THEORY OF COMPUTING, Volume 21 (14), 2025, pp. 1–32
www.theoryofcomputing.org

Symmetric Arithmetic Circuits

Anuj Dawar Gregory Wilsenach*

Received December 23, 2020; Revised July 27, 2023; Published December 27, 2025

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 simultaneous row and column permutations of the matrix. We establish unconditional exponential lower bounds on the size of any symmetric circuit for computing the permanent. In contrast, we show that there are polynomial-size symmetric circuits for computing the determinant over fields of characteristic zero.

1 Introduction

Valiant's conjecture [29], that VP ≠ VNP, is often referred to as the algebraic counterpart to the conjecture that P ≠ 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

A preliminary version of this paper appeared in the Proceedings of the 47th IEEE International Colloquium on Automata, Languages, and Programming (ICALP'20) [14].
*Research of both authors supported by EPSRC grant EP/S03238X/1.

ACM Classification: F.1.3, F.1.2

AMS Classification: 68Q17,68Q15,68Q19

Key words and phrases: arithmetic circuits, algebraic circuits, symmetric circuits, determinant, permanent, lower bounds, descriptive complexity, circuit complexity, matching

 DOI: 10.4086/toc.2025.v021a014

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 subexponential family of *monotone* circuits for the permanent. This was first shown for the field of real numbers [22] and a proof for general fields, with a suitably adapted notion of monotonicity is given in [23]. An exponential lower bound for the permanent is also known for *depth-3* arithmetic circuits [19] over all 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 general 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 distinguishes between the determinant and the permanent. That is to say, we are able to show exponential lower bounds on the size of any family of symmetric arithmetic circuits for computing the permanent, while establishing the existence of polynomial-size symmetric circuits for computing the determinant.

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, e. g., 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 a more restricted group of permutations that includes *simultaneous* permutations of the rows and columns. We consider similar notions of symmetry on circuits. We say that an arithmetic circuit C (seen as a labelled directed acyclic graph) that takes as input an $n \times n$ matrix of variables (i. e., has input gates labelled by x_{ij}, for $i, j \in [n]$) is *matrix symmetric* if the natural action of any $(\sigma, \pi) \in \mathbf{Sym}(n) \times \mathbf{Sym}(n)$ on the inputs (i. e., taking x_{ij} to $x_{\sigma(i)\pi(j)}$) extends to an automorphism of C. Similarly, we say that C is *square symmetric* if the natural action of any $\sigma \in \mathbf{Sym}(n)$ on its inputs (i. e., taking x_{ij} to $x_{\sigma(i)\sigma(j)}$) extends to an automorphism of C.

Our upper bound for the determinant is established for square-symmetric circuits over fields of characteristic 0, and we conjecture it holds for all characteristics. For the permanent we prove exponential lower bounds for square-symmetric circuits over fields of characteristic 0 and for matrix-symmetric circuits over all fields of characteristic other than two. On fields of characteristic two, of course, the permanent and the determinant coincide.

A similar notion of symmetry has been studied previously in the context of Boolean circuits for deciding graph properties, or properties of relational structures (see [17, 25, 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]. In particular, 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 [13]). 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 n, the size, i. e., the number of vertices, of the graph, 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 square-symmetric circuits computing μ implies a sublinear upper bound on its counting width. Hence, using the relationship between the permanent of the adjacency matrix of a graph Γ and the number of perfect matchings in Γ, we obtain our lower bound for the permanent for square-symmetric circuits over fields of characteristic zero by showing a linear lower bound on the counting width of $\mu(\Gamma)$—the number of perfect matchings in G. Indeed, showing the same for $(\mu(\Gamma) \bmod p)$ for every prime $p > 2$ allows us to obtain an exponential lower bound for matrix-symmetric circuits over any field of characteristic other than two.

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 [8]. For general graphs, it is known to be strictly greater than 2 but still bounded above by a constant [3].

Related work. While symmetric arithmetic circuits have not previously been studied, symmetric Boolean circuits have [17, 25, 2, 26]. We review some connections of this previous work with ours in Section 3. The power of symmetric circuits in the context of computing fully symmetric polynomials can be established from a recent result of Bläser and Jindal, which we review further in Section 3.3.

Landsberg and Ressayre [24] 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. We give a detailed comparison of their results with ours in Section 7.4. In summary, their result does not yield any lower bounds for symmetric circuits in the sense we consider. On the other hand, while we cannot derive their results from ours, we can derive a lower bound on the determinantal complexity of the permament from our result which is incomparable with theirs.

A preliminary version of this paper was presented at the ICALP 2020 conference [14]. The lower bounds we present for the permanent here strengthen the results announced in the conference in two significant ways. First, we give the lower bound for *square-symmetric* circuits, that is, the same symmetry group for which we can prove an upper bound for the determinant. In the conference paper the lower bound was only given for the stronger condition of matrix-symmetric circuits. Secondly, the lower bound given in the conference was *nearly* exponential: we showed that there is no circuit of size $O(2^{n^{1-\epsilon}})$ for any positive ϵ. Here we improve this to show that there is no symmetric circuit for the permanent of size $2^{o(n)}$.

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 bounds for the permanent is established in Sections 5, 6, and 7. The first of these gives the symmetry-preserving translation from arithmetic circuits to threshold circuits, the second establishes an upper bound on the counting width of a family in terms of the (orbit) size of the family of circuits deciding it, and the third gives the main construction proving the linear lower bounds 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 Groups

For a set X, $\mathbf{Sym}(X)$ is the symmetric group on X. For $n \in \mathbb{N}$ we write $\mathbf{Sym}_n$ to abbreviate $\mathbf{Sym}([n])$. The *sign* of a permutation $\sigma \in \mathbf{Sym}(X)$ is defined so that if σ is even $\mathrm{sgn}(\sigma) = 1$ and otherwise $\mathrm{sgn}(\sigma) = -1$.

Let G be a group acting on a set X. We denote this as a left action, i. e., σx for $\sigma \in G$, $x \in X$. The action extends in a natural way to powers of X. So, for $(x, y) \in X \times X$, $\sigma(x, y) = (\sigma x, \sigma y)$. It also extends to the powerset of X and functions on X as follows. The action of G on $\mathcal{P}(X)$ is defined for $\sigma \in G$ and $S \in \mathcal{P}(X)$ by $\sigma S = \{\sigma x : x \in S\}$. For Y any set, the action of G on Y^X is defined for $\sigma \in G$ and $f \in Y^X$ by $(\sigma f)(x) = f(\sigma^{-1}x)$ for all $x \in X$. We refer to all of these as the *natural action* of G on the relevant set.

Let $X = \prod_{i \in I} X_i$ and for each $i \in I$ let G_i be a group acting on X_i. The action of the direct product $G := \prod_{i \in I} G_i$ on X is defined for $x = (x_i)_{i \in I} \in X$ and $\sigma = (\sigma_i)_{i \in I} \in G$ by $\sigma x = (\sigma_i x_i)_{i \in I}$. If instead $X = \biguplus_{i \in I} X_i$ then the action of G on X is defined for $x \in X$ and $\sigma = (\sigma_i)_{i \in I} \in G$ such that if $x \in X_i$ then $\sigma x = \sigma_i x$. Again, we refer to either of these as the *natural action* of G on X.

Let G be a group acting on a set X. Let $S \subseteq X$. Let $\mathbf{Stab}_G(S) := \{\sigma \in G : \forall x \in S\ \sigma x = x\}$ denote the *(pointwise) stabiliser* of S.

2.2 Fields and linear algebra

Let A and B be finite non-empty sets. An $A \times B$ *matrix* with entries in X is a function $M : A \times B \to X$. For $a \in A$, $b \in B$ let $M_{ab} = M(a, b)$. We recover the more familiar notion of an $m \times n$ matrix with rows and columns indexed by ordered sets by taking $A = [m]$ and $B = [n]$.

The permanent of a matrix is invariant under taking row and column permutations, while the determinant and trace are invariant under taking *simultaneous* row and column permutations. With this observation in mind, we define these three functions for unordered matrices. Let R be a commutative ring and $M : A \times B \to R$ be a matrix where $|A| = |B|$. Let $\mathrm{Bij}(A, B)$ be the set of bijections from A to B. The *permanent* of M over R is $\mathrm{perm}_R(M) = \sum_{\sigma \in \mathrm{Bij}(A,B)} \prod_{a \in A} M_{a\sigma(a)}$. Suppose $A = B$. The *determinant* of M over R is $\det_R(M) = \sum_{\sigma \in \mathbf{Sym}(A)} \mathrm{sgn}(\sigma) \prod_{a \in A} M_{a\sigma(a)}$. The *trace* of M over R is $\mathrm{Tr}_R(M) = \sum_{a \in A} M_{aa}$. In all three cases we omit reference to the ring R when it is obvious from context or otherwise irrelevant.

We always use $\mathbb{F}$ to denote a field and $\mathrm{char}(\mathbb{F})$ to denote the characteristic of $\mathbb{F}$. For any prime power q we write $\mathbb{F}_q$ for the finite field of order q. We are often interested in polynomials defined over a set of variables X with a natural matrix structure, i. e., $X = \{x_{ab} : a \in A, b \in B\}$. We identify X with this matrix. We also identify any function of the form $f : X \to Y$ with the $A \times B$ matrix with entries in Y defined by replacing each x_{ab} with $f(x_{ab})$.

For $n \in \mathbb{N}$ let $X_n = \{x_{ij} : i, j \in [n]\}$. Let $\mathrm{PERM}_n := \mathrm{perm}(X_n)$ and $\mathrm{DET}_n := \det(X_n)$. In other words, PERM_n is the formal polynomial defined by taking the permanent of an $n \times n$ matrix with (i, j)th entry x_{ij}, and similarly for the determinant. We write $\{\mathrm{PERM}_n\}$ to abbreviate $\{\mathrm{PERM}_n : n \in \mathbb{N}\}$ and $\{\mathrm{DET}_n\}$ to abbreviate $\{\mathrm{DET}_n : n \in \mathbb{N}\}$.

