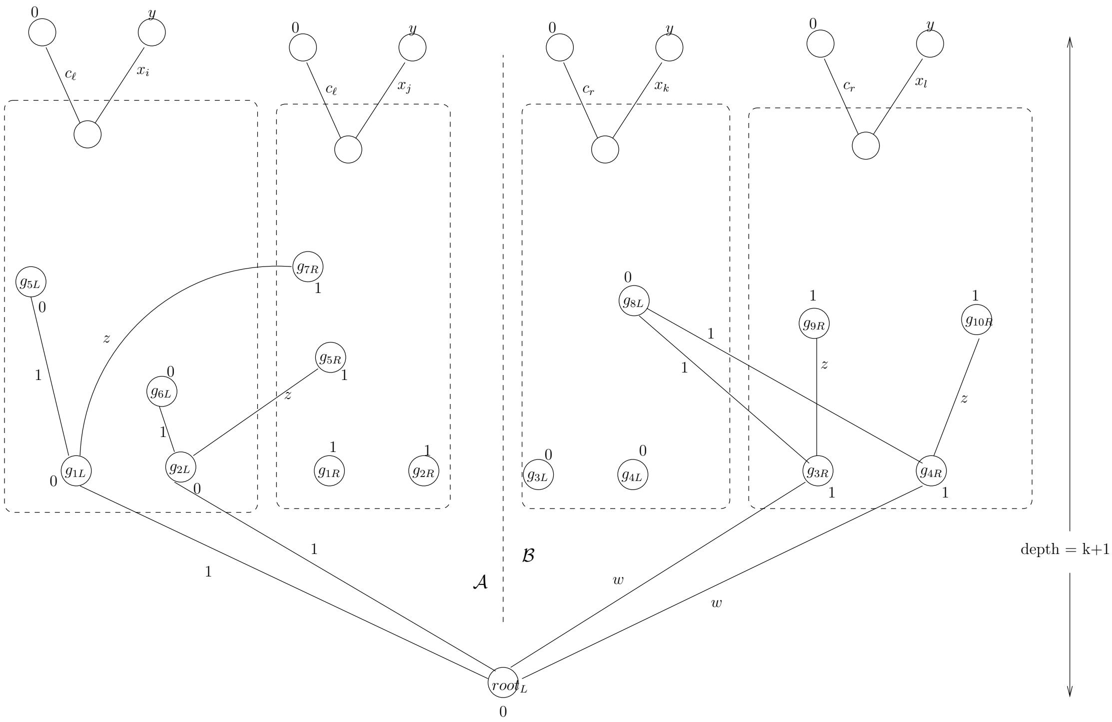

Figure 1 Graph J_n with vertex and edge labels

As before, we use Y variables to pick out J_n from H. We assign special variables w on edges from the root to a node g_R, and z on edges going from a non-root non-input node u to some right copy node g_R. For an input node g in the "left sub-graph" of J_n, the new left and right edges are assigned c_ℓ and x respectively, where x is the corresponding input label of g in C_n, and the node at the end of the x edge is assigned a special variable y. In the right sub-graph, variable c_r is used.

We show that homomorphisms whose monomials have degree 1 in w, $2^k - 2$ in z, 2^{k-1} each in c_ℓ and c_r, and 2^k in y are in bijection with compacted parse trees in J_n. The argument proceeds in stages: first show that the homomorphism is well-rooted (using the degree constraint on w, c_ℓ, c_r and the 0-1 weights in G), then show that it preserves layers (does not fold back) (using the degree constraint on c_ℓ, c_r and y), then show that it is injective within layers (using the degree constraint in z and the 0-1 weights in G).

Proof. Before starting the proof, we set up some notation.

We obtain a sequence of graphs (J_n) from the undirected graphs underlying (C_n). To make the presentation clearer, we first construct an intermediate graph J'_n as follows. Retain the multiplication and input gates of C_n. Let us make two copies of each. For each retained gate, g, in C_n; let g_L and g_R be the two copies of g in J'_n (see Figure 1). We now define the edge connections in J'_n. Assume g is a $\times$ gate retained in J'_n. Let α and β be two $+$ gates feeding into g in C_N. Let $\{\alpha_1, \ldots, \alpha_i\}$ and $\{\beta_1, \ldots, \beta_j\}$ be the gates feeding into α and β, respectively. Assume without loss of generality that α and β feed into g from left and right, respectively. Now we add the following set of edges to J'_N: $\{(\alpha_{1L}, g_L), \ldots, (\alpha_{iL}, g_L)\}$, $\{(\beta_{1R}, g_L), \ldots, (\beta_{jR}, g_L)\}$, $\{(\alpha_{1L}, g_R), \ldots, (\alpha_{iL}, g_R)\}$ and $\{(\beta_{1R}, g_R), \ldots, (\beta_{jR}, g_R)\}$. We now would like to keep a single copy of C_n in these set of edges. So we remove the vertex $root_R$ and we remove the remaining spurious edges in following way. If we assume that all edges are directed from root towards leaves, then we keep only edges induced by the vertices reachable from $root_L$ in this directed graph.

