Optimal Lower Bounds for Symmetric Modular Circuits

Benedikt Pago
University of Cambridge, UK

Abstract

A notorious open question in circuit complexity is whether Boolean operations of arbitrary arity can efficiently be expressed using modular counting gates only. Håstad's celebrated switching lemma yields exponential lower bounds for the dual problem – realising modular arithmetic with Boolean gates – but, a similar lower bound for modular circuits computing the Boolean AND function has remained elusive for almost 30 years.

We solve this problem for the restricted model of symmetric circuits: We consider MOD_m-circuits of arbitrary depth, and for an arbitrary modulus $m \in \mathbb{N}$, and obtain subexponential lower bounds for computing the n-ary Boolean AND function, under the assumption that the circuits are syntactically symmetric under all permutations of their n input gates. This lower bound is matched precisely by a construction due to (Idziak, Kawałek, Krzaczkowski, LICS'22), leading to the surprising conclusion that the optimal symmetric circuit size is already achieved with depth 2.

Motivated by another construction from (LICS'22), which achieves smaller size at the cost of greater depth, we also prove tight size lower bounds for circuits with a more liberal notion of symmetry characterised by a nested block structure on the input variables.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases symmetric circuits, modular counting, lower bounds, CC^0

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.187

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2604.04760` [29]

Funding The author was funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee: grant number EP/X028259/1.

Acknowledgements I am grateful to Piotr Kawałek for posing this question to me, and for his invaluable help in learning and presenting the research context.

1 Introduction

There are many long-standing open questions in circuit complexity that are surprisingly simple in their formulation, yet no solution to them has been found in decades. Among them there is the following: Can the n-ary Boolean AND_n-function be represented with a polynomial-size circuit of depth two using only MOD_6-gates? A MOD_6-gate is a Boolean gate which takes an arbitrary number of inputs and returns 1 if and only if their sum modulo 6 belongs to an accepting set $S \subseteq \mathbb{Z}_6$ (where S can vary for different gates).

More generally, a $\mathrm{CC}_h[m]$-circuit is a depth-h Boolean circuit consisting only of MOD_m-gates. The question is: For fixed positive integers h and m, what is the asymptotic size of the smallest possible $\mathrm{CC}_h[m]$-circuit that computes AND_n? Is it polynomial in n? In common complexity-theoretic terms, this question is phrased as "$\mathrm{CC}^0 = \mathrm{ACC}^0$?". The class CC^0 consists of all constant depth circuits that only use modular counting gates, while in ACC^0, the circuits may additionally contain Boolean disjunction, conjunction and negation gates. It is stunning that very little progress has been made despite the fact that this problem was first raised almost 30 years ago [3].

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 187; pp. 187:1–187:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As of today, only slightly superlinear lower bounds are known, and only for the number of wires, not gates [7], both for general $\mathrm{CC}_h[m]$-circuits as well as for the most restricted open case $\mathrm{CC}_2[6]$. The lack of strong lower bounds is even more surprising when one compares it with the dual question: Can modulo counting be performed efficiently by constant depth circuits using only the Boolean operations $\wedge, \vee, \neg$? This question was famously answered in the negative by Håstad [21] already in the 1980s. Indeed, his switching lemma yields an asymptotically optimal exponential lower bound against constant depth Boolean circuits computing the parity function. More generally, it has been shown that MOD_q is not in $\mathrm{AC}^0[p]$ (the extension of AC^0 with MOD_p gates), whenever $p \neq q$ are distinct primes [30]. To sum up: We have known for a long time that Boolean operations cannot simulate modulo counting, but it is notoriously hard to settle whether modulo counting can simulate Boolean operations.

Common belief is in favour of a negative answer: Barrington, Straubing and Thérien first conjectured an exponential lower bound in [3], and since the work by Barrington, Beigel and Rudich in 1994 [2], a $2^{\Omega(n^\varepsilon)}$ size lower bound for $\mathrm{CC}_h[m]$-circuits computing AND_n is considered likely (where $0 < \varepsilon < 1$). Some refer to this conjecture as the *Exponential Size Hypothesis* (ESH). For the very restricted setting of two layers with two different prime moduli, that is, $\mathrm{MOD}_q \circ \mathrm{MOD}_p$-circuits where $p \neq q$, an even stronger lower bound has been established unconditionally: Such circuits require size $2^{\Omega(n)}$ to compute AND_n [20, 19, 31].

In this paper we study a circuit restriction of a different nature: Since the function AND_n is symmetric under all permutations of its inputs, it admits a *symmetric circuit* representation. Such symmetric constructions are typically natural and intuitive, and it is also reasonable to assume that they are not too far from optimal. Formally, we say that a circuit C is *fully symmetric* (or $\mathbf{Sym}_n$-symmetric) if for every $\pi \in \mathbf{Sym}_n$, there exists an automorphism of C that permutes its input gates $x_1, \ldots, x_n$ according to π. We completely determine the $\mathrm{CC}_h[m]$-circuit complexity of AND_n, for any depth $h \geq 2$, and any modulus m with at least two prime divisors [1] , as far as fully symmetric circuits are concerned.

