Symmetric Arithmetic Circuits

Anuj Dawar
Department of Computer Science and Technology, University of Cambridge, UK
anuj.dawar@cl.cam.ac.uk

Gregory Wilsenach
Department of Computer Science and Technology, University of Cambridge, UK
gregory.wilsenach@cl.cam.ac.uk

Abstract

We introduce symmetric arithmetic circuits, i.e. arithmetic circuits with a natural symmetry restriction. In the context of circuits computing polynomials defined on a matrix of variables, such as the determinant or the permanent, the restriction amounts to requiring that the shape of the circuit is invariant under row and column permutations of the matrix. We establish unconditional, nearly exponential, lower bounds on the size of any symmetric circuit for computing the permanent over any field of characteristic other than 2. In contrast, we show that there are polynomial-size symmetric circuits for computing the determinant over fields of characteristic zero.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Algebraic complexity theory

Keywords and phrases arithmetic circuits, symmetric arithmetic circuits, Boolean circuits, symmetric circuits, permanent, determinant, counting width, Weisfeiler-Leman dimension, Cai-Fürer-Immerman constructions

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.36

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at `https://arxiv.org/abs/2002.06451`.

Funding Research funded in part by EPSRC grant EP/S03238X/1.

1 Introduction

Valiant's conjecture [23], that VP $\neq$ VNP, is often referred to as the algebraic counterpart to the conjecture that P $\neq$ NP. It has proved as elusive as the latter. The conjecture is equivalent to the statement that there is no polynomial-size family of arithmetic circuits for computing the permanent of a matrix, over any field of characteristic other than 2. Here, arithmetic circuits are circuits with input gates labelled by variables from some set X or constants from a fixed field $\mathbb{F}$, and internal gates labelled with the operations $+$ and $\times$. The output of such a circuit is some polynomial in $\mathbb{F}[X]$, and we think of the circuit as a compact representation of this polynomial. In particular, if the set of variables X form the entries of an $n \times n$ matrix, i.e. $X = \{x_{ij} \mid 1 \leq i, j \leq n\}$, then PERM_n denotes the polynomial $\sum_{\sigma \in \mathbf{Sym}_n} \prod x_{i\sigma(i)}$, which is the permanent of the matrix.

While a lower bound for the size of general arithmetic circuits computing the permanent remains out of reach, lower bounds have been established for some restricted classes of circuits. For example, it is known that there is no sub-exponential family of *monotone* circuits for the permanent [17, 18]. An exponential lower bound for the permanent is also known for *depth-3* arithmetic circuits [15] over finite fields. In both these cases, the exponential lower bound obtained for the permanent also applies to the determinant, i.e. the family of polynomials $\{\mathrm{DET}_n\}_{n \in \mathbb{N}}$, where DET_n is $\sum_{\sigma \in \mathbf{Sym}_n} \mathrm{sgn}(\sigma) \prod x_{i\sigma(i)}$. However, the determinant is in VP and so there do exist polynomial-size families of circuits for the determinant.

In this paper, we consider a new restriction on arithmetic circuits based on a natural notion of symmetry, and we show that it distingushes between the determinant and the permanent. That is to say, we are able to show nearly exponential lower bounds on the size of any family of symmetric arithmetic circuits for computing the permanent, while establishing the existence of polynimial-size symmetric circuits for computing the determinant. We prove the upper bound on the determinant for fields of characteristic zero, and conjecture that it holds for all fields. Our lower bound for the permanent is established for all fields of characteristic other than 2, which is the best that can be hoped for as the permanent and the determinant coincide for fields of characteristic 2.

We next define (informally) the notion of symmetry we use. A formal definition follows in Section 3. The permanent and the determinant are not symmetric polynomials in the usual meaning of the word, in that they are not invariant under arbitrary permutations of their variables. However, they do have natural symmetries, i.e. permutations of the variables induced by row and column permutations. Specifically, PERM_n is invariant under arbitrary permutations of the rows and columns of the matrix (x_{ij}), while DET_n is invariant under simultaneous permutations of the rows and columns. We say that an arithmetic circuit C (seen as a labelled directed acyclic graph) for computing DET_n is *symmetric* if the action of any permutation $\sigma \in \mathbf{Sym}(n)$ on its input variables (i.e. taking x_{ij} to $x_{\sigma(i)\sigma(j)}$) extends to an automorphism of C. Similarly, a circuit C for computing PERM_n is symmetric if the action of $(\sigma, \pi) \in \mathbf{Sym}(n) \times \mathbf{Sym}(n)$ on the inputs (taking x_{ij} to $x_{\sigma(i)\pi(j)}$) extends to an automorphism of C.

This notion of symmetry has been studied previously in the context of Boolean circuits for deciding graph properties, or properties of relational structures (see [13, 20, 2]). Specifically, such symmetric circuits arise naturally in the translation into circuit form of specifications of properties in a logic or similar high-level formalism. Similarly, we can think of a symmetric arithmetic circuit as a straight-line program which treats the rows and columns of a matrix as being indexed by unordered sets. Many natural algorithms have this property. For example, Ryser's formula for computing the permanent naturally yields a symmetric circuit.

Polynomial-size families of symmetric Boolean circuits with threshold gates form a particularly robust class, with links to fixed-point logics [2]. This allows us to deploy methods for proving inexpressiblity in such logics to prove lower bounds on the size of symmetric circuits. A close link has also been established between the power of such circuits and linear programming extended formulations with a geometric notion of symmetry [5]. Our lower bound for the permanent is established by first giving a symmetry-preserving translation of arithmetic circuits to Boolean circuits with threshold gates, and then establishing a lower bound there for computing the permanent of a 0-1-matrix.

The lower bounds for symmetric Boolean circuits are based on a measure we call the *counting width* of graph parameters (the term is introduced in [11]). This is also sometimes known as the Weisfeiler-Leman dimension. In short, we have, for each k an equivalence relation $\equiv^k$, known as the k-dimensional Weisfeiler-Leman equivalence, that is a coarse approximation of isomorphism, getting finer with increasing k. The counting width of a graph parameter μ is the smallest k, as a function of the graph size n, such that μ is constant on $\equiv^k$-classes of graphs of size n. From known results relating Boolean circuits and counting width [2, 5], we know that the existence of subexponential size symmetric circuits computing μ implies a sub-linear upper bound on its counting width. Hence, using the standard relationship between the permanent of a 0-1-matrix and the number of perfect matchings in a bipartite graph, we obtain our lower bound for the permanent in fields of characteristic zero by showing a linear lower bound on the counting width of $\mu(G)$ – the number of perfect matchings in G. Indeed, showing the same for $\mu(G) \pmod p$ for every prime $p > 2$ establishes the lower bound for the permanent in all odd positive characteristics.

The linear lower bound on the counting width of the number of perfect matchings is a result of interest in its own right, quite apart from the lower bounds it yields for circuits for the permanent. Indeed, there is an interest in determining the counting width of concrete graph parameters (see, for instance, [4]), and the result here is somewhat surprising. The decision problem of determining whether a graph has any perfect matching is known to have constant counting width. Indeed, the width is 2 for bipartite graphs [7]. For general graphs, it is known to be strictly greater than 2 but still bounded above by a constant [3].

Related Work. Landsberg and Ressayre [19] establish an exponential lower bound on the complexity of the permanent (specifically over the complex field $\mathbb{C}$) under an assumption of symmetry, and it is instructive to compare our results with theirs. Their lower bound is for the *equivariant determinantal complexity* of the permanent. The determinantal complexity (DC) of a polynomial $p \in \mathbb{F}[X]$ refers to the size of the smallest matrix M with entries that are affine linear forms in X such that $\det(M) = p$. Since every polynomial in VP has DC that is at most quasi-polynomial, an exponential lower bound on the DC of the permanent would show that circuits computing PERM_n must have size at least 2^{n^δ} for some positive δ, and hence separate VP from VNP. Landsberg and Ressayre establish exponential lower bounds on any *equivariant* determinantal representation of the permanent, that is one that preserves all the symmetries of the permanent function. This includes not just the permutations on entries that we consider, but the entire projective symmetry group. This does not, unfortunately, yield any lower bounds for symmetric circuits in the sense we consider. This is because the translation of circuits to determinantal representations that establishes that every polynomial in VP has DC at most quasi-polynomial does not preserve symmetries. A symmetric circuit does not, in general, yield a determinantal representation invariant under row-column permutations, let alone the much richer group of symmetries considered in [19]. The latter group also includes continuous group actions that have no counterpart in the realm of circuits and is specific to algebraically-closed fields of characteristic zero. It remains an interesting question to investigate whether there is a deeper connection between the lower bounds presented here and their results.

Outline. In Section 2 we introduce some preliminary definitions and notation. In Section 3, we introduce the key definitions and properties of symmetric circuits. Section 4 establishes the upper bound for symmetric circuit size for the determinant, by translating Le Verrier's method to symmetric circuits. Finally the lower bound for the permanent is established in Sections 5 and 6. The first of these gives the symmetry-preserving translation from arithmetic circuits to Boolean circuits with threshold gates, and the second gives the main construction proving the linear lower bound for the counting width of the number of perfect matchings in a bipartite graph.