We now transform J'_n as follows to get J_n (see Figure 1): for each gate g' in J'_n which corresponds to an input gate in C_n, we add two new distinct vertices and connect them to g'. Note that there are two type of vertices in J_n; one that corresponds to a gate in C_n and others are degree 1 vertices hanging from gates that correspond to input gates in C_n.

▶ **Observation 17.** *There is a one-to-one correspondence between parse trees of C_n and subgraph of J_n that are rooted at $root_L$ and isomorphic to $\mathsf{T}_{2^{k(n)+1}}$.*

Let us set $m := 2^{k(n)+1}$. The choice of poly(m) is such that $4s_n \leq$ poly(m). The $\bar{Y}$ variables are set to $\{0, 1, w, z, c_\ell, c_r, \bar{x}\}$ such that the ones set to non-zero together form the graph J_n. The $\bar{X}$ variables take values in $\{0, 1, y\}$. The ones corresponding to the left copies of gates in C_n are set to 0, whereas to the right copies are set to 1. $\bar{X}$ variables for degree 1 vertices hanging from input gates are set to 0 or 'y' depending on whether they are left or right child, respectively.

For every edge $(root_L, g_R)$, we set $Y_{(root_L, g_R)} := w$. For all $u \in V(J_n)$, except $root_L$, and degree of u not equal to 1, if the edge (u, g_R) exist then we set $Y_{(u,g_R)} := z$.

Let v be a gate, in J_n, corresponding to an input gate g in C_n and v lies in $\mathcal{A}$ part (see Figure 1). Let v_1 and v_2 be the left and right leaf attached to v, then we set $Y_{vv_1} := c_\ell$ and $Y_{vv_2} :=$ the $\bar{x}$-label of g in C_n.

For v a gate, in J_n, corresponding to an input gate g in C_n and lying in $\mathcal{B}$ part (see Figure 1), let v_1 and v_2 be the left and right leaf attached to v. Then we set $Y_{vv_1} := c_r$ and $Y_{vv_2} :=$ the $\bar{x}$-label of g in C_n.

All other remaining edge variables that are not set to 0, are set to 1.

By Observation 17 we easily deduce that for each parse tree p-T of C_n there exist a homomorphism ϕ from $\mathsf{T}_{2^{k(n)+1}}$ to J_n such that $mon(\phi)$ is equal to $mon(p\text{-}\mathsf{T}) \times wz^{(2^k-2)}c_\ell^{2^{(k-1)}}c_r^{2^{(k-1)}}y^{2^k}$, where $k = k(n)$.

We claim that for a homomorphism ϕ, if $mon(\phi)$ has degree 1 in w, $(2^k - 2)$ in z, 2^{k-1} in c_ℓ, 2^{k-1} in c_r and 2^k in y, then the homomorphic image $\phi(\mathsf{T}_{2^{k(n)+1}})$ is isomorphic to $\mathsf{T}_{2^{k(n)+1}}$ rooted at $root_L$.

We will prove the claim in two parts. First we prove that if any node other than the root of $\mathsf{T}_{2^{k(n)+1}}$ is mapped to $root_L$ then the corresponding monomial do not have right degree in w, c_ℓ or c_r. We then consider the case where the root of the complete binary tree is the only node mapped to $root_L$ under ϕ, and we argue that if ϕ has the required degrees then it must be a complete binary tree with $2^{k(n)+1}$ leaves rooted at $root_L$.

Case 1: $\phi^{-1}(root_L) = \emptyset$. Clearly $mon(\phi)$ has degree *zero* in w.

Case 2: $\phi^{-1}(root_L)$ contains a degree 3 vertex, say v. Let v_1 and v_2 be the left and right child of v, respectively. Let v_3 be the parent of v in T_m. Note that v must be labeled 0 for the monomial to survive. Also, at least one of the v_i's is labeled 1.

Case 2a: Suppose two of the v_i's are labeled 1. Hence for the $mon(\phi)$ to survive these v_i's must be mapped to the right of $root_L$. But then $mon(\phi)$ has degree at least 2 in w.