2.3 Graphs, matrices and matchings

Given a graph $\Gamma = (V, E)$, the adjacency matrix A_G of Γ is the $V \times V$ $\{0, 1\}$-matrix with $A_\Gamma(u, v) = 1$ if, and only if, $\{u, v\} \in E$. If Γ is bipartite, with bipartition $V = A \cup B$, then the *biadjacency matrix* B_Γ of Γ is the $A \times B$ $\{0, 1\}$-matrix with $B_\Gamma(u, v) = 1$ if, and only if, $\{u, v\} \in E$.

It is well known that for a bipartite graph Γ, $\mathrm{perm}(B_\Gamma)$ over any field of characteristic zero counts the number of perfect matchings in Γ [20] and for prime p, $\mathrm{perm}_{\mathbb{F}}(B_\Gamma)$ for a field $\mathbb{F}$ of characteristic p counts the number of perfect matchings in Γ modulo p. For bipartite Γ, A_Γ is a block anti-diagonal matrix with two blocks corresponding to B_Γ and B_Γ^T and $\mathrm{perm}(A_\Gamma) = \mathrm{perm}(B_\Gamma)^2$.

2.4 Counting width

For any $k \in \mathbb{N}$, the k-dimensional Weisfeiler–Leman equivalence (see [9]), denoted $\equiv^k$ is an equivalence relation on graphs that provides an approximation of isomorphism in the sense that for isomorphic graphs Γ and Δ, we have $\Gamma \equiv^k \Delta$ for all k. Increasing values of k give finer relations, so $\Gamma \equiv^{k+1} \Delta$ implies $\Gamma \equiv^k \Delta$ for all k. The equivalence relation is decidable in time $n^{O(k)}$, where n is the size of the graphs. If $k \geq n$, then $\Gamma \equiv^k \Delta$ implies that Γ and Δ are isomorphic. The Weisfeiler–Leman equivalences have been widely studied and they have many equivalent characterizations in combinatorics, logic, algebra and linear optimization. One particularly useful characterization in terms of logic (see [9]) is that $\Gamma \equiv^k \Delta$ if, and only if, Γ and Δ cannot be distinguished by any formula of first-order logic with *counting quantifiers* using at most $k+1$ distinct variables. This has been used to establish inexpressibility results in various counting logics and motivates the notion of *counting width*.

A *graph parameter* is a function μ from graphs to a set X which is isomorphism invariant. That is to say, $\mu(\Gamma) = \mu(\Delta)$ whenever Γ and Δ are isomorphic graphs. Most commonly, X is the set $\mathbb{N}$ and examples of such graph parameters are the chromatic number, the number of connected components or the number of perfect matchings. We can also let X be a field $\mathbb{F}$ and let $\mu(\Gamma)$ denote the permanent (over $\mathbb{F}$) of the adjacency matrix of Γ. When $X = \{0, 1\}$ we identify μ with the class of graphs for which it is the indicator function. In this case, we also call it a *graph property*.

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 on graphs with at most n vertices. This motivates the definition.

Definition 2.1. 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 $\Gamma, \Delta \in C$ of size at most n, if $\Gamma \equiv^k \Delta$, then $\mu(\Gamma) = \mu(\Delta)$.

The notion of counting width for classes of graphs was introduced in [13], which we here extend to graph parameters. Note that for any graph parameter $\nu(n) \leq n$, since any non-isomorphic graphs on n vertices can always be distinguished in $\equiv^n$.

Cai, Fürer and Immerman [9] first showed that there is no fixed k for which $\equiv^k$ coincides with isomorphism. Indeed, in our terminology, they construct a graph property with counting width $\Omega(n)$. Since then, many graph properties have been shown to have linear counting width, including Hamiltonicity and 3-colourability (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 6.4).

Our lower bound construction in Section 7 is based on the graphs constructed by Cai et al. [9]. While we review some of the details of the construction in Section 7, a reader unfamiliar with the construction may wish to consult a more detailed introduction. The original construction can be found in [9] and a version closer to what we use is given in [12].

2.5 Circuits

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

Definition 2.2 (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 $b \in \mathbb{B}$ such that the arity of the basis element b matches the in-degree of the vertex.

Note that, in the examples we consider, the elements of the basis often do not have fixed arity. That is, we are considering unbounded fan-in circuits where gates such as AND, OR, $+$, $\times$ can take any number of inputs. The one exception is the NOT gate.

Let $C = (D, W)$, where $W \subset D \times D$, be a circuit with constants K. We call the elements of D *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 $g, h \in D$ we say that h is a *child* of g if $(h, g) \in W$. We write $\mathrm{child}(g)$ to denote the set of children of g. We write C_g to denote the subcircuit of C rooted at g. The *size* of C is the number of gates in it. Unless otherwise stated we always assume a circuit has exactly one output gate. We also assume that distinct input gates in a circuit have distinct labels.

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* $\mathbb{B}_{\mathrm{std}}$ contains the functions $\wedge$, $\vee$, and $\neg$. The *threshold basis* $\mathbb{B}_t$ is the union of $\mathbb{B}_{\mathrm{std}}$ 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 $\vec{x}$ is 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 3.2). This is the case for the arithmetic functions $+$ and $\times$ and for all of the Boolean functions in $\mathbb{B}_t$ and $\mathbb{B}_{\mathrm{std}}$. 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 [28] and for Boolean circuits see [31].

By a standard translation (see [30]), arithmetic circuits with unbounded fan-in can be mapped to equivalent arithmetic circuits with constant fan-in with only a polynomial blowup in size and a logarithmic blowup in depth. This means that so long as we are interested in bounds on circuit size up to polynomial factors we may assume without a loss of generality that all gates have fan-in two. This assumption simplifies the analysis of these circuits and in many cases authors simply define arithmetic circuits to have internal gates with fan-in two (e. g., [28]). In this paper we are interested in symmetric arithmetic circuits and the standard translation does not preserve symmetry. As such, we cannot assume a bound on fan-in without a loss of generality and for this reason we define arithmetic circuits so as to allow for unbounded fan-in.

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

Definition 3.1. 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, which we now define and illustrate with examples.

Definition 3.2. If $G = \mathbf{Sym}(X)$, we call a G-symmetric function $F : K^X \to K$, *fully symmetric*.

Examples of fully symmetric functions are those that appear as labels of gates in a circuit, including $+$, $\times$, $\wedge$, $\vee$ and $t_{\geq k}$.

Definition 3.3. If $G = \mathbf{Sym}(X) \times \mathbf{Sym}(Y)$ and $F : K^{X \times Y} \to K$ is G-symmetric with the natural action of G on $X \times Y$, then we say F is *matrix symmetric*.

matrix-symmetric functions are those where the input is naturally seen as a matrix with the result 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 does satisfy a more restricted notion of symmetry that we define next.

Definition 3.4. If $G = \mathbf{Sym}(X)$ and $F : K^{X \times X} \to K$ is G-symmetric with the natural action of G on $X \times X$, then we say that F is *square symmetric*.

The determinant is one example of a square-symmetric function. However, as the determinant of a matrix is also invariant under the operation of transposing the matrix, we also consider this variation. To be precise, let $\sigma_t \in \mathbf{Sym}(X \times X)$ be the permutation that takes (x, y) to (y, x) for all $x, y \in X$. Let G_{sqr} be the diagonal of $\mathbf{Sym}(X) \times \mathbf{Sym}(X)$ (i. e., the image of $\mathbf{Sym}(X)$ in its natural action on $X \times X$). We write G_{tsp} for the group generated by $G_{\text{sqr}} \cup \{\sigma_t\}$. We say that a G_{tsp}-symmetric function is *transpose symmetric*.

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

Definition 3.5. If $X = \biguplus_{i \in I} X_i$, $G = \prod_{i \in I} \mathbf{Sym}(X_i)$, and $F : K^X \to K$ is G-symmetric with respect to the natural action of G on X, we say that F is *partition symmetric*.

In Section 5, we consider a generalization of circuits to the case where the labels in the basis are not necessarily fully symmetric functions, but they are still partition symmetric. The structure of such a circuit can not be described simply as a DAG, but requires additional labels on wires, as we shall see.

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. Thus, the inputs to a Boolean circuit C might be labelled by pairs of elements (x, y) where $x, y \in V$ and we require the output of C to be invariant under a permutation of V applied to the inputs. In short, the function computed by C is square-symmetric. A generalization to arbitrary symmetry groups was also defined by Rossman [26] who showed a lower bound for the parity function for formulas that are G-symmetric for subgroups G of $\mathbb{Z}_2^n$. Here, we consider circuits that are symmetric with respect 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 3.6 (Circuit Automorphism)**.** Let $C = (D, W)$ be a circuit over the basis $\mathbb{B}$ with variables X and constants K. For $\sigma \in \mathbf{Sym}(X)$, we say that a bijection $\pi : D \to D$ 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 πg.

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 3.7 (Symmetric Circuit)**.** Let C be a circuit with variables X and $G \leq \mathbf{Sym}_X$. We say that C is *G-symmetric* if the action of every $\sigma \in G$ on X extends to an automorphism of C. We say that C is *strictly G-symmetric* if the only automorphisms of C are those extending a permutation in G.

It is easy see that if a circuit is G-symmetric then it computes a G-symmetric polynomial (and hence function). We sometimes omit mention of G when it is obvious from context. 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 extending some permutation in G 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.

We use the same terminology for symmetric circuits as for symmetric functions. That is, if a circuit C with variables $X \times Y$ is $\mathbf{Sym}_X \times \mathbf{Sym}_Y$-symmetric we say that C is matrix symmetric. We similarly define square-symmetric circuits, transpose-symmetric circuits and partition symmetric circuits.