2 Background

In this section we discuss relevant background and introduce notation.

We write $\mathbb{N}$ for the positive integers and $\mathbb{N}_0$ for the non-negative integers. For $m \in \mathbb{N}_0$, $[m]$ denotes the set $\{1, \ldots, m\}$. For a set X we write $\mathcal{P}(X)$ to denote the powerset of X.

2.1 Counting Width

For any $k \in \mathbb{N}$, the k-dimensional Weisfeiler-Leman equivalence (see [8]), denoted $\equiv^k$ is an equivalence relation on graphs that provides an over-approximation of isomorphism in the sense that for isomorphic graphs G and H, we have $G \equiv^k H$ for all k. Increasing values of

k give finer relations, so $G \equiv^{k+1} H$ implies $G \equiv^k H$ for all k. The equivalence relation is decidable in time $n^{\mathcal{O}(k)}$, where n is the size of the graphs. If $k \geq n$, then $G \equiv^k H$ implies that G and H are isomorphic. The Weisfeiler-Leman equivalences have been widely studied and they have many equivalent characterizations in combinatorics, logic, algebra and linear optimization. One of their many uses has been to establish inexpressibility results in logic. These can be understood through the notion of *counting width*.

A *graph parameter* is a function from graphs to $\mathbb{N}$ which is isomorphism invariant. Examples are the chromatic number, the number of connected components or the number of perfect matchings. For a graph parameter μ and any fixed $n \in \mathbb{N}$, there is a smallest value of k such that μ is $\equiv^k$-invariant. This motivates the definition.

▶ **Definition 1.** *For any graph parameter μ, the* counting width *of μ is the function $\nu : \mathbb{N} \to \mathbb{N}$ such that $\nu(n)$ is the smallest k such that for all graphs G, H of size n, if $G \equiv^k H$, then $\mu(G) = \mu(H)$.*

The *counting width* of a class of graphs $\mathcal{C}$ is the counting width of its indicator function. This notion of counting width for classes of graphs was introduced in [11], which we here extend to graph parameters. Note that for any graph parameter $\nu(n) \leq n$.

Cai, Fürer and Immerman [8] first showed that there is no fixed k for which $\equiv^k$ coincides with isomorphism. Indeed, in our terminology, they construct a class of graphs with counting width $\Omega(n)$. Since then, many classes of graphs have been shown to have linear counting width, including the class of Hamiltonian graphs and the class of 3-colourable graphs (see [5]). In other cases, such as the class of graphs that contain a perfect matching, it has been proved that they have counting width bounded by a constant [3]. Our interest in counting width stems from the relation between this measure and lower bounds for symmetric circuits. Roughly, if a class of graphs is recognized by a family of polynomial-sized symmetric threshold circuits, it has bounded counting width (a more precise statement is given in Theorem 13).

Our lower bound construction in Section 6 is based on the graphs constructed by Cai et al. [8]. While we review some of the details of the construction in Section 6, a reader unfamiliar with the construction may wish to consult a more detailed introduction. The original construction can be found in [8] and a version closer to what we use is given in [10].

2.2 Circuits

We provide a general definition that incorporates both Boolean and arithmetic circuits.

▶ **Definition 2** (Circuit)**.** *A* circuit *over the* basis $\mathbb{B}$ *with* variables X *and* constants K *is a directed acyclic graph with a labelling where each vertex of in-degree* 0 *is labelled by an element of $X \cup K$ and each vertex of in-degree greater than* 0 *is labelled by an element of* $\mathbb{B}$.

Let $C = (G, W)$, where $W \subset G \times G$, be a circuit with constants K. We call the elements of G *gates*, and the elements of W *wires*. We call the gates with in-degree 0 *input gates* and gates with out-degree 0 *output gates*. We call those input gates labelled by elements of K *constant gates*. We call those gates that are not input gates *internal gates*. For $g, h \in G$ we say that h is a *child* of g if $(h, g) \in W$. We write $\text{child}(g)$ to denote the set of children of g. We write C_g to denote the sub-circuit of C rooted at g. Unless otherwise stated we always assume a circuit has exactly one output gate.

If K is a field $\mathbb{F}$, and $\mathbb{B}$ is the set $\{+, \times\}$, we have an *arithmetic circuit* over $\mathbb{F}$. If $K = \{0, 1\}$, and $\mathbb{B}$ is a collection of Boolean functions, we have a *Boolean circuit* over the basis $\mathbb{B}$. We define two Boolean bases here. The *standard basis* $\mathbb{B}_{\text{std}}$ contains the functions

$\wedge$, $\vee$, and $\neg$. The *threshold basis* $\mathbb{B}_t$ is the union of $\mathbb{B}_{\text{std}}$ and $\{t_{\geq k} : k \in \mathbb{N}\}$, where for each $k \in \mathbb{N}$, $t_{\geq k}$ is defined for a string $\vec{x} \in \{0,1\}^*$ so that $t_{\geq k}(\vec{x}) = 1$ if, and only if, the number of 1s in $\vec{x}$ at least k. We call a circuit defined over this basis a *threshold circuit*. Another useful function is $t_{=k}$, which is defined by $t_{=k}(x) = t_{\geq k}(x) \wedge \neg t_{\geq k+1}(x)$. We do not explicitly include it in the basis as it is easily defined in $\mathbb{B}_t$.

In general, we require that a basis contain only functions that are invariant under all permutations of their inputs (we define this notion formally in Definition 4). This is the case for the arithmetic functions $+$ and $\times$ and for all of the Boolean functions in $\mathbb{B}_t$ and $\mathbb{B}_{\text{std}}$. Let C be a circuit defined over such a basis with variables X and constants K. We evaluate C for an assignment $M \in K^X$ by evaluating each gate labelled by some $x \in X$ to $M(x)$ and each gate labelled by some $k \in K$ to k, and then recursively evaluating each gate according to its corresponding basis element. We write $C[M](g)$ to denote the value of the gate g and $C[M]$ to denote the value of the output gate. We say that C computes the function $M \mapsto C[M]$.

It is conventional to consider an arithmetic circuit C over $\mathbb{F}$ with variables X to be computing a polynomial in $\mathbb{F}[X]$, rather than a function $\mathbb{F}^X \to \mathbb{F}$. This polynomial is defined via a similar recursive evaluation, except that now each gate labelled by a variable evaluates to the corresponding formal variable, and we treat addition and multiplication as ring operations in $\mathbb{F}[X]$. Each gate then evaluates to some polynomial in $\mathbb{F}[X]$. The polynomial computed by C is the value of the output gate.

For more details on arithmetic circuits see [22] and for Boolean circuits see [24].

3 Symmetric Circuits

In this section we discuss different symmetry conditions for functions and polynomials. We also introduce the notion of a symmetric circuit.

3.1 Symmetric Functions

There is a natural extension of a group action on a set X to functions on X and powers of X.

▶ **Definition 3.** *For any group G, we say that a function $F : K^X \to K$, along with an action of G on X is a* G-symmetric function*, if for every $\sigma \in G$, $\sigma F = F$.*

We are interested in some specific group actions, and we define these next.

▶ **Definition 4.**

- *If $G = \boldsymbol{Sym}(X)$, we call a G-symmetric function $F : K^X \to K$* fully symmetric*.*
- *If $G = \boldsymbol{Sym}(X) \times \boldsymbol{Sym}(Y)$, we call a G-symmetric function $F : K^{X \times Y} \to K$* matrix symmetric*.*
- *If $G = \boldsymbol{Sym}(X)$ we call a G-symmetric $F : K^{X \times X} \to K$, with the natural action of G on $X \times X$* square symmetric*.*

Examples of fully symmetric functions are those that appear as labels of gates in a circuit, including $+$, $\times$, $\wedge$, $\vee$ and $t_{\geq k}$. Matrix symmetric functions are those where the input is naturally seen as a matrix and the result in invariant under aribtrary row and column permutations. The canonical example for us of a matrix symmetric function is the permanent. The determinant is not matrix symmetric over fields of characteristic other than 2, but it is square symmetric. The determinant is also invariant under taking matrix transposes, and we also consider this variation.

▶ **Definition 5.** *Let G be the group generated by the diagonal of $\boldsymbol{Sym}(X) \times \boldsymbol{Sym}(X)$ and the permutation $\sigma_t \in \boldsymbol{Sym}(X) \times \boldsymbol{Sym}(X)$ defined such that $\sigma_t(x, y) = (y, x)$. A function $F : K^{X \times X} \to K$ that is G-symmetric with respect to the natural action of G on $X \times X$ is said to be* transpose symmetric*.*

Finally, another useful notion of symmetry in functions is where the inputs are naturally partitioned into sets.

▶ **Definition 6.** *If $X = \biguplus_{i\in I} X_i$, $G = \prod_{i\in I} \boldsymbol{Sym}(X_i)$, and $F : K^X \to K$ is G-symmetric with respect to the natural action of G on X, we say that it is* partition symmetric.

Unless otherwise stated we treat the permanent, $\mathrm{perm} : \mathbb{F}^{X\times Y} \to \mathbb{F}$ as a matrix symmetric function, and the determinant $\det : \mathbb{F}^{X\times X} \to \mathbb{F}$ as a transpose symmetric function.