Case 2b: Exactly one of the v_i is labeled 1. It must be the right child v_2, for the monomial to survive it should be mapped to the right of $root_L$. Now if v_1 or v_3 is also mapped to the right of $root_L$, $mon(\phi)$ will have degree at least 2 in w. Otherwise, both v_1 and v_3 are mapped to the left of $root_L$. Since v_1 is an internal vertex of T_m, the subtree rooted at v_2 and v_1 has depth at most $k - 1$ in T_m. In the first case $mon(\phi)$ does not have sufficient degree in c_ℓ , whereas in the second case it does not have sufficient degree in c_r.

Case 3: $\phi^{-1}(root_L)$ contains the root of T_m and at least one degree 1 vertex, say v. Also, no degree 3 vertices are mapped to $root_L$. As before, the left child of the root of T_m is mapped to the left of $root_L$ and the right child is mapped to the right of $root_L$, else either the monomial evaluates to zero or has degree at least 2 in w.

Case 3a: For some leaf node v mapped to $root_L$, its neighbour is mapped to the right of $root_L$. In this case if the monomial is not zero, we will have at least degree 2 in w.

Case 3b: For all leaf node v mapped to $root_L$, their neighbour is mapped to the left of $root_L$. But now $mon(\phi)$ will not have sufficient degree in c_ℓ.

Case 4: $\phi^{-1}(root_L)$ contains only degree 1 vertices. But then the homomorphic image is confined only to the left side or right side of $root_L$. Hence the monomial will not have sufficient degree in either c_r or c_ℓ.

Therefore, we have shown that to get the appropriate degrees as claimed, $\phi^{-1}(root_L)$ must only contain the root of T_m. Now to complete the proof we will show that if $mon(\phi)$ has correct degrees in w, z, c_ℓ, c_r and y, then ϕ is injective and preserves left-right labelling of nodes of T_m. Note that for the monomial to survive and have degree 1 in w, it must be the case that the right child of the root of T_m is mapped to the right of $root_L$ and the left child is mapped to the left of $root_L$.

We claim that the homomorphism ϕ can not 'fold back' layers, that is, map a descendant to the node where its ancestor is mapped. This is because otherwise the monomial will not have sufficient degree in either c_ℓ, c_r, or y (if folding happens at depth k+1).

We also claim that the homomorphism ϕ can not 'squish' a layer, that is, map two siblings to the same node. If the two are mapped to a vertex labeled 0, the monomial evaluates to zero. In the other case, they are mapped to a vertex labeled 1 but then the two siblings together, either contribute degree 2 in z or miss out at least degree 1 in c's which cannot be compensated further if the monomial is non-zero.

Therefore we have shown that homomorphisms that are injective, whose image is isomorphic to T_m and rooted at $root_L$, and which preserve left-right labels are in one-to-one correspondence with parse trees of C_n.

As before, to compute the universal polynomial we do an interpolation over the oracle polynomial (Lemma 4) to extract the coefficient of $wz^{(2^k-2)}c_\ell^{2^{(k-1)}}c_r^{2^{(k-1)}}y^{2^k}$.

◀

4 Upper Bounds: membership in VP

In this section we will show that most of the variants of the homomorphism polynomial considered in the previous section are also computable by polynomial size arithmetic circuits. That is, the homomorphism polynomials are VP-Complete. For sake of clarity we describe the membership of a generic homomorphism polynomial in VP in detail. Then we explain how to obtain various instantiations via projections.

We define a set of new variables $\bar{Z} := \{Z_{u,a} \mid u \in V(G) \text{ and } a \in V(H)\}$. Let us generalise the homomorphism polynomial $f_{G,H,\alpha,\mathcal{H}}$ as follows:

$$f_{G,H,\mathcal{H}}(\bar{Z},\bar{Y}) = \sum_{\phi\in\mathcal{H}} \left(\prod_{u\in V(G)} Z_{u,\phi(u)} \right) \left(\prod_{(u,v)\in E(G)} Y_{\phi(u),\phi(v)} \right).$$

Note that for a 0-1 valued α, we can easily obtain $f_{G,H,\alpha,\mathcal{H}}$ from our generic homomorphism polynomial $f_{G,H,\mathcal{H}}$ via substitution of $\bar{Z}$ variables, setting $Z_{u,a}$ to $X_a^{\alpha(u)}$. (If α can take any non-negative values, then we can still do the above substitution. We will need subcircuits computing appropriate powers of the $\bar{X}$ variables. The resulting circuit will still be poly-sized and hence in VP, provided the powers are not too large.)