Though symmetric arithmetic circuits have not previously been studied, symmetric Boolean circuits have [17, 25, 2, 26]. It is known that polynomial-size square-symmetric threshold circuits are more powerful than polynomial-size square-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 [16] 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 signficant, particularly 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 $\sigma \in G$. It is clear that a G-symmetric polynomial determines a G-symmetric function. 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.

What are usually called the *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]. It is worth remarking that the matching upper bound is achieved by a fully symmetric circuit. Thus, at least in this case, there is no gain to be made by breaking symmetries in the circuit. Similarly, we have tight quadratic upper and lower bounds for depth-3 circuits for the elementary symmetric polynomials $\sum_{S\subseteq[n]:|S|=k} \prod_{i\in S} x_i$ over infinite fields [27]. The upper bound is obtained by the interpolation method and it can be seen that this is achieved by fully symmetric circuits. To be precise, the polynomial is computed as the coefficient of t^{n-k} in $\prod_{i=1}^n (t + x_i)$, which is obtained by interpolation from computing $\prod_{i=1}^n (t + x_i)$ at $n + 1$ distinct values of t. Note that, for any fixed constant t, $\prod_{i=1}^n (t + x_i)$ is given by a fully symmetric circuit of size $O(n)$, and these can be combined to get the interpolant. The resulting circuit is still fully symmetric since a permutation of the variables x_i fixes the polynomial $\prod_{i=1}^n (t + x_i)$.

Indeed, we can say something more general about fully symmetric polynomials. If any such polynomial $f \in \mathbb{C}[x_1, ..., x_n]$ has a circuit of size polynomial in n then it has a $\mathbf{Sym}_n$-circuit of size polynomial in n. This follows from a result of Bläser and Jindal [7] who establish that for any fully symmetric polynomial $f \in \mathbb{C}[x_1, \dots, x_n]$ which has a polynomial-size circuit there exists a witness $g \in \mathbb{C}[y_1, \dots, y_n]$ computable via an arithmetic circuit of size polynomial in n such that $f = g(e_1, \dots, e_n)$, where the e_i are the elementary symmetric polynomials. To see why this implies the result, observe that if f is a fully symmetric polynomial and g is the corresponding witness computable via a polynomial size circuit C, and E_i are the (fully symmetric and polynomial size) circuits computing the polynomials e_i, then we can build a circuit for f by replacing each input y_i in C with the output gate of E_i. The resultant circuit is symmetric since any permutation on the input gates fixes the output gate of each E_i.

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 matrix symmetric, and it yields a matrix-symmetric circuit of size $O(2^n n^2)$. Our main result, Theorem 7.1, gives us a near matching lower bound on the size of matrix-symmetric circuits (or even square-symmetric circuits) for expressing PERM_n.

A G-symmetric circuit C expressing a polynomial p is also a G-symmetric circuit computing the function determined by p. In establishing our upper bound for the determinant, we show the existence of small transpose-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 square-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 [18].

4 An upper bound for the determinant

In this section we show that for any field $\mathbb{F}$ with characteristic zero there is a polynomial-size family of transpose-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 [21] for more detail.

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

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

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of M (in a suitable extension of $\mathbb{F}$), 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

$$p_i = \frac{1}{i}[p_{i-1}s_1 - p_{i-2}s_2 + \dots \pm s_i],$$

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 transpose-symmetric we begin with some permutation $\sigma \in G_{\mathrm{tsp}}$ and show that σ extends to an automorphism of the circuit. We construct this automorphism layer by layer. We map each gate computing some (i, j) entry of M^k to the gate computing the $\sigma(i, j)$ entry of M^k. We fix each gate computing some s_k. Since each gate computing some p_l uses only the gates computing $s_1, \ldots, s_k$ and a constant gate computing 1, we can also fix each of these gates. We now present this argument formally.

Theorem 4.1. *For $\mathbb{F}$ a field of characteristic 0, there exists a family of transpose-symmetric arithmetic circuits $(\Phi_n)_{n\in\mathbb{N}}$ over $\mathbb{F}$ computing $\{\mathrm{DET}_n\}$ for which the function $n \mapsto \Phi_n$ is computable in time $O(n^4)$.*

Proof. Let $n \in \mathbb{N}$ and let $X = (x_{ij})_{i,j\in I}$ be an $I\times I$ matrix of variables, for an index set I with $|I| = n$. We now describe an implementation of Le Verrier's method for $I \times I$ matrices as arithmetic circuit Φ_n over the set of variables X. We construct this circuit as follows.

- For each $k \in [n]$ we include a family of gates intended to compute the entries in the kth power of the matrix X. For each $i, j \in I$ we include a gate $(k; i, j)$ intended to compute $(X^k)_{ij}$. Let $(\Phi_n)_{(2;i,j)} = \sum_{a\in I} x_{ia}x_{aj}$ and for all $k > 2$, $(\Phi_n)_{(k;i,j)} = \sum_{a\in I}(\Phi_n)_{(k-1;i,a)}x_{a,j}$.
- For each $k \in [n]$ we include a gate (Tr, k) intended to compute the trace of X^k. Let $(\Phi_n)_{(\mathrm{Tr},1)} = \sum_{a\in I} x_{a,a}$ and for $k > 1$, $(\Phi_n)_{(\mathrm{Tr},k)} = \sum_{a\in I}(\Phi_n)_{(k;a,a)}$.
- For each $k \in [n]$ we include a gate (p, k) intended to compute the coefficient p_k in the characteristic polynomial. Let $(\Phi_n)_{(p,1)} = (\Phi_n)_{(\mathrm{Tr},1)}$ and for all $k > 1$ let

$$(\Phi_n)_{(p,k)} = \frac{1}{k}[(\Phi_n)_{(p,k-1)}(\Phi_n)_{(\mathrm{Tr},1)} - (\Phi_n)_{(p,k-2)}(\Phi_n)_{(\mathrm{Tr},2)} \\ + (\Phi_n)_{(p,k-3)}(\Phi_n)_{(\mathrm{Tr},3)} - \ldots \pm (\Phi_n)_{(\mathrm{Tr},k)}].$$

Let (p, n) be the output gate of Φ_n. It follows from the discussion preceding the statement of the theorem that (p, n) computes DET_n.

It remains to show that the circuit is transpose-symmetric. Let $\sigma \in \mathbf{Sym}(I)$. Let $\pi : \Phi_n \to \Phi_n$ be defined such that for each input gate labelled x_{ij} we have $\pi(x_{ij}) = x_{\sigma(i)\sigma(j)}$, for each gate of the form $(k; i, j)$ we have $\pi(k; i, j) = (k; \sigma(i), \sigma(j))$, and for every other gate g we have $\pi(g) = g$. It can be verified that π is a circuit automorphism extending σ. Similarly, if $\sigma_t \in \mathbf{Sym}(I \times I)$ is the transpose permutation, i. e., $\sigma_t(i, j) = (j, i)$, then we can extend it to an automorphism π_t of Φ_n by letting $\pi_t(k; i, j) = (k; j, i)$. It follows that Φ_n is a transpose-symmetric arithmetic circuit.

The circuit contains constant gates labelled by $-1, 0, 1, \frac{1}{2} \ldots \frac{1}{n}$. There are n^2 other input gates. Computing each gate $(k; i, j)$ uses $O(1 + n)$ gates (n products and 1 sum). Then, since there are

n^2 entries in each matrix and $n-1$ matrices to compute, the total number of gates needed to compute all of the $(k; i, j)$ gates is $O(n^2(n-1)(n+1))$. There are n additional gates required to compute all gates of the form (Tr, i). There are at most $n(2n-1)$ gates required to compute all gates of the form (p, i). It follows that the circuit is of size $O(n^4)$. The above description of the circuit Φ_n can be adapted to define an algorithm that computes the function $n \mapsto \Phi_n$ in time $O(n^4)$. ◻

Le Verrier's method explicitly involves multiplications by field elements $\frac{1}{k}$ for $k \in [n]$, and so cannot be directly applied to fields of positive characteristic. We conjecture that it is also possible to give square symmetric arithmetic circuits of polynomial size to compute the determinant over arbitrary fields. Indeed, there are many known algorithms that yield polynomial-size families of arithmetic circuits over fields of positive characteristic computing $\{\mathrm{DET}_n\}$. It seems likely that some of these could be implemented symmetrically.

5 From arithmetic to boolean circuits

In this section we establish the following symmetry and orbit-size preserving translation from arithmetic circuits to threshold circuits. Importantly, this translation does not preserve circuit size, which may grow exponentially.

Theorem 5.1. *Let G be a group acting on a set of variables X. Let Φ be a G-symmetric arithmetic circuit over a field $\mathbb{F}$ with variables X. Let $B \subseteq \mathbb{F}$ be finite. Then there is a G-symmetric threshold circuit C with variables X and $\boldsymbol{ORB}(C) = \boldsymbol{ORB}(\Phi)$, such that for all $M \in \{0,1\}^X$ we have $C[M] = 1$ if, and only if, $\Phi[M] \in B$.*

We use Theorem 5.1 in Section 7 to transfer a lower bound on threshold circuits to arithmetic circuits, a crucial step in establishing our lower bound for the permanent. This lower bound relies on the preservation of orbit-size in Theorem 5.1, and the connection between orbit-size and counting width.

We prove Theorem 5.1 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 Boolean threshold circuit.

To enable this second step, we first show that each partition symmetric function can be computed by a rigid strictly symmetric threshold circuit. The proof of this 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 5.2. *Let F be a partition symmetric function. There exists a rigid strictly partition 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 D_q, one for each $q \in Q$. Each D_q is a one-input, one-output circuit computing the identity function. For example, D_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 D_q^a of the gadget D_q.