3.2 Symmetric Circuits

Symmetric Boolean circuits have been considered in the literature, particularly in connection with definability in logic. In that context, we are considering circuits which take relational structures (such as graphs) as inputs and we require their computations to be invariant under re-orderings of the elements of the structure. Here, we generalize the notion to arbitrary symmetry groups, and also consider them in the context of arithmetic circuits. In order to define symmetric circuits, we first need to define the automorphisms of a circuit.

▶ **Definition 7** (Circuit Automorphism)**.** *Let $C = (G, W)$ be a circuit over the basis $\mathbb{B}$ with variables X and constants K. For $\sigma \in \boldsymbol{Sym}(X)$, we say that a bijection $\pi : G \to G$ is an* automorphism *extending σ if for every gate g in C we have that*

- *if g is a constant gate then $\pi(g) = g$,*
- *if g is a non-constant input gate then $\pi(g) = \sigma(g)$,*
- *if $(h, g) \in W$ is a wire, then so is $(\pi h, \pi g)$*
- *if g is labelled by $b \in \mathbb{B}$, then so is $\pi(g)$.*

We say that a circuit C with variables X is *rigid* if for every permutation $\sigma \in \mathbf{Sym}(X)$ there is at most one automorphism of C extending σ.

We are now ready to define the key notion of a symmetric circuit.

▶ **Definition 8** (Symmetric Circuit)**.** *For a G-symmetric function $F : K^X \to K$, a circuit C computing F is said to be* symmetric *if for every $\sigma \in G$, the action of σ on X extends to an automorphism of C. We say C is* strictly symmetric *if it has no other automorphisms.*

For a gate g in a symmetric circuit C, the *orbit* of g, denoted by $\mathbf{Orb}(g)$, is the the set of all $h \in C$ such that there exists an automorphism π of C with $\pi(g) = h$. We write $|\mathbf{Orb}(C)|$ for the maximum size of an orbit in C, and call it the *orbit size* of C.

Though symmetric arithmetic circuits have not previously been studied, symmetric Boolean circuits have [13, 20, 2]. It is known that polynomial-size symmetric threshold circuits are more powerful than polynomial-size symmetric circuits over the standard basis [2]. In particular, the majority function is not computable by any family of polynomial-size symmetric circuits over the standard basis. On the other hand, it is also known [12] that adding any fully symmetric functions to the basis does not take us beyond the power of the threshold basis. Thus, $\mathbb{B}_t$ gives the robust notion, and that is what we use here. It is also this that has the tight connection with counting width mentioned above.

3.3 Polynomials

In the study of arithmetic complexity, we usually think of a circuit over a field $\mathbb{F}$ with variables in X as expressing a polynomial in $\mathbb{F}[X]$, rather than computing a function from $\mathbb{F}^X$ to $\mathbb{F}$. The distinction is significant when $\mathbb{F}$ is a finite field, as it is possible for distinct polynomials to represent the same function.

The definitions of symmetric functions given in Section 3.1 extend easily to polynomials. So, for a group G acting on X, a polynomial $p \in \mathbb{F}[X]$ is said to be G-symmetric if $\sigma p = p$ for all $\sigma \in G$. We define *fully symmetric*, *matrix symmetric*, *square symmetric* and *transpose symmetric* polynomials analogously. Every matrix symmetric polynomial is also square symmetric. Also, every transpose symmetric polynomial is square symmetric. The permanent PERM_n is both matrix symmetric and transpose symmetric, while the determinant DET_n is transpose symmetric, but not matrix symmetric. In this paper, we treat PERM_n as a matrix symmetric polynomial and DET_n as a transpose symmetric polynomial. It is clear that a G-symmetric polynomial determines a G-symmetric function.

An arithmetic circuit C expressing a G-symmetric polynomial is said to be *symmetric* if the action of each $\sigma \in G$ on the inputs of C extends to an automorphism of C.

Standard *symmetric polynomials* are, in our terminology, fully symmetric. In particular, the homogeneous polynomial $\sum_{i\in[n]} x_i^r$ is fully symmetric. There is a known lower bound of $\Omega(n \log r)$ on the size of any circuit expressing this polynomial [6]. Notably, the matching upper bound is achieved by a symmetric circuit. Similarly, we have tight upper and lower bounds for the elementary symmetric polynomials $\sum_{S\subseteq[n]:|S|=k} \prod_{i\in S} x_i$ over infinite fields [21]. Again, the upper bound is achieved by symmetric circuits.

The best known upper bound for general arithmetic circuits for expressing the permanent is given by Ryser's formula: $\mathrm{PERM}_n = (-1)^n \sum_{S\subseteq[n]} (-1)^{|S|} \prod_{i=1}^{n} \sum_{j\in S} x_{ij}$. It is easily seen that this expression is symmetric, and it yields a symmetric circuit of size $\mathcal{O}(2^n n^2)$. Our main result, Theorem 12, gives us a near matching lower bound on the size of symmetric circuits for expressing PERM_n.

A symmetric circuit C expressing a G-symmetric polynomial p is also a symmetric circuit computing the function determined by p. In establishing our upper bound for the determinant, we show the existence of small symmetric circuits for the polynomial, and hence also for the function. For the lower bound on the permanent, we show that there are no small symmetric circuits for computing the function, hence also none for the polynomial. For a discussion of functional lower bounds, as opposed to polynomial lower bounds, see [14].

4 An Upper-Bound for the Determinant

In this section we show that for any field $\mathbb{F}$ of characteristic 0 there is a polynomial-size family of symmetric arithmetic circuits over $\mathbb{F}$ computing $\{\mathrm{DET}_n\}$. We define this family using Le Verrier's method for calculating the characteristic polynomial of a matrix. We review this method briefly, and direct the reader to Section 3.4.1 in [16] for more detail.

The characteristic polynomial of an $n \times n$ matrix M is

$$\det(xI_n - M) = \prod_{i=1}^{n}(x - \lambda_i) = x^n - p_1 x^{n-1} + p_2 x^{n-2} - \ldots + (-1)^n p_n,$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of M, counted with multiplicity. It is known that $p_n = \det(M)$ and $p_1 = \mathrm{Tr}(M)$. Le Verrier's method gives, for each $i \in [n]$, the linear recurrence given by $p_i = \frac{1}{i}[p_{i-1}s_1 - p_{i-2}s_2 + \ldots \pm s_i]$ where $p_0 = 1$ and for each $j \in [n]$, $s_j = \mathrm{Tr}(M^j)$.

The determinant can thus be computed as follows. First, for each $k \in [n]$ we compute entries in the matrix M^k. Then for each $k \in [n]$ we compute $s_k = \mathrm{Tr}(M^k)$. Finally, we recursively compute each p_i and output p_n. There is a natural arithmetic circuit Φ with variables $M = \{m_{ij} : i, j \in [n]\}$ implementing this algorithm.

To see that Φ is symmetric we pick some $\sigma \in \mathbf{Sym}(n) \times \mathbf{Sym}(n)$ such that σ is either in the diagonal or $\sigma(i,j) = (j,i)$ for all $i,j \in [n]$. We now retrace the description of the algorithm implemented by Φ and show how the action of σ on the input gates extends naturally to an automorphism π of Φ. For each input gate labelled by a variable v let $\pi(v) = \sigma(v)$, and let π fix each constant gate. For each gate v computing $(M^k)_{ij}$ let $\pi(v)$ be the gate computing $(M^k)_{\sigma(i,j)}$. For each $k \in [n]$ let v_k be the gate computing the trace of M^k. Then v_k is the sum of those gates computing the diagonal of M^k, which is fixed setwise by π, and so we let $\pi(v_k) = v_k$. We note that each p_i is defined in terms of $s_1, \dots, s_i$ and constants $-1, 1, \frac{1}{2}, \dots, \frac{1}{i}$. All of these gates are fixed by π and so we can take π to fix all remaining gates in the circuit.

It is possible to show that the construction of Φ for a given $n \in \mathbb{N}$ can be carried out in time $\mathcal{O}(n^3)$. In particular, we have the following.

▶ **Theorem 9.** *For $\mathbb{F}$ a field of characteristic 0, there exists a family of symmetric arithmetic circuits $(\Phi_n)_{n\in\mathbb{N}}$ over $\mathbb{F}$ computing $\{\mathrm{DET}_n\}$ as a family of transpose symmetric polynomials and for which the function $n \mapsto \Phi_n$ is computable in time $\mathcal{O}(n^3)$.*

Le Verrier's method explicitly involves multiplications by $\frac{1}{k}$ for $k \in [n]$, and so cannot be directly applied to fields of positive characteristic. There are many known algorithms for computing the determinant over fields of positive characteristic, and it seems reasonable to conjecture that some could be implemented symmetrically.

5 From Arithmetic To Boolean Circuits

In this section we establish the following symmetry-preserving translation from arithmetic circuits to threshold circuits.