▶ **Theorem 18.** *The family of homomorphism polynomials* $(f_m) = f_{G_m,H_m,\textbf{Hom}}(\bar{Z},\bar{Y})$ *where*
- G_m *is* T_m*, the complete balanced binary tree with* $m = 2^k$ *leaves,*
- H_m *is* K_n*, complete graph on* $n = poly(m)$ *nodes,*

is in VP*.*

Proof. The idea is to group the homomorphisms based on where they send the root of G_m and its children, and to recursively compute sub-polynomials within each group. The sub-polynomials in a specific group will have a specific set of variables in all their monomials. Thus the group can be identified by suitably combining partial derivatives of the recursively constructed sub-polynomials. (Note: this is why we consider the generalised polynomial with $\bar{Z}$ instead of $\bar{X}$ and α. If for some u, $\alpha(u) = 0$, then we cannot use partial derivatives to force sending u to a specific vertex of H.) The partial derivatives themselves can be computed efficiently using Lemma 3.

We show by induction on m that $f = f_m$ can be computed in size $S(m,n) = O(m^3n^3)$. In the base case when $m = 1$, $f = \sum_{a \in V(H)} Z_{u,a}$, and the trivial circuit is of size n.

For $m \geq 2$, let r, r_1, r_2 denote the root of $T = G_m$ and its two children. Then T is the disjoint union of the node r, the edges (r,r_1) and (r,r_2), and the two trees T_1 and T_2 rooted at r_1, r_2 respectively. Note that a homomorphism can be decomposed on these subtrees and vice versa. i.e.,

$$\{\phi \in \textbf{Hom} \mid \phi : T \to H\} = \left\{ a \circ \phi_1 \circ \phi_2 \mid \begin{array}{l} a \in V(H),\ \phi_i : T_i \to H,\ \phi_i \in \textbf{Hom}, \text{ and} \\ (\phi_1(r_1), a), (\phi_2(r_2), a) \in E(H) \end{array} \right\}$$

This allows us to construct the polynomial recursively.

$$\begin{aligned}
f = \sum_{\phi : T \to H} mon(\phi) &= \sum_{\substack{h \in V(H) \\ \phi_1 : T_1 \to H \in \textbf{Hom} \\ \phi_2 : T_2 \to H \in \textbf{Hom} \\ (\phi_1(r_1),h),(\phi_2(r_2),h) \in E(H)}} mon(h \circ \phi_1 \circ \phi_2) \\
&= \sum_{\substack{h,h_1,h_2 \in V(H) \\ (h,h_1),(h,h_2) \in E(H)}} \sum_{\substack{\phi_1 : T_1 \to H \\ \phi_1(r_1) = h_1}} \sum_{\substack{\phi_2 : T_2 \to H \\ \phi_1(r_2) = h_2}} Z_{r,h} Y_{h,h_1} Y_{h,h_2} mon(\phi_1) mon(\phi_2) \\
&= \sum_{\substack{h,h_1,h_2 \in V(H) \\ (h,h_1),(h,h_2) \in E(H)}} Z_{r,h} Y_{h,h_1} Y_{h,h_2} \left(\sum_{\substack{\phi_1 : T_1 \to H \\ \phi_1(r_1) = h_1}} mon(\phi_1) \right) \left(\sum_{\substack{\phi_2 : T_2 \to H \\ \phi_1(r_2) = h_2}} mon(\phi_2) \right) \\
&= \sum_{\substack{h,h_1,h_2 \in V(H) \\ (h,h_1),(h,h_2) \in E(H)}} Z_{r,h} Y_{h,h_1} Y_{h,h_2} \left(Z_{r_1,h_1} \frac{\partial f_{T_1,H,\textbf{Hom}}}{\partial Z_{r_1,h_1}} \right) \left(Z_{r_2,h_2} \frac{\partial f_{T_2,H,\textbf{Hom}}}{\partial Z_{v_2,h_2}} \right)
\end{aligned}$$

By induction, we have two circuits of size $S(\frac{m}{2}, n)$ each, computing $f_{T_1,H,\textbf{Hom}}$ and $f_{T_2,H,\textbf{Hom}}$ respectively. Using Lemma 3, we have a circuit of size $6S(\frac{m}{2}, n)$ computing both these sub-polynomials and all their derivatives. Thus we can construct f in size $6S(\frac{m}{2}, n) + O(n^3)$, giving the result. ◀

▶ Remark. In the above theorem and proof, if G_m is AT^u_m instead of T_m, essentially the same construction works. The grouping of homomorphisms should be based on the images of the root and its children and grandchildren as well.

If G_m and H_m have directions, again everything goes through the same way.

If we want to consider a restricted set $\mathcal{H}$ of homomorphisms **DirHom** or **ColHom** instead of all of **Hom**, again the same construction works. All we need is that $\mathcal{H}$ can be decomposed into independent parts with a local stitching-together operator. That is, whether ϕ belongs to $\mathcal{H}$ can be verified locally edge-by-edge and/or vertex-by-vertex, so that this can be built into the decomposition and the recursive construction.