Clearly $C(F)$ computes F. We now argue $C(F)$ is rigid and strictly partition symmetric. To see that it is partition 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 D_q^a to $D_q^{\sigma a}$ 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 D_q and $D_{q'}$ 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 $c \in \mathbb{F}$. Formally, we define for any $h \in \{0,1\}^A$ the functions $+^A_{Q,c} : \{0,1\}^A \to \{0,1\}$ and $\times^A_{Q,c} : \{0,1\}^A \to \{0,1\}$ 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 5.1 by first defining for a given G-symmetric arithmetic circuit a corresponding G-symmetric Boolean 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 5.1. We associate with each $v \in \Phi$ a finite set $Q_v \subseteq \mathbb{F}$ such that for any assignment of 0-1 values to the inputs, $M \in \{0,1\}^X$, we have $\Phi[M](v) \in Q_v$. This can be defined by induction on the structure of Φ: If v is an input gate, $Q_v = \{0,1\}$; and if v is an $\odot$-gate for $\odot \in \{+,\times\}$ with children $u_1,\dots,u_t$ we let $Q_v = \{a_1 \odot \cdots \odot a_t \mid a_i \in Q_{u_i}\}$. 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 a 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 $Q = \bigcup_{u \in \text{child}(v)} Q_u$. 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 G-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 G-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 $u \in \mathbf{Orb}(v)$. Since π' preserves the labelling on the gates in D, a simple induction on the depth of the gate in the circuit shows 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 $Q = \bigcup_{u \in \mathrm{child}(v)} Q_u$, for $q \in Q$, $A_q = \{u \in \mathrm{child}(v) : q \in Q_u\}$, and $A = \biguplus_{q \in Q} A_q$. Then

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

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 $F \in \mathbb{B}^{\mathbb{F}}_{\mathrm{arth}}$ with the rigid strictly G-symmetric threshold circuit $C(F)$ computing F defined in Lemma 5.2. C computes the same function as D. We now argue that $\mathbf{ORB}(C) = \mathbf{ORB}(D)$. Suppose that some gate g in $C(F)$ corresponding to a gate (v, c) in D is mapped by an automorphism of C to a gate g' in $C(F')$ corresponding to (v', c') in D. Since each $C(F)$ has a unique OR gate, it must be the case that the OR gate in $C(F)$ then maps to the OR gate in $C(F')$ and so we have an isomorphism between $C(F)$ and $C(F')$. The fact that $C(F)$ is rigid and *strictly* partition symmetric ensures that the isomorphism respects the partition on the input and so the circuits compute the same function, i. e., $F = F'$. We can conclude that the only automorphisms of C are those that are obtained from automorphisms of D. Thus, $\mathbf{ORB}(C) = \mathbf{ORB}(D) = \mathbf{ORB}(\Phi)$. ◻

6 Supports and counting-width

Lower bounds that have been established for symmetric Boolean circuits are based on showing lower bounds on *supports* in such circuits. In this section, we review the connection between the orbit size of circuits, the size of supports and the counting width of graph parameters computed by such circuits. We improve on the known connection between support size and orbit size to show that it can be used to obtain exponential lower bounds. We begin by reviewing the definition of supports.

Definition 6.1. Let C be a rigid G-symmetric circuit with variables $\{x_{i,j} : i, j \in [n]\}$. We say that a set $S \subseteq [n]$ is a *support* of a gate g if $\mathbf{Stab}_G(S) \leq \mathbf{Stab}_G(g)$.

Let $\mathrm{sp}(g)$ be the minimum size of a support of a gate g. Let $\mathbf{SP}(C)$ be the maximum size of $\mathrm{sp}(g)$ for g a gate in C. We refer to $\mathbf{SP}(C)$ as the *support size* of C.

Upper bounds on the orbit size of a square-symmetric circuit yield upper bounds on its support size. Indeed, it was shown in [2, Theorem 4] that circuit families of size at most $s = O(2^{n^{1/3}})$ have supports of size at most $O(\frac{\log s}{\log n})$. This result was extended to orbit size

$s = O(2^{n^{1-\epsilon}})$ for arbitrary positive ϵ in [5, Theorem 1]. The result there is stated in terms of the *size* of the circuit rather than its orbit size. However, the proof easily yields the bound for orbit size. These results immediately yield that polynomial-size families of symmetric circuits have $O(1)$ support size. It also implies that a linear lower bound on support size yields a lower bound of $\Omega(2^{n^{1-\epsilon}})$ on orbit size. It was this relationship that was used to obtain lower bounds on the size of symmetirc circuits for the permanent in the early version of this paper [14]. Here we improve the lower bound by showing that a linear lower bound on support size implies an exponential lower bound on orbit size, in Theorem 6.3 below. First we recall the following theorem.

Theorem 6.2 ([16, Theorem 4.10])**.** *Let C be a rigid square-symmetric Boolean circuit with order $n > 8$. For every $1 \leq k \leq \frac{n}{4}$ if the maximum size of an orbit in C is bounded by $\binom{n}{k}$ then each gate in C has a support of size less than k.*

Theorem 6.2 should be understood as a restatement of [16, Theorem 4.10] using the language of this paper. In [16] we dealt with a more general notion of circuits where individual gates could be labelled by functions that are not fully symmetric. What are called circuits with *injective labels* and *unique extensions* in that paper, restricted to the circuits we consider here, are exactly the rigid circuits.

We now extract from Theorem 6.2 an asymptotic relationship between orbits and supports.

Theorem 6.3. *Let $(C_n)_{n\in\mathbb{N}}$ be a family of rigid square-symmetric Boolean circuits over the threshold basis. If $\mathbf{ORB}(C_n) = 2^{o(n)}$ then $\mathbf{SP}(C_n) = o(n)$.*

Proof. Let k be the least value such that $\mathbf{ORB}(C_n) \leq \binom{n}{k}$. By the assumption that $\mathbf{ORB}(C_n) = 2^{o(n)}$, we have that k is $o(n)$. Indeed, otherwise there is a constant c with $0 < c < \frac{1}{2}$, such that $k-1 \geq cn$ for infinitely many n. And since $\binom{n}{l} \geq (\frac{n}{l})^l$ for all l it follows that $\binom{n}{k-1} \geq (\frac{n}{cn})^{cn} > 2^{cn}$. Since k is the *least* value such that $\mathbf{ORB}(C_n) \leq \binom{n}{k}$, it follows that $\mathbf{ORB}(C_n) \geq 2^{cn}$ for infinitely many n, contradicting the assumption that $\mathbf{ORB}(C_n) = 2^{o(n)}$.

From $k = o(n)$ it follows that for all large enough n, $k \leq \frac{n}{4}$ and so, by Theorem 6.2, $\mathbf{SP}(C_n) \leq k$ and therefore $\mathbf{SP}(C_n) = o(n)$ as required.

□

We now use the connection between support size and counting width established in [2]. Indeed, Theorem 6 of [2] asserts that a query on relational structures (e. g., a graph property) is decidable by a family of square-symmetric circuits with polynomial orbit size if, and only if, it is definable in $C^{\omega}_{\infty\omega}$, an infinitary logic with counting quantifiers. It is known that definability in this logic is the same as having bounded counting width. Moreover, the proof of [2, Theorem 6] establishes this by showing that a circuit of support size k translates into a formula with $O(k)$ variables. Thus, if a class of graphs $\mathcal{C}$ is decidable by a family of symmetric circuits $(C_n)_{n\in\mathbb{N}}$ with supports of size at most $k(n)$ then $\mathcal{C}$ has counting width $O(k)$. This, along with Theorem 6.3, immediately yields the following.

Theorem 6.4. *Let $\mathcal{C}$ be a class of graphs decidable by a family of square symmetric Boolean circuits with threshold gates and with orbit size $2^{o(n)}$, then $\mathcal{C}$ has counting width $o(n)$.*

The statement of Theorem 6.4 does not make mention of the rigiditity condition. This suffices, as from [2, Lemma 7] we have that any symmetric Boolean circuit over the threshold basis may be converted into an equivalent *rigid* symmetric circuit with only a linear increase in size. It is easily seen from the proof of that lemma that the conversion does not increase orbit size.

For a field $\mathbb{F}$ and a graph parameter μ with values in $\mathbb{F}$, we say that μ is computed by a family of $\mathbb{F}$-arithmetic circuits $(C_n)_{n\in\mathbb{N}}$ if the inputs to C_n are labelled by the variables x_{ij} for $i, j \in [n]$ and, given the adjacency matrix of a graph Γ on its inputs, C computes $\mu(\Gamma)$.

Corollary 6.5. *If a graph parameter μ is computed by a family of square-symmetric arithmetic circuits with orbit size $2^{o(n)}$, then the counting width of μ is $o(n)$.*

Proof. Let k be the counting width of μ. Then, by definition, we can find for each $n \in \mathbb{N}$ a pair of graphs Γ_n and Δ_n with at most n vertices such that $\Gamma_n \equiv^{k(n)-1} \Delta_n$ but $\mu(\Gamma_n) \neq \mu(\Delta_n)$. Let $B_n = \{\mu(\Gamma_n)\}$. Then, by Theorem 5.1 there is a family of square-symmetric circuits with threshold gates of orbit size $2^{o(n)}$ that decides for a graph Γ whether $\mu(\Gamma) \in B_n$. It follows from Theorem 6.4 that the counting width of this decision problem is $o(n)$. Since the counting width of this decision problem is, by choice of Γ_n, k, it follows that $k = o(n)$. ◻

7 A lower bound for the permanent

In this section we establish exponential lower bounds on the size of symmetric arithmetic circuits for the permanent. We state the result for square symmetric arithmetic circuits over fields of characteristic zero in Section 7.1 and show how it can be derived from a lower bound on the counting width of the number of perfect matchings. The bulk of the section is the construction in Section 7.2 establishing this counting width lower bound. In Section 7.3, we explain how the argument extends to fields of positive characteristic other than two, but at the expense of making the stronger requirement that the circuits are matrix symmetric. Finally, in Section 7.4 we make a comparison of our lower bounds with lower bounds on equivariant determinantal representations of the permanent.