▶ **Theorem 10.** *Let G be a group acting on a set of variables X. Let Φ be a symmetric arithmetic circuit over a field $\mathbb{F}$ with variables X and computing a G-symmetric function. Let $B \subseteq \mathbb{F}$ be finite. Then there is a symmetric threshold circuit C with variables X, such that for all $M \in \{0,1\}^X$ we have $C[M] = 1$ if, and only if, $\Phi[M] \in B$ and $|\boldsymbol{Orb}(C)| = |\boldsymbol{Orb}(\Phi)|$.*

We use Theorem 10 in Section 6 to transfer a lower bound on threshold circuits to arithmetic circuits, a crucial step in establishing our lower bound for the permanent.

We prove Theorem 10 by first establishing a similar translation from arithmetic circuits over a field $\mathbb{F}$ to Boolean circuits over a basis $\mathbb{B}^{\mathbb{F}}_{\mathrm{arth}}$ of partition symmetric functions. We then complete the proof by replacing each gate labelled by a partition symmetric function with an appropriate symmetric threshold circuit.

To enable this second step, we show that each partition symmetric function can be computed by a rigid strictly symmetric threshold circuit. The proof of this result follows from the fact that if a function $F : \{0,1\}^A \to \{0,1\}$ is partition symmetric, then its output for $h \in \{0,1\}^A$ depends only on the *number* of elements in each part of A that h maps to 1. We can thus evaluate F by counting the number of 1s in each part, a procedure which we now show can be implemented via a symmetric threshold circuit.

▶ **Lemma 11.** *Let F be a partition symmetric function. There exists a rigid strictly symmetric threshold circuit $C(F)$ computing F.*

Proof. Let $A := \biguplus_{q\in Q} A_q$ be a disjoint union of finite sets A_q indexed by Q, and $F : \{0,1\}^A \to \{0,1\}$ be a partition symmetric function. The fact that F is partition symmetric means that whether $F(h) = 1$ for some $h \in \{0,1\}^A$ is determined by the number of $a \in A_q$

(for each q) for which $h(a) = 1$. Write h_q for this number. Then, there is a set $c_F \subseteq \mathbb{N}_0^Q$ such that $F(h) = 1$ if, and only if, $(h_q)_{q\in Q} \in c_F$. Since each A_q is finite, so is c_F. Then $F(h) = 1$ if, and only if, the following Boolean expression is true: $\bigvee_{c\in c_F} \bigwedge_{q\in Q}(h_q = c(q))$. We can turn this expression into a circuit C with an OR gate at the output, whose children are AND gates, one for each $c \in c_F$, let us call it $\wedge_c$. The children of $\wedge_c$ are a set of gates, one for each $q \in Q$, let us call it $T_{c,q}$, which is labelled by $t_{=c(q)}$ and has as children all the inputs $a \in A_q$.

This circuit C is symmetric and rigid, but not necessarily strictly symmetric, as it may admit automorphisms that do not respect the partition of the inputs A as $\biguplus_{q\in Q} A_q$. To remedy this, we create pairwise non-isomorphic gadgets G_q, one for each $q \in Q$. Each G_q is a one-input, one-output circuit computing the identity function. For example, G_q could be a tower of single-input AND gates, and we choose a different height for each q. We now modify C to obtain $C(F)$ by inserting between each input $a \in A_q$ and each gate $T_{c,q}$ a copy G_q^a of the gadget G_q.

Clearly $C(F)$ computes F. We now argue $C(F)$ is rigid and strictly symmetric. To see that it is symmetric, consider any $\sigma \in \prod_{q\in Q} \mathbf{Sym}(A_q)$ in its natural action on A. This extends to an automorphism of $C(F)$ that takes the gadget G_q^a to $G_q^{\sigma a}$ while fixing all gates $T_{c,q}$ and $\wedge_c$. To see that there are no other automorphisms, suppose π is an automorphism of $C(F)$. It must fix the output OR gate. Also π cannot map a gate $T_{c,q}$ to $T_{c',q'}$ for $q' \neq q$ because the gadgets G_q and $G_{q'}$ are non-isomorphic. Suppose that π maps $\wedge_c$ to $\wedge_{c'}$. Then, it must map $T_{c,q}$ to $T_{c',q}$. Since the labels of these gates are $t_{=c(q)}$ and $t_{=c'(q)}$ respectively, we conclude that $c(q) = c'(q)$ for all q and therefore $c = c'$. ◀

We now define for each field $\mathbb{F}$ the basis $\mathbb{B}^{\mathbb{F}}_{\text{arth}}$. The functions in this basis are intended to be Boolean analogues of addition and multiplication. Let $Q \subseteq \mathbb{F}$ be finite, $A = \biguplus_{q\in Q} A_q$ be a disjoint union of non-empty finite sets, and $c \in \mathbb{F}$. We define a function $+^A_{Q,c} : \{0,1\}^A \to \{0,1\}$ that given $h \in \{0,1\}^A$ computes the sum over all q of the number of elements of A_q that h maps to 1, weighted by q, and returns 1 if this sum equals c. We also define an analogous function for multiplication $\times^A_{Q,c} : \{0,1\}^A \to \{0,1\}$. Formally, these functions are defined for $h \in \{0,1\}^A$ as follows: $+^A_{Q,c}(h) = 1$ if, and only if, $\sum_{q\in Q} |\{a \in A_q : h(a) = 1\}| \cdot q = c$ and $\times^A_{Q,c}(h) = 1$ if, and only if, $\prod_{q\in Q} q^{|\{a\in A_q : h(a)=1\}|} = c$. Both $+^A_{Q,c}$ and $\times^A_{Q,c}$ are partition symmetric. Let $\mathbb{B}^{\mathbb{F}}_{\text{arth}}$ be the set of all functions $+^A_{Q,c}$ and $\times^A_{Q,c}$.

We aim to prove Theorem 10 by first defining for a given symmetric arithmetic circuit a corresponding symmetric circuit over a partition symmetric basis. To ensure unambiguous evaluation, the circuit must include for each gate labelled by a partition symmetric function a corresponding partition on its children. Let C be a circuit with variables X and let g be a gate in C labelled by a partition symmetric function $F : \{0,1\}^A \to \{0,1\}$, where $A = \biguplus_{q\in Q} A_q$ is a disjoint union of finite non-empty sets. We associate with g a bijection $L_g : A \to \text{child}(g)$. We evaluate g for an input as follows. For $M \in \{0,1\}^X$ we let $L_g^M : A \to \{0,1\}$ be defined such that $L_g^M(a) = C[M](L_g(a))$ for all $a \in A$. Let $C[M](g) = F(L_g^M)$.

Proof of Theorem 10. For $v \in \Phi$ let Q_v be the set of possible evaluations of v if the input gates are assigned to 0 or 1, i.e. $Q_v = \{\Phi[M](v) : M \in \{0,1\}^X\}$. The restriction to 0-1-matrices ensures that Q_v is finite. Let z be the output gate of Φ. If $Q_z \subseteq B$ let C be the circuit consisting of a single gate labelled by 1 and if $Q_z \cap B = \emptyset$ let C consist of a single gate labelled by 0. Suppose that neither of these two cases hold.

We now construct a $\mathbb{B}^{\mathbb{F}}_{\text{arth}} \cup \mathbb{B}_{\text{std}}$-circuit D from Φ by replacing each internal gate v in Φ with a family of gates (v, q) for $q \in Q_v$ such that $D[M](v, q) = 1$ if, and only if, $\Phi[M](v) = q$. Each (v, q) is labelled by a function of the form $+^A_{Q,q}$ or $\times^A_{Q,q}$, depending on if v is an addition or multiplication gate. We also add a single output gate in D that has as children exactly those gates (z, q) where $q \in Q_z \cap B$. We define D from Φ recursively as follows. Let $v \in \Phi$.

- If v is an non-constant input gate in Φ let $(v,1)$ be an input gate in D labelled by the same variable as v and let $(v,0)$ be a NOT-gate with child $(v,1)$.
- If v is a constant gate in Φ labelled by some field element q let (v,q) be a constant gate in D labelled by 1.
- Suppose v is an internal gate. Let $Q = \bigcup_{u \in \text{child}(v)} Q_v$. For $q \in Q$ let $A_q = \{u \in \text{child}(v) : q \in Q_u\}$. Let $A = \biguplus_{q \in Q} A_q$. For each $c \in Q_v$ let (v,c) be a gate in D such that if v is an addition gate or multiplication gate then (v,c) is labelled by $+^A_{Q,c}$ or $\times^A_{Q,c}$, respectively. The labelling function $L_{(v,c)} : A \to \text{child}(v,c)$ is defined for $u \in A$ such that if $u \in A_q$ then $L_{(v,c)}(u) = (u,q)$.

We add one final OR-gate w to form D with $\text{child}(w) = \{(z,q) : q \in B \cap Q_z\}$.

We now show that D is a symmetric circuit. Let $\sigma \in G$ and π be an automorphism of Φ extending σ. Let $\pi' : D \to D$ be defined such that for each gate $(v,c) \in D$, $\pi'(v,c) = (\pi(v),c)$ and for the output gate w, $\pi'(w) = w$. It can be verified by induction that π' is an automorphism of C extending σ.