From Theorem 18, the discussion preceding it and the remark following it, we have:

▶ **Corollary 19.** *The polynomial families from Proposition 12, Theorems 14, 15, and 16 are all in* VP*.*

▶ Remark. It is not clear how to get a similar upper bound for **InjDirHom** when the target graph is the complete directed graph (remark following Proposition 12), or for the family from Proposition 13. We need a way of enforcing that the recursive construction above respects injectivity. This is not a problem for Proposition 12, though, because there the target graph is the graph underlying a multiplicatively disjoint circuit. Injectivity at the root and its children and grandchildren can be checked locally; the recursion beyond that does not fold back because the homomorphisms are direction-preserving. The construction may not work if the target graph is the complete directed graph.

From Corollary 19, Proposition 12, and Theorems 14, 15 and 16, we obtain our main result:

▶ **Theorem 20.** ***1.*** *The polynomial families from Proposition 12 and Theorem 15 are complete for* VP *with respect to p-projections.*

2. *The polynomial families from Theorems 14 and 16 are complete for* VP *with respect to constant-depth c-reductions.*

5 Characterizing other complexity classes

We complement our result of VP-completeness by showing that appropriate modification of G can lead to VBP-complete and VNP-complete polynomial families.

VBP Completeness

VBP is the class of polynomials computed by polynomial-sized algebraic branching programs. These are layered directed graphs, with edges labeled by field constants or variables, and with a designated source node s and target node t. For any path ρ in G, the monomial $mon(\rho)$ is the product of the labels of all edges in ρ. For two nodes u, v, the polynomial p_{uv} sums $mon(\rho)$ for all paths ρ from u to v. The branching program computes the polynomial p_{st}.

A well-known polynomial family complete for VBP is the determinant of a generic matrix. A generic complete polynomial for VBP is the polynomial computed by an ABP with (1) a source node s, $m-1$ layers of m nodes each, and a target node t, (2) complete bipartite graphs between layers, and (3) distinct variables $\bar{x}$ on all edges. This is also the iterated matrix multiplication polynomial IMM. It is easy to see that st paths play the same role here as parse trees did in the multiplicatively disjoint circuits.

▶ **Theorem 21.** *Consider the homomorphism polynomial where*

- *G is a simple path on $m+1$ nodes, $(u_1, u_2, \ldots, u_{m+1})$.*
- *H is a complete graph (undirected) on m^2 nodes.*
- *$\mathcal{H} :=$ set of all homomorphisms from G to H.*

- *Define α such that,*

$$\alpha(u) = \begin{cases} 1 & u = u_1 \text{ or } u = u_{m+1} \\ 0 & \text{otherwise} \end{cases}$$

Then, the family $(f_{G,H,\alpha,\mathbf{Hom}}(\bar{X},\bar{Y}))_m$, where $m \in \mathbb{N}$, is complete for VBP *under c-reductions.*

Proof. **Hardness:** We show how IMM can be computed from this polynomial. Set the $\bar{Y}$ variables to 0 or to variables from $\bar{x}$ so that H looks like the undirected graph underlying the generic ABP. Set the X variables to 0, except at the nodes S, T corresponding to s, t. Note that in the resulting graph H', the shortest path between S and T has exactly m edges. Though there are no directions, the bijection between $\{S, T\}$ and $\{s, t\}$ is established by the setting of the $\bar{Y}$ variables.

For every st path ρ in the ABP, there is a homomorphism ϕ from G to H such that $mon(\phi) = X_s mon(\rho) X_T$. Conversely, for any homomorphism ϕ from G to H, if $mon(\phi)$ contains $X_S X_T$, then ϕ must map G to a proper path between S, T. So $mon(\phi)/X_S X_T$ is in fact $mon(\rho)$ for some st path ρ. Hence the homogeneous component of $f_{G,H,\alpha,\mathbf{Hom}}$ of degree 1 in X_S and degree 1 in X_T is exactly the generic IMM polynomial.

Membership: We show that more generally, for $G, H, \mathcal{H}$ as defined and for any 0-1 valued α, the polynomial can be computed in VBP. For any non-negative n, m, let $q_{n,m}$ denote the generalised homomorphism polynomial $f_{P_n,K_m,\mathbf{Hom}}(\bar{Z},\bar{Y})$ as defined in Section 4. (The polynomial in the Theorem statement, $f_{P_m,K_{m^2},\alpha,\mathbf{Hom}}$, is obtained from q_{m,m^2} by setting $Z_{u,a}$ to $X_a^{\alpha(u)}$.) Let us see the construction of an ABP computing $q_{n,m}$. We describe it in detail for $q_{2,m}$; it generalizes in a straightforward way to all n. Let $P_2 = \langle u, v, w\rangle$ be a path of length 2, with edges (u, v) and (v, w). We will demonstrate the ABP construction by reducing the polynomial computation to an iterated matrix multiplication instance. We need the following easily-verifiable fact.