7.1 Characteristic zero

Theorem 7.1. *There is no family of square-symmetric arithmetic circuits over any field $\mathbb{F}$ of characteristic 0 of orbit size $2^{o(n)}$ computing $\{\mathrm{PERM}_n\}$.*

Our proof of this result establishes something stronger. We actually show that there is no family of symmetric arithmetic circuits of orbit size $2^{o(n)}$ that computes the function $\mathrm{perm}(M)$ for matrices $M \in \mathbb{F}^{n\times n}$. Clearly, a circuit that computes the polynomial PERM_n also computes this function. Theorem 7.1 is proved by showing lower bounds on the counting widths of functions which determine the number of perfect matchings in a bipartite graph.

For a graph Γ let $\mu(\Gamma)$ be the number of perfect matchings in Γ. Our construction establishes a linear lower bound on the counting width of $\mu(\Gamma)$. Indeed, it also shows a linear lower bound on the counting width of $(\mu(\Gamma) \bmod p)$ for all odd values p. Thus, we aim to prove the following.

Theorem 7.2. *There is, for each $k \in \mathbb{N}$, a pair of balanced bipartite graphs X and Y with $O(k)$ vertices, such that $X \equiv^k Y$, and $\mu(X) - \mu(Y) = 2^l$ for some $l > 0$.*

Before giving the proof of Theorem 7.2 we show how Theorem 7.1 now follows.

Proof of Theorem 7.1. By Theorem 7.2, we have, for each k, a pair of graphs X and Y with $O(k)$ vertices such that $X \equiv^k Y$ and $\mu(X) \neq \mu(Y)$ and hence $\mu(X)^2 \neq \mu(Y)^2$. Thus, the counting width of μ^2 is $\Omega(n)$. Suppose that there is a family of square-symmetric arithmetic circuits over a field of characteristic 0 with orbit size $2^{o(n)}$ computing $\{\mathrm{PERM}_n\}$. Then, since the permanent of the adjacency matrix of a bipartite graph G is exactly $\mu(\Gamma)^2$, it follows from Corollary 6.5 that the counting width of the μ^2 is $o(n)$, giving a contradiction. ☐

It is worth noting why we consider the parameter μ^2 rather than μ itself in the proof above. The proof of Theorem 6.4, relying on [2, Theorem 6] relates the counting witdth of a class C to the size of supports in symmetric circuits deciding C. Specifically, this is proved for circuits whose input is the adjacency matrix of a graph Γ and which are symmetric with respect to permutations of the vertices of the graphs. This is why we need to take the permanent of the adjacency matrix, rather than the biadjacency matrix of the graph Γ. We consider this point in more detail in Section 7.3 when we consider lower bounds in fields of positive characteristic.

7.2 Construction

The construction to prove Theorem 7.2 is an adaptation of a standard construction by Cai, Fürer and Immerman [9] which gives non-isomorphic graphs X and Y with $X \equiv^k Y$ for arbitrary k (see also [12]). 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, $\Gamma = (V, E)$ is always a 3-regular 2-connected graph. From this, we first define a graph $X(\Gamma)$. The vertex set of $X(\Gamma)$ 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, $X(\Gamma)$ 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 $e \in \{f, g, h\}$, $X(\Gamma)$ 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 $X(\Gamma)$.

The construction of $X(\Gamma)$ from Γ 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 $\tilde{X}_x(\Gamma)$ is obtained by modifying the construction of $X(\Gamma)$ 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 $e \in \{f, g, h\}$, $X(\Gamma)$ contains the

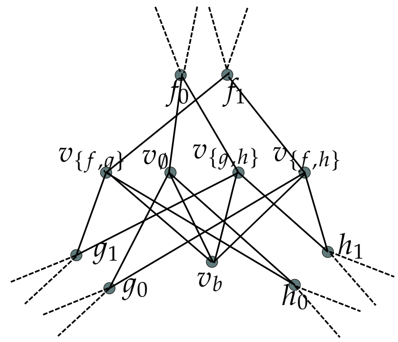

Figure 1: A gadget in $X(\Gamma)$ corresponding to vertex v with incident edges f, g, h

edge $\{e_1, v_S\}$ if $e \in S$ and the edge $\{e_0, v_S\}$ otherwise. Equivalently, we could describe this by saying that in this gadget, we interchange the roles of g_0 and g_1.

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

Lemma 7.3. *For any $x, y \in V$, $\tilde{X}_x(\Gamma)$ and $\tilde{X}_y(\Gamma)$ are isomorphic.*

Proof (sketch). Note that the gadget corresponding to a vertex v as in Figure 1 admits automorphisms that swap e_0 and e_1 for any two edges e incident on v. Now, let $x = v_0, v_1, \ldots, v_t = y$ be a simple path from x to y in Γ. We obtain an isomorphism from $\tilde{X}_x(\Gamma)$ to $\tilde{X}_y(\Gamma)$ by interchanging e_0 and e_1 for all edges on this path, and extending this to the induced automorphisms of the gadgets corresponding to $v_1, \ldots, v_{t-1}$. □

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

By known properties of the CFI construction, we also have the following (see [12, Theorem 3]).

Lemma 7.4. *If the treewidth of Γ is greater than k, then $X(\Gamma) \equiv^k \tilde{X}(\Gamma)$.*

The purpose of the balance vertices is to change the structure of the perfect matchings. Indeed, if we consider the subgraph of $X(\Gamma)$ 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, $X(\Gamma)$ 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 $X(\Gamma)$ and $\tilde{X}(\Gamma)$ contain different numbers of perfect matchings.

In the sequel, we write X to denote either one of the graphs $X(\Gamma)$ or $\tilde{X}(\Gamma)$, $V(X)$ to denote its vertices and $E(X)$ to denote its edges. We continue to use V and E for the vertices and edges of Γ. 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:

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

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

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 7.5. *The number of non-uniform matchings in $X(\Gamma)$ is the same as in $\tilde{X}(\Gamma)$.*

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 $X(\Gamma)$ and $\tilde{X}(\Gamma)$. 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 $X(\Gamma)^-$ and $\tilde{X}(\Gamma)^-$ be the subgraphs of $X(\Gamma)$ and $\tilde{X}(\Gamma)$, respectively, obtained by removing the edges between $\{e_0, e_1\}$ and I_v. It is clear that any matching M in $X(\Gamma)$ with $p^M = p$ is also a perfect matching in $X(G)^-$, and similarly for $\tilde{X}(\Gamma)$. However, $X(\Gamma)^-$ and $\tilde{X}(\Gamma)^-$ are isomorphic. This follows by an argument analogous to the proof of Lemma 7.3. Since Γ is 2-connected, there is a path $s = uv_1 \cdots v_{t-1}x$ from u to the special vertex x that does not involve the edge e. We can then define an isomorphism from $X(\Gamma)^-$ to $\tilde{X}(\Gamma)^-$ by mapping e_0 to e_1, for each edge f on the path s, 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 $\tilde{X}(\Gamma)$. For this, it is useful to first analyse the orientations of the underlying graph Γ.

Orientations. An *orientation* of Γ is a directed graph obtained from Γ by assigning to each edge $\{u, v\} \in E$ a direction, either (u, v) or (v, u). There are exactly $2^{|E|}$ distinct orientations of Γ. We say that a vertex $v \in V$ is *odd* with respect to an orientation $\overrightarrow{\Gamma}$ of Γ if it has an odd number of incoming directed edges and *even* otherwise. For an orientation $\overrightarrow{\Gamma}$ of Γ, we write $\mathrm{odd}(\overrightarrow{\Gamma})$ for the set of its odd vertices. We say that the orientiation $\overrightarrow{\Gamma}$ is *odd* if $|\mathrm{odd}(\overrightarrow{\Gamma})|$ is odd, and we say that it is *even* otherwise.

Lemma 7.6. *If $|V|/2$ is even, then all orientations of Γ are even. If $|V|/2$ is odd, then all orientations of Γ are odd.*

Proof. Note that since Γ 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 $\overrightarrow{\Gamma}$, let $\mathrm{in}(v)$ denote the number of edges incoming to the vertex v. Then, $|E| = \sum_v \mathrm{in}(v)$. But, $\sum_v \mathrm{in}(v) \equiv |\mathrm{odd}(\overrightarrow{\Gamma})| \pmod 2$. □

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

Lemma 7.7. *If $\Gamma = (V, E)$ is even then for every set $S \subseteq V$ with $|S|$ even, there is an orientation $\overrightarrow{\Gamma}$ of Γ with $\mathrm{odd}(\overrightarrow{\Gamma}) = S$. Similarly if $\Gamma = (V, E)$ is odd, then for every set $S \subseteq V$ with $|S|$ odd, there is an orientation $\overrightarrow{\Gamma}$ of Γ with $\mathrm{odd}(\overrightarrow{\Gamma}) = S$.*

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 $\overrightarrow{\Gamma}$ of Γ with $\mathrm{odd}(\overrightarrow{\Gamma}) = S$, then there is also an orientation $\overrightarrow{\Gamma}'$ with $\mathrm{odd}(\overrightarrow{\Gamma}') = S \triangle \{u, v\}$. Now, consider any simple path from u to v in Γ and let $\overrightarrow{\Gamma}'$ be the orientation obtained from $\overrightarrow{\Gamma}$ by reversing the direction of every edge on this path. □

Indeed, we can say more.

Lemma 7.8. *For every set $S \subseteq V$ with $|S| = |E| \pmod 2$, there are exactly $2^{|V|/2+1}$ distinct orientations $\overrightarrow{\Gamma}$ with $\mathrm{odd}(\overrightarrow{\Gamma}) = S$.*

Proof. Let A be the $V \times E$ incidence matrix of the graph Γ. 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 Γ: for a vector $\pi \in \mathbb{F}_2^E$, and an orientation $\overrightarrow{\Gamma}$, define $\pi\overrightarrow{\Gamma}$ to be the orientation obtained from $\overrightarrow{\Gamma}$ by changing the orientation of each edge e with $\pi(e) = 1$. Indeed, fixing one particular orientation $\overrightarrow{\Gamma}$, the action generates all orientations and gives a bijective correspondence between the vectors in $\mathbb{F}_2^E$ and the orientations of Γ. 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 subsets of V and gives a bijection between $\mathbb{F}_2^V$ and the powerset of V.