We now show that $|\mathbf{Orb}(D)| = |\mathbf{Orb}(\Phi)|$. It suffices to prove that for $v,u \in \Phi$ and $c \in Q_v$ that $u \in \mathbf{Orb}(v)$ if, and only if, $(u,c) \in \mathbf{Orb}(v,c)$. The forward direction follows from the above argument establishing that D is symmetric. Let $v,u \in \Phi$ and $c \in Q_v$ and suppose $(u,c) \in \mathbf{Orb}(v,c)$. For each gate $t \in \Phi$ pick some $c_t \in Q_t$ such that if $t = u$ or $t = v$ then $c_t = c$ and for all $t_1, t_2 \in \Phi$, if $Q_{t_1} = Q_{t_2}$ then $c_{t_1} = c_{t_2}$. Let π' be an automorphism of D such that $\pi'(v,c) = (u,c)$. Let $\pi : \Phi \to \Phi$ be defined for $t \in \Phi$ such that $\pi'(t,c_t) = (\pi(t),c_t)$. We now show that π is an automorphism of Φ, and so $u \in \mathbf{Orb}(v)$. Note that, since π' preserves the labelling on the gates in D, it follows that for all $t \in \Phi$, $Q_t = Q_{\pi(t)}$ and so $c_{\pi(t)} = c_t$. Let $t,t' \in \Phi$ and suppose $\pi(t) = \pi(t')$. Then $\pi'(t,c_t) = (\pi(t),c_t) = (\pi(t),c_{\pi(t)}) = (\pi(t'),c_{\pi(t')}) = (\pi(t'),c_{t'}) = \pi'(t',c_{t'})$, and so $(t,c_t) = (t',c_{t'})$ and $t = t'$. It follows that π is injective, and so bijective. Let $t,s \in \Phi$. Then $t \in \text{child}(s) \iff (t,c_t) \in \text{child}(s,c_s) \iff \pi'(t,c_t) \in \text{child}(\pi'(s,c_s)) \iff (\pi(t),c_t) \in \text{child}(\pi(s),c_s) \iff \pi(t) \in \text{child}(\pi(s))$. The first and last equivalences follow from the construction of the circuit. The remaining conditions for π to be an automorphism can be easily verified.

Let $M \in \{0,1\}^X$. We now show by induction that for all $v \in \Phi$ and $c \in Q_v$, $\Phi[M](v) = c$ if, and only if, $D[M](v,c) = 1$. Let $v \in \Phi$. If v is an input gate then the claim holds trivially. Suppose v is an internal gate and let $c \in Q_v$. Suppose v is an addition gate. Then (v,c) is labelled by the function $+^A_{Q,c}$ where $Q = \bigcup_{u \in \text{child}(v)} Q_u$, for $q \in Q$, $A_q = \{u \in \text{child}(v) : q \in Q_u\}$, and $A = \biguplus_{q \in Q} A_q$. Then

$$\begin{aligned}
\Phi[M](v) = c &\iff \sum_{u \in \text{child}(v)} \Phi[M](u) = c \iff \sum_{q \in Q} |\{u \in \text{child}(v) : \Phi[M](u) = q\}| \cdot q = c \\
&\iff \sum_{q \in Q} |\{u \in A_q : D[M](u,q) = 1\}| \cdot q = c \\
&\iff \sum_{q \in Q} |\{u \in A_q : L^M_{(v,c)}(u) = 1\}| \cdot q = c \\
&\iff D[M](v,c) = 1
\end{aligned}$$

A similar argument suffices if v is a multiplication gate. It follows that $D[M](w) = 1$ if, and only if, there exists $c \in B$ such that $D[M](z,c) = 1$ if, and only if, $\Phi[M] \in B$.

We define C from D by replacing each internal gate $(v,c) \in D$ labelled by some $F \in \mathbb{B}^{\mathbb{F}}_{\text{arth}}$ with the rigid strictly symmetric threshold circuit $C(F)$ computing F defined in Lemma 11. C computes the same function as D. Since $C(F)$ is symmetric, C is symmetric. Since $C(F)$ is rigid and *strictly* symmetric, $|\mathbf{Orb}(C)| = |\mathbf{Orb}(D)| = |\mathbf{Orb}(\Phi)|$. ◀

6 A Lower-Bound for the Permanent

We now establish the lower bound on symmetric arithmetic circuits for the permanent.

▶ **Theorem 12.** *If $\mathbb{F}$ is a field with char$(\mathbb{F}) \neq 2$, then for any $\epsilon > 0$ there is no family of symmetric arithmetic circuits over $\mathbb{F}$ of orbit size $\mathcal{O}(2^{n^{1-\epsilon}})$ computing $\{\mathrm{PERM}_n\}$.*

Our proof establishes something stronger. We actually show that there are no symmetric arithmetic circuits of orbit size $\mathcal{O}(2^{n^{1-\epsilon}})$ that compute the function perm(M) for matrices $M \in \mathbb{F}^{n\times n}$. Indeed, the lower bound holds even when restricting the input to matrices $M \in \{0,1\}^{n\times n}$. Theorem 12 is proved by showing lower bounds on the counting widths of functions which determine the number of perfect matchings in a bipartite graph. The connection of orbit size to counting width comes through the following theorem (see [2, 5]).

▶ **Theorem 13.** *Let $(C_n)_{n\in\mathbb{N}}$ be a family of symmetric threshold circuits of orbit size $s = \mathcal{O}(2^{n^{1-\epsilon}})$ for some $\epsilon > 0$ deciding a class of graphs $\mathcal{C}$. Then, the counting width of $\mathcal{C}$ is $\mathcal{O}(\frac{\log s}{\log n})$.*

If G is a bipartite graph, let $\mu(G)$ denote the number of perfect matchings in G and, for a prime number p, we write $\mu^p(G)$ for the congruence class of $\mu(G) \pmod p$. It is well known if G is a balanced bipartite graph with vertex bipartition $V(G) = A \cup B$, and $M_G \in \{0,1\}^{A\times B}$ is the biadjacency matrix of G, then the permanent of M_G (say, over the rational field $\mathbb{Q}$) is the number of distinct perfect matchings of G. Moreover, since M_G is a 0-1-matrix, $\mathrm{perm}_{\mathbb{F}}(M_G) = \mathrm{perm}_{\mathbb{F}'}(M_G)$ whenever $\mathbb{F}'$ is a subfield of $\mathbb{F}$. In particular, for any field $\mathbb{F}$ of characteristic zero, $\mathrm{perm}_{\mathbb{F}}(M_G) = \mathrm{perm}_{\mathbb{Q}}(M_G) = \mu(G)$ and for any field $\mathbb{F}$ of characteristic p, $\mathrm{perm}_{\mathbb{F}}(M_G) = \mathrm{perm}_{\mathbb{F}_p}(M_G) = \mu^p(G)$. To avoid unnecessary case distinctions, we write $\mu^c(G)$ where c is either 0 or a prime p, with the understanding that $\mu^0(G) = \mu(G)$. Then, we can say that for any field $\mathbb{F}$ with $\mathrm{char}(\mathbb{F}) = c$, $\mathrm{perm}_{\mathbb{F}}(M_G) = \mu^c(G)$.

Combining Theorem 10 with Theorem 13 gives us the following consequences.

▶ **Corollary 14.** *If there exists a family of symmetric circuits of orbit size $s = \mathcal{O}(2^{n^{1-\epsilon}})$ over a field $\mathbb{F}$ of characteristic c computing $\{\mathrm{PERM}_n\}$, then the counting width of μ^c is $\mathcal{O}(\frac{\log s}{\log n})$.*

Proof. Let k be the counting width of μ^c. Then, by definition, we can find for each $n \in \mathbb{N}$, a pair of balanced bipartite graphs G_n and H_n on at most $2n$ vertices such that $G_n \equiv^{k(n)-1} H_n$ but $\mu(G_n) \neq \mu(H_n)$. Let $B_n = \{\mu(G_n)\}$. Then, by Theorem 10 and the assumption that there is a family of symmetric circuits over $\mathbb{F}$ computing $\{\mathrm{PERM}_n\}$ of orbit size $s = \mathcal{O}(2^{n^{1-\epsilon}})$, there is a family of symmetric Boolean threshold circuits of orbit size $s = \mathcal{O}(2^{n^{1-\epsilon}})$ which decides for a matrix $M \in \{0,1\}^{n\times n}$ whether perm$(M) \in B_n$. In other words, when $c = 0$, this family of circuits then decides whether a balanced bipartite graph G on $2n$ vertices has exactly $\mu(G_n)$ perfect matchings, and when $c = p$ for some prime p, it decides whether G has $\mu^p(G_n)$ perfect matchings, modulo p. It follows by Theorem 13 that the counting width of this decision problem is $\mathcal{O}(\frac{\log s}{\log n})$. Since the counting width of this decision problem is, by choice of G_n, k, it follows that $k = \mathcal{O}(\frac{\log s}{\log n})$. ◀

Thus, to establish Theorem 12, we aim to prove the following.