▶ **Fact 22.** *Graph homomorphisms from a path of length 'n' to any graph H are in one-to-one correspondence with walks of length exactly 'n' in graph H.*

We claim that $q_{2,m}$ equals the following matrix product,

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^T \begin{bmatrix} Z_{u,1}Y_{1,1} & Z_{u,1}Y_{1,2} & \cdots & Z_{u,1}Y_{1,m} \\ Z_{u,2}Y_{2,1} & Z_{u,2}Y_{2,2} & \cdots & Z_{u,2}Y_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{u,m}Y_{m,1} & Z_{u,m}Y_{m,2} & \cdots & Z_{u,m}Y_{m,m} \end{bmatrix} \begin{bmatrix} Z_{v,1}Y_{1,1} & Z_{v,1}Y_{1,2} & \cdots & Z_{v,1}Y_{1,m} \\ Z_{v,2}Y_{2,1} & Z_{v,2}Y_{2,2} & \cdots & Z_{v,2}Y_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{v,m}Y_{m,1} & Z_{v,m}Y_{m,2} & \cdots & Z_{v,m}Y_{m,m} \end{bmatrix} \begin{bmatrix} Z_{w,1} \\ Z_{w,2} \\ \vdots \\ Z_{w,m} \end{bmatrix}$$

A typical entry in the final polynomial is $Z_{u,i}Y_{i,j} \times Z_{v,j}Y_{j,k} \times Z_{w,k}$. The initial row vector is only used for summation over i. So let us focus on the adjacency matrix of K_m, where each entry is multiplied with a suitable Z variable. Intuitively, the first matrix picks a vertex i in K_m to match u, and then also picks an edge (i, j) from i to map the edge (u, v) in the path. The second matrix then picks the Z variable for v, j, and chooses an edge (j, k) to map (v, w). The last column vector just picks the Z variable for w, k, as there are no more edges left. ◀

We now proceed to provide variants of path homomorphism polynomial that are complete for VBP under projections. Unfortunately, this strong completeness result comes with a caveat. The graphs underlying the homomorphism polynomials are now directed graphs. But we believe that the restriction is not a severe one (compare it with Theorem 14).

▶ **Theorem 23.** *Consider the homomorphism polynomial where*

- *G is a simple directed path on $m+1$ nodes, $\langle u_1, u_2, \ldots, u_{m+1}\rangle$.*
- *H is a complete directed graph on $m(m+1)$ nodes.*
- *$\mathcal{H} :=$ set of all directed homomorphisms from G to H.*
- *α is 1 everywhere.*

Then, the family $(f_{G,H,\alpha,\textbf{DirHom}}(\bar{X}, \bar{Y}))_m$, where $m \in \mathbb{N}$, is complete for VBP *under projections.*

Proof. Hardness: We show how IMM can be computed from this polynomial. Set the $\bar{Y}$ variables to 0 or to variables from $\bar{x}$ so that H looks like the layered directed acyclic graph underlying the generic ABP. Set the X variables to 1. Note that in the resulting graph H', the shortest path between S and T has exactly m edges.

For every *st* path ρ in the ABP, there is a homomorphism ϕ from G to H such that $mon(\phi) = mon(\rho)$. Conversely, for any homomorphism ϕ from G to H, ϕ must map G to a proper path between S, T. This follows from the directed version of Fact 22 and acyclicity of H' (which forces that paths of length m in H' exist only between S and T). So $mon(\phi)$ is in fact $mon(\rho)$ for some *st* path ρ. Hence the polynomial $f_{G,H,\alpha,\textbf{DirHom}}$ is exactly the generic IMM polynomial.

Membership: It is exactly the same as the membership proof of Theorem 21, and the correctness follows from the directed version of Fact 22.

◀

▶ **Theorem 24.** *Consider the homomorphism polynomial where*

- *G is a simple directed cycle on m nodes, $\langle u_1, u_2, \ldots, u_m, u_1\rangle$.*
- *H is a complete directed graph on m nodes.*
- *$\mathcal{H} :=$ set of all directed homomorphisms from G to H.*
- *α is 1 everywhere.*

Then, the family $(f_{G,H,\alpha,\textbf{DirHom}}(\bar{X}, \bar{Y}))_m$, where $m \in \mathbb{N}$, is complete for VBP *under projections.*

Before proceeding with the proof, we note down two facts that are needed in the proof.