Then, it can be seen that $\mathrm{odd}(\pi\overrightarrow{\Gamma}) = (A\pi)\mathrm{odd}(\overrightarrow{\Gamma})$. 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., $\{v \mid (A\pi)(v) = 1\} = \mathrm{odd}(\overrightarrow{\Gamma}) \triangle \mathrm{odd}(\pi\overrightarrow{\Gamma})$ for any orientation $\overrightarrow{\Gamma}$.

Fixing a particular orientation $\overrightarrow{\Gamma}$, the action of $\mathbb{F}_2^E$ generates all orientation $\pi\overrightarrow{\Gamma}$, and A maps this to the collection of all sets $\mathrm{odd}(\overrightarrow{\Gamma})\triangle\mathrm{odd}(\pi\overrightarrow{\Gamma})$. Then, by Lemmas 7.6 and 7.7 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 $\mathrm{odd}(\pi\overrightarrow{\Gamma}) = T\triangle\mathrm{odd}(\overrightarrow{\Gamma})$. ◻

Matchings in gadgets. Any uniform perfect matching M of X induces an orientation of Γ, which we denote $\overrightarrow{\Gamma}^M$ as follows: any edge $e = \{u, v\} \in E$ is oriented from u to v in $\overrightarrow{\Gamma}^M$ 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.

S has exactly four perfect matchings:

$$\begin{array}{cccc}
f_0 - v_\emptyset & g_0 - v_{\{f,h\}} & h_0 - v_{\{f,g\}} & v_b - v_{\{g,h\}} \\
f_0 - v_{\{g,h\}} & g_0 - v_\emptyset & h_0 - v_{\{f,g\}} & v_b - v_{\{f,h\}} \\
f_0 - v_{\{g,h\}} & g_0 - v_{\{f,h\}} & h_0 - v_{\{f,g\}} & v_b - v_\emptyset \\
f_0 - v_{\{g,h\}} & g_0 - v_{\{f,h\}} & h_0 - v_\emptyset & v_b - v_{\{f,g\}}.
\end{array}$$

T has exactly two perfect matchings:

$$\begin{array}{cccc}
f_0 - v_\emptyset & g_0 - v_{\{f,h\}} & h_1 - v_{\{g,h\}} & v_b - v_{\{f,g\}} \\
f_0 - v_{\{g,h\}} & g_0 - v_\emptyset & h_1 - v_{\{f,h\}} & v_b - v_{\{f,g\}}.
\end{array}$$

Hence, for any orientation $\overrightarrow{\Gamma}$, we get a matching $M \subseteq X$ with $\overrightarrow{\Gamma}^M = \overrightarrow{\Gamma}$ 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 $\overrightarrow{\Gamma}$. In $X(\Gamma)$, the relevant subgraph at v is isomorphic to S if v is even in $\overrightarrow{\Gamma}$ and it is isomorphic to T if v is odd in $\overrightarrow{\Gamma}$. The same is true for all vertices in $\tilde{X}(\Gamma)$, apart from the special vertex x. For this one, the relevant subgraph is isomorphic to the induced subgraph on S if x is odd and to T if x is even. In either case, we get a perfect matching M with $\overrightarrow{\Gamma}^M = \overrightarrow{\Gamma}$ 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 $\vec{\Gamma}$ of Γ, the number of uniform perfect matchings M of $X(\Gamma)$ with $\vec{\Gamma}^M = \vec{\Gamma}$ is $2^{|\mathrm{odd}(\vec{\Gamma})|}4^{|V|-|\mathrm{odd}(\vec{\Gamma})|}$. The number of uniform perfect matchings in $\tilde{X}(\Gamma)$ depends on whether the special vertex x is odd in $\vec{\Gamma}$ or not. If it is, the number is $2^{|\mathrm{odd}(\vec{\Gamma})|-1}4^{|V|-|\mathrm{odd}(\vec{\Gamma})|+1}$ otherwise it is $2^{|\mathrm{odd}(\vec{\Gamma})|+1}4^{|V|-|\mathrm{odd}(\vec{\Gamma})|-1}$. Thus, if we denote the number of uniform perfect matchings in X by $\#MX$, then we have

$$\#MX(\Gamma) = \sum_{\vec{\Gamma}} 2^{|\mathrm{odd}(\vec{\Gamma})|}4^{|V|-|\mathrm{odd}(\vec{\Gamma})|}$$

where the sum is over all orientations of Γ. Then, by Lemma 7.8,

$$\#MX(\Gamma) = 2^{|V|/2+1} \sum_{S\subseteq V\,:\,|S|\equiv|E| \pmod 2} 2^{|S|}4^{|V|-|S|}.$$

By the same token,

$$\#M\tilde{X}(\Gamma) = 2^{|V|/2+1} \sum_{S\subseteq V\,:\,|S|\not\equiv|E| \pmod 2} 2^{|S|}4^{|V|-|S|}.$$

We aim to show that $\#MX(\Gamma)$ and $\#M\tilde{X}(\Gamma)$ are different. Let P_m denote the number $\sum_{S\subseteq[2m]:|S|\mathrm{even}} 2^{|S|}4^{2m-|S|}$ and Q_m denote the number $\sum_{S\subseteq[2m]:|S|\mathrm{odd}} 2^{|S|}4^{2m-|S|}$.

Lemma 7.9. *For all $m \geq 1$, $P_m - Q_m = 4^m$.*

Proof. We have

$$\begin{aligned}
P_m - Q_m &= \sum_{S\subseteq[2m]:|S|\mathrm{even}} 2^{|S|}4^{2m-|S|} - \sum_{S\subseteq[2m]:|S|\mathrm{odd}} 2^{|S|}4^{2m-|S|}\\
&= \sum_{S\subseteq[2m]} (-1)^{|S|}2^{|S|}4^{2m-|S|}\\
&= 4^{2m}\sum_{S\subseteq[2m]} (\frac{-2}{4})^{|S|}\\
&= 4^{2m}(1-\frac{1}{2})^{2m} = 4^m.
\end{aligned}$$

□

Proof of Theorem 7.2 By a standard expander graph construction (e. g., [1]), for any k, we can find a 3-regular graph Γ with treewidth at least k and $2n = O(k)$ vertices. Then $X(\Gamma)$ and $\tilde{X}(\Gamma)$ both have $O(k)$ vertices and by Lemma 7.4 we have $X(\Gamma) \equiv^k \tilde{X}(\Gamma)$. Moreover, $X(\Gamma)$ and $\tilde{X}(\Gamma)$ have the same number of non-uniform perfect matchings by Lemma 7.5. 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, $|\mu(X(\Gamma)) - \mu(\tilde{X}(\Gamma))| = 2^{3n+1}$, which is a power of 2 as required.

7.3 Positive characteristics

Theorem 7.1 gives a lower bound for square-symmetric circuits computing the permanent in characteristic zero, which contrasts neatly with the upper bound for the determinant established in Theorem 4.1. We now briefly sketch the lower bounds that our method yields for computing the permanent in positive characteristic. The short statement is that we can get exponential lower bounds for all odd characteristics, but only with respect to a more stringent symmetry requirement—namely *matrix symmetry*.

Theorem 7.2 establishes a lower bound on the counting width of μ—the number of perfect matchings in a graph. The theorem also establishes a lower bound on (μ mod p), the number of perfect matchings modulo p for any odd value of p. This is because for the graphs X and Y obtained from the theorem, we have $\mu(X) - \mu(Y) = 2^l$ and so $\mu(X) \not\equiv \mu(Y) \pmod{p}$ for any odd p.

Corollary 7.10. *For any odd p, the counting width of the number of perfect matchings modulo p of a bipartitioned graph with n vertices is $\Omega(n)$.*

However, we do not have a lower bound on the counting width of (μ^2 mod p).It is quite possible that, for the graphs X and Y of Theorem 7.2 we have $\mu(X)^2 \equiv \mu(Y)^2 \pmod{p}$. This is the reason why Theorem 7.1 is only formulated for characteristic zero.

The reason that we have to use μ^2 in the proof of Theorem 7.1 has to do with our use of the connection between the counting width of a class C of relational structures and the orbit size of a circuit family deciding membership in C as established in [2], which we use as a black box to get Theorem 6.4. This connection between counting width and orbit size of circuits is established in [2] specifically for circuits taking relational structures as input and which are symmetric under the action of permutations of the elements. In the context of graphs, this means it applies to circuits taking the adjacency matrix of a graph $\Gamma = (V, E)$ as input and symmetric under all permutations of V. For such circuits, it establishes that if Γ and Δ are two graphs on vertex set $[n]$ with $\Gamma \equiv^k \Delta$, then their adjacency matrices cannot be distinguished by a $\mathbf{Sym}_n$ circuit of small, i. e., $2^{o(k)}$ orbit size. From this, we cannot directly obtain lower bounds for circuits that take the *biadjacency* matrix of a graph as input. To do this, we have to look inside the black box of Theorem 6.4, relating counting width to circuits.

Consider a bipartite graph $\Gamma = (V, E)$ with bipartition $V = A \cup B$, where A and B both have n elements. If we identify both sets A and B with the set $[n]$ (equivalently, if we fix a bijection between A and B), then the biadjacency matrix $\mathcal{B}_\Gamma$ of Γ can be seen as the adjacency matrix of a directed graph $\hat{\Gamma}$ on vertex set $[n]$ with an arc (i, j) whenever there is an edge between $i \in A$ and $j \in B$. It then follows directly from the results of [2] that if we have a pair of bipartite graphs Γ and Δ with $\hat{\Gamma} \equiv^k \hat{\Delta}$, then the biadjacency matrices of Γ and Δ cannot be distinguished by small symmetric circuits. Unfortunately, for the graphs X and Y of Theorem 7.2, we are not able to prove that $\hat{X} \equiv^k \hat{Y}$.