▶ **Theorem 15.** *There are, for each $k \in \mathbb{N}$, a pair of balanced bipartite graphs X and Y with $\mathcal{O}(k)$ vertices, such that $X \equiv^k Y$, and $\mu(X) - \mu(Y) = 2^l$ for some $l > 0$.*

Before giving the proof of Theorem 15 we show how Theorem 12 now follows.

Proof of Theorem 12. By Theorem 15, we have for each k, a pair of graphs X and Y with $\mathcal{O}(k)$ vertices such that $\mu(X) \neq \mu(Y)$ and $X \equiv^k Y$ thus, the counting width of μ is $\Omega(n)$. Moreover, since $\mu(X) - \mu(Y)$ is a power of 2, it follows that for any prime $p \neq 2$, $\mu(X) \not\equiv \mu(Y) \pmod{p}$. Hence, the counting width of μ^p is also $\Omega(n)$.

Suppose then that $\mathbb{F}$ is a field of characteristic $c \neq 2$ and that there is a family of symmetric arithmetic circuits over $\mathbb{F}$ of orbit size $s = \mathcal{O}(2^{n^{1-\epsilon}})$ computing $\{\mathrm{PERM}_n\}$. Then, it follows from Corollary 14 that the counting width of μ^c is at most $k = \mathcal{O}(\frac{\log s}{\log n}) = \mathcal{O}(n^{1-\epsilon})$, giving a contradiction. ◀

The construction used to prove Theorem 15 is an adaptation of a standard construction by Cai, Fürer and Immerman [8] which gives non-isomorphic graphs X and Y with $X \equiv^k Y$ for arbitrary k (see also [10]). We tweak it somewhat to ensure that both graphs have perfect matchings (indeed, they are both balanced bipartite graphs). The main innovation is in the analysis of the number of perfect matchings the graphs contain.

Gadgets. In what follows, $G = (V, E)$ is always a 3-regular 2-connected graph. From this, we first define a graph $X(G)$. The vertex set of $X(G)$ contains, for each edge $e \in E$, two vertices that we denote e_0 and e_1. For each vertex $v \in V$ with incident edges f, g and h, $X(G)$ contains five vertices. One of these we call the *balance vertex* and denote v_b. The other four are called *inner vertices* and there is one v_S, for each subset $S \subseteq \{f, g, h\}$ of even size. For each $v \in V$, the neighbours of v_b are exactly the four vertices of the form v_S. Moreover, for each $e \in \{f, g, h\}$, $X(G)$ contains the edge $\{e_1, v_S\}$ if $e \in S$ and the edge $\{e_0, v_S\}$ otherwise. There are no other edges in $X(G)$.

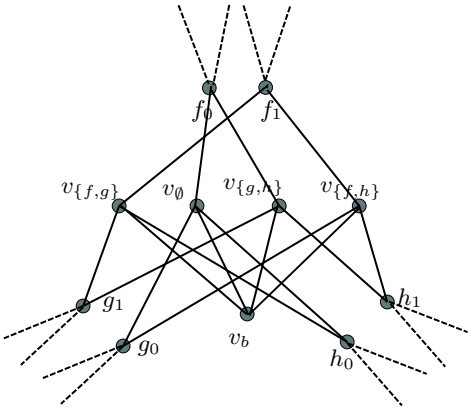

Figure 1 A gadget in $X(G)$ corresponding to vertex v with incident edges f, g, h.

The construction of $X(G)$ from G essentially replaces each vertex v with incident edges f, g and h with the gadget depicted in Figure 1, where the dashed lines indicate edges whose endpoints are in other gadgets. The vertices e_0, e_1 for each $e \in \{f, g, h\}$ are shared with neighbouring gadgets.

For any fixed vertex $x \in V$ with incident edges f, g, h, the graph $\tilde{X}_x(G)$ is obtained by modifying the construction of $X(G)$ so that, for the one vertex x, the gadget contains inner vertices x_S for subsets $S \subseteq \{f, g, h\}$ of odd size. Again, for each $e \in \{f, g, h\}$, $X(G)$ contains the edge $\{e_1, v_S\}$ if $e \in S$ and the edge $\{e_0, v_S\}$ otherwise.

If we remove the balance vertices v_b, the graphs $X(G)$ and $\tilde{X}_x(G)$ are essentially the Cai-Fürer-Immerman (CFI) graphs associated with G. The balance vertex v_b is adjacent to all the inner vertices associated with v and so does not alter the automorphism structure of $X(G)$ (or $\tilde{X}_x(G)$) at all. Nor do these vertices alter any other essential properties of the CFI construction. In particular, since G is connected, we have the following lemma.

▶ **Lemma 16.** *For any $x, y \in V$, $\tilde{X}_x(G)$ and $\tilde{X}_y(G)$ are isomorphic.*

With this in mind, we refer simply to the graph $\tilde{X}(G)$ to mean a graph $\tilde{X}_x(G)$ for some fixed x, and we refer to x as the special vertex of G.

By known properties of the CFI construction, we also have the following (see [10, Theorem 3]).

▶ **Lemma 17.** *If the treewidth of G is greater than k, then $X(G) \equiv^k \tilde{X}(G)$.*

The purpose of the balance vertices is to change the structure of the perfect matchings. Indeed, if we let CFI(G) denote the subgraph of $X(G)$ that excludes the balance vertices, it is easily seen that this contains no perfect matchings. It is a bipartite graph where one part contains the $4|V|$ inner vertices and the other part contains the $2|E| = 3|V|$ edge vertices and so no perfect matching is possible. But, $X(G)$ is a bipartite graph where in one part we have the $4|V|$ inner vertices and in the other the $3|V|$ edge vertices along with the $|V|$ balance vertices. In short, this is a 4-regular bipartite graph and so contains perfect matchings. We next analyse the structure of the set of such perfect matchings. In particular, we show that $X(G)$ and $\tilde{X}(G)$ contain different numbers of perfect matchings.

In the sequel, we write X to denote either one of the graphs $X(G)$ or $\tilde{X}(G)$, $V(X)$ to denote its vertices and $E(X)$ to denote its edges. We continue to use V and E for the vertices and edges of G. Also, for each $v \in V$, we write I_v to denote the set of four inner vertices in X associated with v.

Non-Uniform Matchings. Let $M \subseteq E(X)$ be a perfect matching in X. For each $v \in V$ and $e \in E$ incident on v, we define the projection $p^M(v, e)$ of M on (v, e) to be the value in $\{0, 1, 2\}$ which is the number of edges between $\{e_0, e_1\}$ and I_v that are included in M. These satisfy the following equations:

$p(u, e) + p(v, e) = 2$ for each edge $e = \{u, v\} \in E$; and

$p(v, f) + p(v, g) + p(v, h) = 3$ for each vertex $v \in V$ with incident edges f, g, h.

The first of these holds because M must include exactly one edge incident on each of e_0 and e_1. The second holds because M must include an edge between v_b and one vertex of I_v. Thus, the three remaining vertices in I_v must be matched with vertices among $f_0, f_1, g_0, g_1, h_0, h_1$.

One solution to the set of equations is obtained by taking the constant projection $p^M(v, e) = 1$ for all such pairs (v, e). Say that a matching M is uniform if $p^M(v, e) = 1$ everywhere and non-uniform otherwise.

▶ **Lemma 18.** *The number of non-uniform matchings in $X(G)$ is the same as in $\tilde{X}(G)$.*

Proof. It suffices to prove that for any non-constant projection p, the number of matchings M with $p^M = p$ is the same for both $X(G)$ and $\tilde{X}(G)$. For then, taking the sum over all possible projections gives the result. So, let p be a non-constant projection. Then, for some edge $e = \{u, v\} \in E$, we have $p(u, e) = 2$ and $p(v, e) = 0$. Then, let $X(G)^-$ and $\tilde{X}(G)^-$ be the subgraphs of $X(G)$ and $\tilde{X}(G)$ respectively obtained by removing the edges between $\{e_0, e_1\}$ and I_v. It is clear that any matching M in $X(G)$ with $p^M = p$ is also a perfect matching in $X(G)^-$, and similarly for $\tilde{X}(G)$. However, $X(G)^-$ and $\tilde{X}(G)^-$ are isomorphic. This follows by an argument analogous to the proof of Lemma 16. Since G is 2-connected, there is a path p from u to the special vertex x that does not involve the edge e. We can then define an isomorphism from $X(G)$ to $\tilde{X}(G)$ by mapping e_0 to e_1, for each edge f on the path p, mapping f_0 to f_1 and extending this using the induced automorphisms of the gadgets corresponding to $v_1, \ldots, v_{t-1}$. We conclude that the numbers of such matchings are the same for both. ◀

Now, we aim to show that the number of uniform matchings of $X(G)$ is different to that of $\tilde{X}(G)$. For this, it is useful to first analyse the orientations of the underlying graph G.