▶ **Fact 25.** *Directed graph homomorphisms from a directed cycle of length 'm' to a directed graph H are in one-to-one correspondence with directed closed walks of length exactly 'm' in H.*

Consider the families of polynomials (F_m) and (G_m) defined by $F_m = \mathsf{Tr}(A^m)$ and $G_m = \mathsf{Tr}(A_1 \cdot A_2 \cdots A_m)$, where Tr is the trace, and A or A_i are generic $m \times m$ matrices with m^2 variables.

▶ **Proposition 26** ([8]). *The families (F_m) and (G_m) are* VBP*-complete over any field.*

Proof of Theorem 24. Hardness: We show how $F_m = \mathsf{Tr}(A^m)$ can be computed by this polynomial. Let H be a complete directed graph on m nodes such that Y variables are given by the matrix A and X variables are all set to 1. Now using Fact 25 it follows easily that $f_{G,H,\alpha,\textbf{DirHom}}$ is exactly $\mathsf{Tr}(A^m)$.

Membership: We show that more generally, for $G, H, \mathcal{H}$ as defined and for any 0-1 valued α, the polynomial can be computed in VBP. For any non-negative n, m, let $q_{n,m}$ denote the generalised homomorphism polynomial $f_{DC_n,DK_m,\textbf{DirHom}}(\bar{Z}, \bar{Y})$ as defined in Section 4. (The polynomial in the Theorem statement, $f_{DC_m,DK_m,\alpha,\textbf{DirHom}}$, is obtained from $q_{m,m}$ by setting $Z_{u,a}$ to X_a.)

Let us see the construction of an ABP computing $q_{n,m}$. We will demonstrate the ABP construction by reducing the polynomial computation to the trace computation of an iterated matrix multiplication instance. Consider the following iterated matrix multiplication instance.

$$W = \underbrace{\begin{bmatrix} Z_{u_1,1}Y_{1,1} & Z_{u_1,1}Y_{1,2} & \cdots & Z_{u_1,1}Y_{1,m} \\ Z_{u_1,2}Y_{2,1} & Z_{u_1,2}Y_{2,2} & \cdots & Z_{u_1,2}Y_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{u_1,m}Y_{m,1} & Z_{u_1,m}Y_{m,2} & \cdots & Z_{u_1,m}Y_{m,m} \end{bmatrix} \cdots\cdots \begin{bmatrix} Z_{u_n,1}Y_{1,1} & Z_{u_n,1}Y_{1,2} & \cdots & Z_{u_n,1}Y_{1,m} \\ Z_{u_n,2}Y_{2,1} & Z_{u_n,2}Y_{2,2} & \cdots & Z_{u_n,2}Y_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{u_n,m}Y_{m,1} & Z_{u_n,m}Y_{m,2} & \cdots & Z_{u_n,m}Y_{m,m} \end{bmatrix}}_{\text{n matrices}}$$

We claim that the *trace* of the resulting matrix, $\mathsf{Tr}(W)$, is $q_{n,m}$. Assuming the claim, the construction of an algebraic branching program computing it follows easily from Proposition 26.

To verify the claim, it suffices to show that both the polynomials have same set of monomials. We first argue that monomials in the polynomial $q_{n,m}$ appears in $\mathsf{Tr}(W)$. Each monomial corresponds to a directed homomorphism from cycle of length n to the graph. But by Fact 25, every homomorphic image corresponds to a closed walk of length n starting at the vertex where u_1 is mapped. Say u_1 is mapped to a vertex j in the graph, then this monomial also contribute to $W_{j,j}$. For the other direction, note that monomials of $\mathsf{Tr}(W)$ is uniquely identifiable by a closed walk of length n and a vertex on the walk designated as the start/end vertex. From this it is easy to recover a unique directed homomorphism for $\langle u_1, u_2, \ldots, u_n, u_1 \rangle$.

◀

VNP **Completeness**

▶ **Theorem 27.** *Consider the homomorphism polynomial where*

- *G is the complete graph (undirected) on m nodes.*
- *H is the complete graph (undirected) on m nodes.*
- *$\mathcal{H} :=$ set of all homomorphisms from G to H.*
- *All $\bar{Y}$ variables are set to 1.*

Then, the family $(f_{G,H,\boldsymbol{Hom}}(\bar{Z}))_m$, where $m \in \mathbb{N}$, is complete for VNP *under p-projections.*

Proof. We show that computing the family of permanent polynomials is equivalent to computing this family. The variables $\bar{Y}$ certify that the functions ϕ considered in the sum are homomorphisms. This property stays true if we set all $\bar{Y}$ variables to 1.