The proof that a pair of structures are equivalent with respect to $\equiv^k$ is often given by as a Duplicator winning strategy in the k-pebble *bijection game* (see [12]). The relation between such winning strategies and lower bounds for symmetric circuits is made explicit in [11]. This has been greatly expanded to a method for proving lower bounds for Γ-symmetric circuits for arbitrary groups Γ in [15]. What this means in our context is that to prove that the biadjacency matrices of the bipartite graphs X and Y are not distinguished by small *square-symmetric* circuits, we need to show a Duplicator winning strategy that respects a fixed bijection between the two parts of the bipartition in X and Y. We are not able to do this. What we do know is that the Duplicator winning strategy that shows $X \equiv^k Y$ does respect the bipartition itself. In other words, we can expand the graphs X and Y with colours for the sets A and B (the two parts of the bipartition) and these coloured graphs are still equivalent with respect to $\equiv^k$. This is sufficient to establish that their biadjacency matrices are not distnguished by *matrix-symmetric* circuits of size $2^{o(k)}$. Since for a bipartite graph G, the permanent of its biadjacency matrix $\mathrm{perm}_{\mathbb{F}}(\mathcal{B}_\Gamma)$, over a field $\mathbb{F}$ of characteristic p is exactly $(\mu(\Gamma) \bmod p)$, this allows us to establish our lower bound.

Theorem 7.11. *There is no family of matrix-symmetric arithmetic circuits over any field of odd characteristic of orbit size* $2^{o(n)}$ *computing* $\{\mathrm{PERM}_n\}$.

7.4 Equivariant determinantal representations

Lower bounds for computing the permanent in symmetric models of computation have previously been established, notably in the work of Landsberg and Ressayre [24]. They establish an exponential lower bound on the *equivariant determinantal complexity* of the permanent, specifically over the complex field $\mathbb{C}$. In this section we make a brief comparison of our results with theirs.

The determinantal complexity (DC) of a polynomial $p \in \mathbb{F}[X]$ is defined to be the least m such that there is an $m \times m$ matrix M with entries that are affine linear forms in X such that $\det(M) = p$. Such a matrix is called a *determinantal representation* of p. It is known [29] that every polynomial in VP has DC that is at most quasi-polynomial. It follows that an exponential lower bound on the DC of the permanent would show that it is not in VP, separating VP from VNP. Indeed, such a bound would show that circuits computing PERM_n must have size at least 2^{n^δ} for some positive δ. On the other hand, an exponential lower bound on the circuit complexity of the permanent would also yield a similar lower bound for its determinental complexity. To see this note that using an $O(n^3)$ family of circuits for computing $\{\mathrm{DET}_n\}$ and an $m \times m$ determinantal representation M of the permanent, we get an $O(m^3)$ family of circuits computing $\{\mathrm{PERM}_n\}$. This is obtained by taking the circuit computing the determinant and attaching to its m^2 inputs the circuits (of at most $O(n)$ size) computing the affine linear forms that form the entries of M. Hence a $2^{f(n)}$ lower bound on the circuit complexity fo the permanent gives us a $2^{f(n)/3}$ lower bound on its determinantal complexity.

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. Our aim is to see how this relates to our lower bounds on symmetric circuit complexity. Unfortunately, the relationship is not straightforward in either direction because of the different notions of symmetry used and the symmetry-breaking nature of the translation from circuits to determinantal representations. To make this explicit, we first introduce some definitions. These are simplified from (and so less general than) those given by Landsberg and Ressayre but suffice to show that our results are incomparable with theirs.

Formally, consider a homogeneous polynomial $p \in \mathbb{C}[X]$. Let $\mathbf{GL}_X$ denote the group of invertible linear maps on the vector space $\mathbb{C}^X$. In what follows, we identify $\mathbb{C}^X$ with the set of linear forms in the variables X, so we can write Al for $A \in \mathbf{GL}_X$ and l a linear form in X. We extend the notation to affine linear forms by the convention that $Am = c + Al$ when $m = c + l$ for $c \in \mathbb{C}$ and l a linear form.

For a map $A \in \mathbf{GL}_X$, we write $p(AX)$ to mean the polynomial obtained from p by replacing each variable $x \in X$ by the linear form Ax. We now define the *symmetry group* of p to be the group $\mathcal{S}_p$ of linear maps $A \in \mathbf{GL}_X$ such that $p(AX) = p$. In particular, when $X = \{x_{ij} \mid i, j \in [n]\}$ we can think of the elements of $\mathbb{C}^X$ as $n \times n$ matrices and the symmetry group of DET_n can be identified with the group $\mathcal{S}_{\mathrm{DET}_n} = (\mathbf{GL}_n \times \mathbf{GL}_n)/\mathbb{C} \rtimes \mathbb{Z}_2$. Here the action of $(A, B) \in \mathbf{GL}_n \times \mathbf{GL}_n$ takes $V \in \mathbb{C}^X$ to $AVB^{-1}/\det(AB^{-1})$ and the action of the non-trivial element in $\mathbb{Z}_2$ takes V to V^T.

Let M be an $m \times m$ determinantal representation of a polynomial p. For $A \in \mathbf{GL}_X$, write M^A to be the matrix obtained from M by replacing each entry m by Am (where we see each affine linear form m as a polynomial in $\mathbb{C}[X]$). We say that M is an *equivariant* determinantal representation of p if for each $A \in \mathcal{S}_p$ there is a $B \in \mathcal{S}_{\mathrm{DET}_n}$ such that $M^A = BM$. In other words, all symmetries of p extend to symmetries of M. Landsberg and Ressayre prove that any equivariant determinantal representation of PERM_n must have size $\Omega(4^m)$.

We could ask if this lower bound yields a lower bound for symmetric circuits just as an exponential lower bound on the *determinantal complexity* of the permanent yields a lower bound for its unrestricted circuit complexity. This would require a translation, along the lines of Valiant [29] from symmetric circuits to equivariant determinantal representations. There is little reason to believe that we could have such a translation. For one thing, the symmetry requirement for *square-symmetric* circuits is only that they are invariant under the natural action of $\mathbf{Sym}_n$ on PERM_n, and this is a rather small subgroup of $\mathcal{S}_{\mathrm{PERM}_n}$. Secondly, Valiant's translation of circuits to determinantal representations is not symmetry preserving. Thus, the representations obtained from this translation applied to square-symmetric circuits are not even guaranteed to be equivariant with respect to the action of $\mathbf{Sym}_n$ on PERM_n, let alone that of $\mathcal{S}_{\mathrm{PERM}_n}$.

In the other direction, we could ask if our lower bounds for symmetric circuits for the permanent yield any lower bounds for equivariant determinantal complexity, especially in combination with the polynomial upper bound for transpose-symmetric circuits for the determinant proved in Theorem 4.1. Indeed, given a circuit C of size s computing DET_m and an $m \times m$ determinantal representation M of p, a polynomial on n variables, we obtain a circuit C' computing p of size $s(m) + O(nm)$, where the second term represents the size of the subcircuits required to compute the affine expressions making up the entries of M. If an equivariant determinantal expression translates to a symmetric circuit, then a symmetric circuit lower bound can be translated to a lower bound on equivariant determinantal complexity. Since the symmetry conditions for circuits are less restrictive, this seems plausible, but there is a mismatch.

Consider the case when p is PERM_n, and C is the square-symmetric circuit for DET_m obtained from Theorem 4.1. For the circuit C' to be square-symmetric, we require that the action of $\mathbf{Sym}_n$ on the variables $X = \{x_{ij} \mid i, j \in [n]\}$ extends to automorphisms of the circuit. Since this action gives a subgroup of $\mathcal{S}_{\mathrm{PERM}_n}$ acting on PERM_n, we know that for each $\pi \in \mathbf{Sym}_n$ there is a $B \in \mathcal{S}_{\mathrm{DET}_m}$ such that $M^\pi = BM$. If this map B was itself the action of a permutation in $\mathbf{Sym}_m$ on the rows and columns of M, the square-symmetry of C would guarantee that C' was also square-symmetric. However, the equivariance of M does not enforce this. So, to state the lower bound on determinantal complexity that we can get from our results, we define an alternative notion of equivariance.

Say that M is *permutation equivariant* if for each $\pi \in \mathbf{Sym}_n$, there is a permutation matrix $B \in \mathbf{GL}_m$ such that $M^\pi = BMB^{-1}$. Note that this notion is incomparable with equivariance of M. We have relaxed the requirement by only asking that permutations π in $\mathcal{S}_{\mathrm{PERM}_n}$ extend to symmetries of M, but we have made it more stringent by asking that the symmetry they extend to is itself a permutation matrix in $\mathcal{S}_{\mathrm{DET}_m}$. Here, we identify the permutation matrix B with the element $((B, B)/1, 1)$ in $\mathcal{S}_{\mathrm{DET}_m}$ as this yields the desired permutation action.

We can now state the following corollary of our results.

Corollary 7.12. *Any permutation equivariant determinantal representation of* PERM_n *has size* $2^{\Omega(n)}$.

8 Concluding discussion

We have introduced a novel restriction of arithmetic circuits 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 is computable using a polynomial-size family of square-symmetric circuits, while the latter requires at least exponential-size families of square-symmetric circuits for fields of characteristic 0. The lower bound for the permanent can be extended to fields of odd characteristic for matrix-symmetric circuits.

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 clear if any of them can be carried out symmetrically. Indeed, there are many other efficient ways of computing a determinant and it seems quite plausible 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., [10]) 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 [21]. It is when we restrict to *arithmetic* circuits, and also require symmetry, that the question is open.