Orientations. An *orientation* of G is a directed graph obtained from G by assigning to each edge $\{u, v\} \in E$ a direction. There are exactly $2^{|E|}$ distinct orientations of G. We say that a vertex $v \in V$ is *odd* with respect to an orientation $\overrightarrow{G}$ of G if it has an odd number of incoming directed edges and *even* otherwise. For an orientation $\overrightarrow{G}$ of G, we write $\mathrm{odd}(\overrightarrow{G})$ for the set of its odd vertices. We say that the orientiation $\overrightarrow{G}$ is *odd* if $|\mathrm{odd}(\overrightarrow{G})|$ is odd, and we say it is *even* otherwise. Since G is 3-regular, it is not hard to see that $|V|$ is always even and $|E|$ is even if, and only if, $|V|/2$ is. Moreover, in all orientations of G, $|\mathrm{odd}(\overrightarrow{G})| = |E| \pmod 2$.

▶ **Lemma 19.** *If $|V|/2$ is even, then all orientations of G are even. If $|V|/2$ is odd, then all orientations of G are odd.*

Proof. Note that since G is 3-regular, $3|V| = 2|E|$, so $|V|$ is always even. Moreover, $|V|/2$ is even if, and only if, $|E|$ is. For an orientation $\overrightarrow{G}$, let $\mathrm{in}(v)$ denote the number of edges incoming to the vertex v. Then, $|E| = \sum_v \mathrm{in}(v)$. But, $\sum_v \mathrm{in}(v) \equiv |\mathrm{odd}(\overrightarrow{G})| \pmod 2$. ◀

Thus, we say that a graph G is *odd* if $|E|$ is odd, and hence all orientations of G are odd, and G is *even* if $|E|$ is even and hence all orientations of G are even.

We can now quantify exactly, for any set $S \subseteq V$ the number of distinct orientations $\overrightarrow{G}$ with $\mathrm{odd}(\overrightarrow{G}) = S$. To do this, we first establish an auxilliary lemma.

▶ **Lemma 20.** *If $G = (V, E)$ is even, then for every set $S \subseteq V$ with $|S|$ even, there is an orientation $\overrightarrow{G}$ of G with $\mathrm{odd}(\overrightarrow{G}) = S$. Similarly if $G = (V, E)$ is odd, then for every set $S \subseteq V$ with $|S|$ odd, there is an orientation $\overrightarrow{G}$ of G with $\mathrm{odd}(\overrightarrow{G}) = S$.*

Proof. It suffices to show, for any set $S \subseteq V$ and any pair of vertices $u, v \in V$, if there is an orientation $\overrightarrow{G}$ of G with $\mathrm{odd}(\overrightarrow{G}) = S$, then there is also an orientation $\overrightarrow{G}'$ with $\mathrm{odd}(\overrightarrow{G}') = S \triangle \{u, v\}$. Now, consider any simple path from u to v in G and let $\overrightarrow{G}'$ be the orientation obtained from $\overrightarrow{G}$ by reversing the direction of every edge on this path. ◀

▶ **Lemma 21.** *For every set $S \subseteq V$ with $|S| = |E| \pmod 2$, there are exactly $2^{|V|/2+1}$ distinct orientations $\overrightarrow{G}$ with $\mathrm{odd}(\overrightarrow{G}) = S$.*

Proof. Let A be the $V \times E$ incidence matrix of the graph G. This defines a linear transformation from the vector space $\mathbb{F}_2^E$ to $\mathbb{F}_2^V$. The additive group of $\mathbb{F}_2^E$ has a natural action on the orientations of G: for a vector $\pi \in \mathbb{F}_2^E$, and an orientation $\overrightarrow{G}$, define $\pi\overrightarrow{G}$ to be the orientation obtained from $\overrightarrow{G}$ by changing the orientation of each edge e with $\pi(e) = 1$. Indeed, fixing one particular orientation $\overrightarrow{G}$, the action generates all orientations and gives a bijective correspondence between the vectors in $\mathbb{F}_2^E$ and the orientations of G. Similarly, the additive group of $\mathbb{F}_2^V$ has a natural action on the powerset of V: for a vector $\sigma \in \mathbb{F}_2^V$ and a set $S \subseteq V$, let σS be the set $S \triangle \{v \mid \sigma(v) = 1\}$. Again, for any fixed set S, this action generates all subsetes of V and gives a bijection between $\mathbb{F}_2^V$ and the powerset of V.

Then, it can be seen that $\mathrm{odd}(\pi\overrightarrow{G}) = (A\pi)\mathrm{odd}(\overrightarrow{G})$. Indeed, if $v \in V$ is a vertex with incident edges f, g, h, then $(A\pi)(v) = \pi(f)+\pi(g)+\pi(h) \pmod 2$. In other words $(A\pi)(v) = 1$ just in case the direction of an odd number of edges incident on v is flipped by π. Thus, the set of vertices $\{v \mid (A\pi)(v) = 1\}$ are exactly the ones that change from being odd to even or vice versa under the action of π, i.e. $\{v \mid (A\pi)(v) = 1\} = \mathrm{odd}(\overrightarrow{G}) \triangle \mathrm{odd}(\pi\overrightarrow{G})$ for any orientation $\overrightarrow{G}$.

Fixing a particular orientation $\overrightarrow{G}$, the action of $\mathbb{F}_2^E$ generates all orientation $\pi\overrightarrow{G}$, and A maps this to the collection of all sets $\text{odd}(\overrightarrow{G})\triangle\text{odd}(\pi\overrightarrow{G})$. Then, by Lemmas 19 and 20 the image of A consists of exactly the set of vectors with an even number of 1s. Hence, the image of A has dimension $|V|-1$ and so its kernel has size $2^{|E|}/2^{|V|-1}$. Since $|E| = 3|V|/2$, this is $2^{|V|/2+1}$. By linearity, the pre-image of any vector v in the image of A has exactly this size. Thus, for each even size set $T \subseteq V$, there are exactly $2^{|V|/2+1}$ vectors $\pi \in \mathbb{F}_2^E$ with $\text{odd}(\pi\overrightarrow{G}) = T\triangle\text{odd}(\overrightarrow{G})$. ◀

Matchings in Gadgets. Any uniform perfect matching M of X induces an orientation of G, which we denote $\overrightarrow{G}^M$: any edge $e = \{u, v\} \in E$ is oriented from u to v in $\overrightarrow{G}^M$ if M contains an edge between e_0 and a vertex in I_u and an edge between e_1 and a vertex in I_v.

Furthermore, every orientation arises from some perfect matching. To see this, consider again the gadget in Figure 1. This has eight subgraphs induced by taking the vertices $\{v_b\}\cup I_v$, together with exactly one vertex from each of the sets $\{f_0, f_1\}$, $\{g_0, g_1\}$ and $\{h_0, h_1\}$. We claim that each of these eight subgraphs contains a perfect matching. Indeed, it suffices to verify this for the two cases $S = I_v \cup \{v_b\} \cup \{f_0, g_0, h_0\}$ and $T = I_v \cup \{v_b\} \cup \{f_0, g_0, h_1\}$ as the other six are obtained from these by automorphisms of the gadget. In what follows, we also write S and T for the subgraphs of the gadget in Figure 1 induced by these sets. It is easily seen by inspection that S has exactly four perfect matchings and T has exactly two perfect matchings.

Hence, for any orientation $\overrightarrow{G}$, we get a matching $M \subseteq X$ with $\overrightarrow{G}^M = \overrightarrow{G}$ by choosing one matching from each gadget. To be precise, for each vertex $v \in V$, define the *relevant subgraph* of X at v to be the subgraph induced by $I_v \cup \{v_b\}$ along with the vertices e_1 for each edge e incoming at v in $\overrightarrow{G}$ and e_0 for each edge e outgoing at v in $\overrightarrow{G}$. In $X(G)$, the relevant subgraph at v is isomorphic to S if v is even in $\overrightarrow{G}$ and it is isomorphic to T if v is odd in $\overrightarrow{G}$. The same is true for all vertices in $\tilde{X}(G)$, apart from the special vertex x. For this one, the relevant subgraph is isomorphic to S if x is odd and to T if x is even. In either case, we get a perfect matching M with $\overrightarrow{G}^M = \overrightarrow{G}$ by independently choosing exactly one matching in each relevant subgraph. There are 4 such choices when the relevant subgraph is like S and 2 choices when it is like T.

Uniform Matchings. It follows that for any orientation $\overrightarrow{G}$ of G, the number of uniform perfect matchings M of $X(G)$ with $\overrightarrow{G}^M = \overrightarrow{G}$ is $2^{|\text{odd}(\overrightarrow{G})|}4^{|V|-|\text{odd}(\overrightarrow{G})|}$. The number of uniform perfect matchings in $\tilde{X}(G)$ depends on whether the special vertex x is odd in $\overrightarrow{G}$ or not. If it is, the number is $2^{|\text{odd}(\overrightarrow{G})|-1}4^{|V|-|\text{odd}(\overrightarrow{G})|+1}$ otherwise it is $2^{|\text{odd}(\overrightarrow{G})|+1}4^{|V|-|\text{odd}(\overrightarrow{G})|-1}$. Thus, if we denote the number of uniform perfect matchings in X by $\#MX$, then we have $\#MX(G) = \sum_{\overrightarrow{G}} 2^{|\text{odd}(\overrightarrow{G})|}4^{|V|-|\text{odd}(\overrightarrow{G})|}$ where the sum is over all orientations of G. Then, by Lemma 21, $\#MX(G) = 2^{|V|/2+1}\sum_{S\subseteq V\,:\,|S|\equiv|E|\pmod 2} 2^{|S|}4^{|V|-|S|}$. By the same token, $\#M\tilde{X}(G) = 2^{|V|/2+1}\sum_{S\subseteq V\,:\,|S|\not\equiv|E|\pmod 2} 2^{|S|}4^{|V|-|S|}$.