Since G and H are complete graphs without self-loops, a map $\phi : [m] \to [m]$ fails to be a homomorphism exactly when it is not injective. Thus

$$\begin{aligned} f_{G,H,\mathbf{Hom}}(\bar{1}, \bar{Z}) &= \sum_{\phi:[m]\to[m];\phi\in\mathbf{Hom}} \left(\prod_{i\in[m]} Z_{i,\phi(i)} \right) \\ &= \sum_{\phi:[m]\to[m];\phi \text{ injective}} \left(\prod_{i\in[m]} Z_{i,\phi(i)} \right) \\ &= \sum_{\phi\in S_m} \left(\prod_{i\in[m]} Z_{i,\phi(i)} \right) \\ &= \mathrm{per}_m(\bar{Z}) \end{aligned}$$

Thus $f_{G,H,\mathbf{Hom}}$ is exactly the per_m polynomial, and hence is complete for VNP. ◀

6 Conclusion

We have shown that several natural homomorphism polynomials are complete for the algebraic complexity class VP. Our results are summarised below.

Complexity	G	H	$\mathcal{H}$	polynomial type	reduction
VP-complete	AT_m	$C_{m^{O(1)}}$	**InjDirHom**	$\alpha = \underline{1}$	p-projections
	AT^d_m	$DK_{m^{O(1)}}$	**DirHom**	$\alpha = \underline{1}$	$O(1)$-depth c-reductions
	AT^c_m	coloured $K_{m^{O(1)}}$	**ColHom**	$\alpha = \underline{1}$	p-projections
	T^u_m	$K_{m^{O(1)}}$	**Hom**	0-1 valued	$O(1)$-depth c-reductions
VBP-complete	Path_m	$K_{m^{O(1)}}$	**Hom**	0-1 valued	$O(1)$-depth c-reductions
	DPath_m	$DK_{m^{O(1)}}$	**DirHom**	$\alpha = \underline{1}$	p-projections
	DCycle_m	DK_m	**DirHom**	$\alpha = \underline{1}$	p-projections
VNP-complete	K_m	K_m	**Hom**	generalised ($\bar{Z}$ variables)	p-projections

It would be interesting to show all the hardness results with respect to p-projections. It would also be very interesting to obtain completeness while allowing all homomorphisms on simple graphs and eliminating vertex weights. Another question is extending the completeness results of this paper to fields of characteristic other than zero.

Perhaps more importantly, it would be nice to get still more examples of natural VP-complete problems, preferably from different areas. The completeness of determinant or iterated matrix multiplication for VBP underlies the importance of linear algebra as a source of "efficient" computations. Finding natural VP-complete polynomials in some sense means finding computational techniques which are (believed to be) stronger than linear algebra.

References

1. Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317 – 330, 1983.
2. Christian Borgs, Jennifer T. Chayes, László Lovász, Vera T. Sós, and Katalin Vesztergombi. Counting graph homomorphisms. In *Topics in Discrete Math*, pages 315–371. Springer, 2006.
3. P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics*. Springer, 2000.
4. Florent Capelli, Arnaud Durand, and Stefan Mengel. The arithmetic complexity of tensor contractions. In *Symposium on Theoretical Aspects of Computer Science STACS*, volume 20 of *LIPIcs*, pages 365–376, 2013.
5. Nicolas de Rugy-Altherre. A dichotomy theorem for homomorphism polynomials. In *Mathematical Foundations of Computer Science 2012*, volume 7464 of *LNCS*, pages 308–322. Springer Berlin Heidelberg, 2012.
6. Arnaud Durand and Stefan Mengel. The complexity of weighted counting for acyclic conjunctive queries. *J. Comput. Syst. Sci.*, 80(1):277–296, 2014.
7. Martin E. Dyer and David Richerby. An effective dichotomy for the counting constraint satisfaction problem. *SIAM J. Comput.*, 42(3):1245–1274, 2013.
8. Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. *Journal of Complexity*, 24(1):16–38, 2008.

9 Stefan Mengel. Characterizing arithmetic circuit classes by constraint satisfaction problems. In *Automata, Languages and Programming*, volume 6755 of *LNCS*, pages 700–711. Springer Berlin Heidelberg, 2011.

10 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6:135–177, 2010.

11 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

12 L. G. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic: International Symposium in honour of Ernst Specker*, volume 30, pages 365–380. Monograph. de l'Enseign. Math., 1982.

13 Leslie G. Valiant. Completeness classes in algebra. In *Symposium on Theory of Computing STOC*, pages 249–261, 1979.

14 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de