There is a corresponding question for the permanent lower bound. That is, can the lower bound on square-symmetric circuits for the permanent be extended to all fields of odd positive characteristic. This might be done by adapting our construction to analyse the counting width of the number of cycle covers of general graphs. Another approach would be to adapt our construction and choose Γ so that the sum of the numbers of perfect matchings in $X(\Gamma)$ and $\tilde{X}(\Gamma)$ is a power of two. This would suffice to establish that $\mu(X(\Gamma))^2$ and $\mu(\tilde{X}(\Gamma))^2$ also differ by a power of two.

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. However, we could consider a richer group that allowed for arbitrary even permutations of the rows and columns. In a recent paper [15] we were able to show, with this rich group of symmetries, an exponential lower bound for the determinant. It would be interesting to identify the exact boundary on the spectrum of symmetries between the tractability and the intractability of the determinant.

Finally, it is reasonable to think that even just considering square-symmetric circuits, 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] Miklós Ajtai: Recursive construction for 3-regular expanders. *Combinatorica*, 14(4):379–416, 1994. [doi:10.1007/BF01302963] 24

[2] Matthew Anderson and Anuj Dawar: On symmetric circuits and fixed-point logics. *Theory Computing Sys.*, 60(3):521–551, 2017. Preliminary version in STACS'14. [doi:10.1007/s00224-016-9692-2] 2, 3, 10, 16, 17, 18, 19, 25

[3] Matthew Anderson, Anuj Dawar, and Bjarki Holm: Solving linear programs without breaking abstractions. *J. ACM*, 62(6):48:1–26, 2015. [doi:10.1145/2822890] 3, 6

[4] Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky: On Weisfeiler–Leman invariance: Subgraph counts and related graph properties. *J. Comput. System Sci.*, 113:42–59, 2020. Preliminary version in FCT'19. [doi:10.1016/j.jcss.2020.04.003] 3

[5] Albert Atserias, Anuj Dawar, and Joanna Ochremiak: On the power of symmetric linear programs. *J. ACM*, 68(4), 2021. Preliminary version in LICS'19. [doi:10.1145/3456297] 3, 6, 17

[6] Walter Baur and Volker Strassen: The complexity of partial derivatives. *Theoret. Comput. Sci.*, 22(3):317–330, 1983. [doi:10.1016/0304-3975(83)90110-X] 10

[7] Markus Bläser and Gorav Jindal: On the complexity of symmetric polynomials. In *Proc. 10th Innovations in Theoret. Comp. Sci. Conf. (ITCS'19)*, pp. 47:1–14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.ITCS.2019.47] 11

[8] Andreas Blass, Yuri Gurevich, and Saharon Shelah: On polynomial time computation over unordered structures. *J. Symbolic Logic*, 67(3):1093–1125, 2002. [doi:10.2178/jsl/1190150152] 3

[9] JIN-YI. CAI, MARTIN FÜRER, AND NEIL IMMERMAN: An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992. [doi:10.1007/BF01305232] 5, 6, 19

[10] ANUJ DAWAR: The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2(1):8–21, 2015. [doi:10.1145/2728816.2728820] 28

[11] ANUJ DAWAR: On symmetric and choiceless computation. In MOHAMMAD TAGHI HAJIAGHAYI AND MOHAMMAD REZA MOUSAVI, editors, *Topics in Theoretical Computer Science (TTCS'15)*, pp. 23–29. Springer, 2016. [doi:10.1007/978-3-319-28678-5_2] 25

[12] ANUJ DAWAR AND DAVID RICHERBY: The power of counting logics on restricted classes of finite structures. In *Proc. 21st Internat. Workshop Computer Science Logic (CSL'07)*, pp. 84–98. Springer, 2007. [doi:10.1007/978-3-540-74915-8_10] 6, 19, 20, 25

[13] ANUJ DAWAR AND PENGMING WANG: Definability of semidefinite programming and Lasserre lower bounds for CSPs. In *Proc. 32nd Ann. ACM/IEEE Symp. on Logic in Computer Science (LICS'17)*, pp. 1–12. ACM Press, 2017. [doi:10.1109/LICS.2017.8005108] 3, 6

[14] ANUJ DAWAR AND GREGORY WILSENACH: Symmetric arithmetic circuits. In *Proc. 47th Internat. Colloq. on Automata, Languages, and Programming (ICALP'20)*, pp. 36:1–18. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. [doi:10.4230/LIPIcs.ICALP.2020.36] 1, 4, 17

[15] ANUJ DAWAR AND GREGORY WILSENACH: Lower bounds for symmetric circuits for the determinant. In *Proc. 13th Innovations in Theoret. Comp. Sci. Conf. (ITCS'22)*, pp. 52:1–22. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2022. [doi:10.4230/LIPIcs.ITCS.2022.52] 26, 29

[16] ANUJ DAWAR AND GREGORY WILSENACH: Symmetric circuits for rank logic. *ACM Trans. Comput. Logic*, 23(1):6:1–35, 2022. [doi:10.1145/3476227] 10, 17

[17] LARRY DENENBERG, YURI GUREVICH, AND SAHARON SHELAH: Definability by constant-depth polynomial-size circuits. *Inform. Control*, 70(2–3):216–240, 1986. [doi:10.1016/S0019-9958(86)80006-7] 2, 3, 10

[18] MICHAEL A. FORBES, MRINAL KUMAR, AND RAMPRASAD SAPTHARISHI: Functional lower bounds for arithmetic circuits and connections to Boolean circuit complexity. In *Proc. 31st Comput. Complexity Conf. (CCC'16)*, pp. 33:1–19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.CCC.2016.33] 11

[19] DIMA GRIGORIEV AND MAREK KARPINSKI: An exponential lower bound for depth 3 arithmetic circuits. In *Proc. 30th STOC*, pp. 577–582. ACM Press, 1998. [doi:10.1145/276698.276872] 2

[20] FRANK HARARY: Determinants, permanents and bipartite graphs. *Math. Magazine*, 42(3):146–148, 1969. [doi:10.1080/0025570X.1969.11975950] 5

[21] BJARKI HOLM: *Descriptive Complexity of Linear Algebra*. Ph. D. thesis, U. Cambridge, 2010. Available at U. Cambridge. 11, 28

[22] MARK JERRUM AND MARC SNIR: Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982. [doi:10.1145/322326.322341] 2

[23] NEERAJ KAYAL AND RAMPRASAD SAPTHARISHI: A selection of lower bounds for arithmetic circuits. In MANINDRA AGRAWAL AND VIKRAM ARVIND, editors, *Perspectives in Computational Complexity*, pp. 77–115. Birkhäuser, 2014. Available at Microsoft Research. [doi:10.1007/978-3-319-05446-9_5] 2

[24] JOSEPH M. LANDSBERG AND NICOLAS RESSAYRE: Permanent v. determinant: An exponential lower bound assuming symmetry and a potential path towards Valiant's conjecture. *Diff. Geom. Appl*, 55:146–166, 2017. Preliminary version in ITCS'16. [doi:10.1016/j.difgeo.2017.03.017] 3, 26

[25] MARTIN OTTO: The logic of explicitly presentation-invariant circuits. In *Proc. 10th Internat. Workshop Computer Science Logic (CSL'96)*, pp. 369–384. Springer, 1996. [doi:10.1007/3-540-63172-0_50] 2, 3, 10

[26] BENJAMIN ROSSMAN: Subspace-invariant AC^0 formulas. *Logical Methods Comp. Sci.*, 15:3:1–12, 2019. [doi:10.23638/LMCS-15(3:3)2019] 3, 9, 10

[27] AMIR SHPILKA AND AVI WIGDERSON: Depth-3 arithmetic circuits over fields of characteristic zero. *Comput. Complexity*, 10:1–27, 2001. [doi:10.1007/PL00001609] 10

[28] AMIR SHPILKA AND AMIR YEHUDAYOFF: Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comp. Sci.*, 5(3–4):207–388, 2010. [doi:10.1561/0400000039] 7, 8

[29] LESLIE G. VALIANT: Completeness classes in algebra. In *Proc. 11th STOC*, pp. 249–261. ACM Press, 1979. [doi:10.1145/800135.804419] 1, 26, 27

[30] LESLIE G. VALIANT AND SVEN SKYUM: Fast parallel computation of polynomials using few processors. In *Proc. Internat. Symp. Math. Foundations of Comp. Sci. (MFCS'81)*, pp. 132–139. Springer, 1981. [doi:10.1007/3-540-10856-4_79] 7

[31] HERIBERT VOLLMER: *Introduction to Circuit Complexity – A Uniform Approach*. Springer, 1999. [doi:10.1007/978-3-662-03927-4] 7

AUTHORS

Anuj Dawar
Department of Computer Science and Technology
University of Cambridge
Cambridge, U.K.
anuj.dawar@cl.cam.ac.uk
`http://www.cst.cam.ac.uk/people/ad260`

Gregory Wilsenach
Department of Computer Science and Technology
University of Cambridge
Cambridge, U.K.
greg.b.wils@gmail.com
https://www.gregorywilsenach.com

ABOUT THE AUTHORS

ANUJ DAWAR is Professor of Logic and Algorithms at Cambridge, where he has been on the faculty since 1999. He obtained degrees from the Indian Institute of Technology, Delhi and the University of Delaware before studying for his Ph. D. at the University of Pennsylvania in 1993. He worked as a postdoctoral researcher and a lecturer at Swansea University before moving to Cambridge. Much of his research has revolved around descriptive complexity and its connections to broader questions in logic and theoretical computer science.

GREGORY WILSENACH, known to his friends as Greg, studied mathematics and engineering at the University of Cape Town in South Africa before moving to Cambridge in England to undertake a Master's degree in mathematics. He stayed on in Cambridge to pursue a Ph. D. in computer science under the supervision of Anuj Dawar. After completing his Ph. D. in 2019, he was employed as a postdoctoral researcher in the same department until the end of 2021. He now works at Catapult.