Finally, to show that $\#MX(G)$ and $\#M\tilde{X}(G)$ are different, let P_m denote the number $\sum_{S\subseteq[2m]:|S|\text{even}} 2^{|S|}4^{2m-|S|}$ and Q_m denote the number $\sum_{S\subseteq[2m]:|S|\text{odd}} 2^{|S|}4^{2m-|S|}$.

▶ **Lemma 22.** *For all $m \geq 1$, $P_m - Q_m = 4^m$.*

Proof. We prove this by induction on m. For $m = 1$, there are exactly two odd sized subsets and two even sized subsets of $[2m]$. So $P_m = 20$ and $Q_m = 16$. For larger values of m, we have the following identity, where S ranges over subsets of $[2m-2]$

$$\begin{aligned} P_m &= \sum_{|S|\text{even}} 2^{|S|}4^{2m-|S|} + 2\sum_{|S|\text{odd}} 2^{|S|+1}4^{2m-|S|-1} + \sum_{|S|\text{even}} 2^{|S|+2}4^{2m-|S|} \\ &= 16P_{m-1} + 16Q_{m-1} + 4P_{m-1} \\ &= 20P_{m-1} + 16Q_{m-1}. \end{aligned}$$

Here, in the first line, the first sum accounts for all even size subsets of [2m] that exclude the last two elements, the second one for those that include exactly one of the last two elements and the third sum for all that include the last two elements.

Similarly, we have

$$\begin{aligned} Q_m &= \sum_{|S|\text{odd}} 2^{|S|}4^{2m-|S|} + 2\sum_{|S|\text{even}} 2^{|S|+1}4^{2m-|S|-1} + \sum_{|S|\text{odd}} 2^{|S|+2}4^{2m-|S|} \\ &= 16Q_{m-1} + 16P_{m-1} + 4Q_{m-1} \\ &= 20Q_{m-1} + 16P_{m-1}. \end{aligned}$$

Thus, $P_m - Q_m = 4P_{m-1} - 4Q_{m-1}$. By, induction hypothesis, the right hand side is $4 \cdot 4^{m-1}$ and we're done. ◀

Proof of Theorem 15. By a standard expander graph construction (e.g. [1]), for any k, we can find a 3-regular graph G with treewidth at least k and $2n = \mathcal{O}(k)$ vertices. Then $X(G)$ and $\tilde{X}(G)$ both have $\mathcal{O}(k)$ vertices and by Lemma 17 we have $X(G) \equiv^k \tilde{X}(G)$. Moreover, $X(G)$ and $\tilde{X}(G)$ have the same number of non-uniform perfect matchings by Lemma 18. The number of uniform matchings is $2^{n+1}P_n$ in one case and $2^{n+1}Q_n$ in the other (which is which depends on whether n is even or odd). Either way, $|\mu(X(G)) - \mu(\tilde{X}(G))| = 2^{3n+1}$, which is a power of 2 as required. ◀

7 Concluding Discussion

We have introduced a novel restriction of arithmetic circuits, which is based on a natural notion of symmetry. On this basis, we have shown a fundamental difference between circuits for the determinant and the permanent. The former admits a description through polynomial-size symmetric circuits and the latter does not.

There are several ways in which our results could be tightened. The first would be to show the existence of polynomial-size circuits for computing the determinant over arbitrary fields. Our construction for fields of characteristic zero is based on Le Verrier's method, which does not easily transfer to other fields as it relies on division by arbitrarily large integers. There are general methods for simulating such division on small fields, but it is not immediately clear if they can be carried out symmetrically. However, there are many efficient ways of computing a determinant and it seems likely that some method that works on fields of positive characteristic could be implemented symmetrically. It should be noted, however, that Gaussian elimination is not such a method. Known results about the expressive power of fixed-point logic with counting (see, e.g. [9]) tell us that there is no polynomial-size family of symmetric circuits that can carry out Gaussian elimination. On the other hand, we do know that the determinant, even over finite fields, can be computed by exactly such a family of Boolean circuits, as shown by Holm [16]. It is when we restrict to *arithmetic* circuits, and also require symmetry, that the question is open.

The notions of symmetry used in our upper bound for the determinant and the lower bound for the permanent are slightly different. Essentially, we consider symmetric circuits for the determinant where we require that each *simultaneous* row and column permutation extend to an automorphism of the circuit, while for the permanent we require that each permutation generated by separate row and column permutations extend to an automorphism of the circuit. We could improve the result by showing that the lower bound for the permanent still holds even if we only require the circuits be symmetric with respect to simultaneous row and column permutations. We think this could be established by adapting our construction to analyse the counting width of the number of cycle covers of general graphs.

We could consider more general symmetries. For example, the determinant has other symmetries besides simultaneous row and column permutations. The construction we use already yields a circuit which is symmetric not only with respect to these but also transposition of rows and columns. We could consider a richer group that allowed for even permutations of the rows and columns. Could our upper bound be improved by constructing circuits for the determinant that are symmetric with respect to larger groups of permutations?

Finally, it is reasonable to think that there are polynomials in VP which do not admit polynomial-size symmetric arithmetic circuits, by analogy with the case of Boolean circuits. Can we give an explicit example of such a polynomial?

References

1. M. Ajtai. Recursive construction for 3-regular expanders. *Combinatorica*, 14:379–416, 1994.
2. M. Anderson and A. Dawar. On symmetric circuits and fixed-point logics. *Theory Comput. Syst.*, 60(3):521–551, 2017.
3. M. Anderson, A. Dawar, and B. Holm. Solving linear programs without breaking abstractions. *J. ACM*, 62, 2015.
4. V. Arvind, F. Fuhlbrück, J. Köbler, and O. Verbitsky. On Weisfeiler-Leman invariance: Subgraph counts and related graph properties. In *Fundamentals of Computation Theory – 22nd International Symposium, FCT 2019*, pages 111–125, 2019. doi:10.1007/978-3-030-25027-0_8.
5. A. Atserias, A. Dawar, and J. Ochremiak. On the power of symmetric linear programs. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, pages 1–13, 2019. doi:10.1109/LICS.2019.8785792.
6. W. Baur and V. Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22, 1983. doi:10.1016/0304-3975(83)90110-X.
7. A. Blass, Y. Gurevich, and S. Shelah. On polynomial time computation over unordered structures. *Journal of Symbolic Logic*, 67(3):1093–1125, 2002.
8. J-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.
9. A. Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2(1):8–21, 2015.
10. A. Dawar and D. Richerby. The power of counting logics on restricted classes of finite structures. In *CSL 2007:Computer Science Logic*, volume 4646 of *LNCS*, pages 84–98. Springer, 2007.
11. A. Dawar and P. Wang. Definability of semidefinite programming and Lasserre lower bounds for CSPs. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, 2017. doi:10.1109/LICS.2017.8005108.
12. A. Dawar and G. Wilsenach. Symmetric circuits for rank logic. In *27th EACSL Annual Conference on Computer Science Logic, CSL 2018*, pages 20:1–20:16, 2018.
13. L. Denenberg, Y. Gurevich, and S. Shelah. Definability by constant-depth polynomial-size circuits. *Information and Control*, 70:216–240, 1986.

14 M. A. Forbes, M. Kumar, and R. Saptharishi. Functional lower bounds for arithmetic circuits and connections to boolean circuit complexity. In *31st Conference on Computational Complexity, CCC 2016*, pages 33:1–33:19, 2016. doi:10.4230/LIPIcs.CCC.2016.33.

15 D. Grigoriev and M. Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 577–582, 1998. doi:10.1145/276698.276872.

16 B. Holm. *Descriptive Complexity of Linear Algebra*. PhD thesis, University of Cambridge, 2010.

17 M. Jerrum and M. Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29:874–897, 1982. doi:10.1145/322326.322341.

18 N. Kayal and R. Saptharishi. A selection of lower bounds for arithmetic circuits. In M. Agrawal and V. Arvind, editors, *Perspectives in Computational Complexity*. Birkhäuser Basel, 2014.

19 J.M. Landsberg and N. Ressayre. Permanent v. determinant: An exponential lower bound assuming symmetry. In *Proc. ACM Conference on Innovations in Theoretical Computer Science*, pages 29–35. ACM, 2016. doi:10.1145/2840728.2840735.

20 M. Otto. The logic of explicitly presentation-invariant circuits. In *Computer Science Logic, 10th International Workshop, CSL '96, Annual Conference of the EACSL*, pages 369–384, 1996.

21 A. Shpilka and A. Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10:1–27, 2001. doi:10.1007/PL00001609.

22 A. Shpilka and A. Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. doi:10.1561/0400000039.

23 L. G. Valiant. Completeness classes in algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing STOC*, pages 249–261, 1979. doi:10.1145/800135.804419.

24 H. Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 1999. doi:10.1007/978-3-662-03927-4.