▶ **Theorem 1.** *Fix an integer $m \geq 6$ with at least $r \geq 2$ distinct prime divisors. For every family of $\mathbf{Sym}_n$-symmetric MOD_m-circuits $(C_n)_{n\in\mathbb{N}}$ computing the Boolean function AND_n, the circuit size is at least $|C_n| \geq 2^{\Omega(n^{1/r}\cdot \log n)}$. There exists a family of $\mathrm{CC}_2[m]$-circuits that achieves this bound.*

The new contribution is the lower bound; the upper bound was presented by Idziak, Kawałek, Krzaczkowski in 2022 [24, Proposition 3.1], and independently by Chapman and Williams [6], based on an idea from [2]. For completeness, we review the upper bound in Section 5. The most surprising insight from Theorem 1 is the discovery that the natural and elegant depth-2 construction from [24, Proposition 3.1] is in fact optimal under the assumption of $\mathbf{Sym}_n$-symmetry. Thus, one can never gain savings in the asymptotic size by increasing circuit depth beyond 2, except by breaking symmetries. This answers a question implicitly posed by Kawałek and Weiß in [27], which is detailed below.

Interestingly, a more involved construction from [24] demonstrates that (partially) breaking symmetries does indeed allow to build smaller circuits, at the expense of greater depth: For every constant $h > 2$, there exist $\mathrm{CC}_h[m]$-circuits for AND_n whose size is strictly smaller than $2^{\Theta(n^{1/r}\cdot \log n)}$, and the savings in size increase with h.

[1] The case where m is a prime power can be ignored. It is known that then, $\mathrm{CC}_h[m]$-circuits cannot compute AND_n for arbitrarily large n, see [24, Proposition 2.1]

Examining this depth-h construction [24, Proposition 4.3], one finds that it is not $\mathbf{Sym}_n$-symmetric (which it cannot be by Theorem 1), but respects a smaller group of symmetries, that we call *nested block symmetry*. This notion of symmetry is naturally exhibited by recursive circuit constructions that follow a divide-and-conquer approach; another such example can be found in [6].

The symmetry group is best described as the automorphism group of a tree whose leaves correspond to the input variables $x_1, \dots, x_n$ of the circuit. We formalise this idea by fixing an h-tuple $\boldsymbol{k} = (k_1(n), \dots, k_h(n))$ of functions in n such that $\prod_{i\in[h]} k_i(n) = n$, which defines for each $n \in \mathbb{N}$ a tree $\mathcal{T}_n^{\boldsymbol{k}}$: This tree has h levels and on the i-th level, every node has $k_i(n)$ many children. The leaves of the tree are identified with the variables $x_1, \dots, x_n$ (for this to be possible, we need that the product over the $k_i(n)$ is n). The automorphism group $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ is the group of all permutations of the nodes of $\mathcal{T}_n^{\boldsymbol{k}}$ that preserve the edges and non-edges of the tree. A circuit is $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric circuit if for every $\pi \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$, there is an automorphism of the circuit that permutes the inputs $x_1, \dots, x_n$ precisely as π permutes the leaves of $\mathcal{T}_n^{\boldsymbol{k}}$ (see Section 2 for details). Note that not for every permutation $\pi \in \mathbf{Sym}_n$, there is a tree automorphism in $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ that acts like π on the leaves. Hence, $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetry is a laxer requirement of circuits than $\mathbf{Sym}_n$-symmetry.

Our next result extends Theorem 1 to the nested block symmetric setting and pins down the asymptotic circuit size exactly, depending only on the choice of symmetry group $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$. Theorem 1 is in fact a special case of Theorem 2, when $\mathcal{T}_n^{\boldsymbol{k}}$ is taken to be the depth-1 tree with n leaves. Nevertheless, Theorem 1 is important enough to be stated separately, and its proof is more instructive.

▶ **Theorem 2.** *Fix an integer $m \geq 6$ with at least $r \geq 2$ distinct prime divisors. Moreover, fix a constant depth $h \in \mathbb{N}$, and a tuple $\boldsymbol{k} = (k_1(n), \dots, k_h(n))$ of block sizes such that $\prod_i k_i(n) = n$ for all $n \in \mathbb{N}$. Assume that $k_i(n) > 8$ for each $i \in [h]$ and all large enough $n \in \mathbb{N}$. Let $k_{max}(n) \coloneqq \max_{i\in[h]} k_i(n)$.*
Every $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric MOD_m-circuit that computes the Boolean function AND_n has size at least $2^{\Omega(k_{max}(n)^{1/r}\cdot\log(k_{max}(n)))}$, and there exists a $\mathrm{CC}_{2h}[m]$-circuit that achieves this bound.

The upper bound is achieved by applying the construction from Theorem 1 in a recursive fashion, which requires 2 circuit layers for each nested block of the symmetry group. An immediate consequence of the theorem is that one cannot gain savings in size by varying the block sizes on different levels of the symmetry group. Thus, the symmetric circuit complexity is just controlled by the choice of h in this setting:

▶ **Corollary 3.** *For $h \in \mathbb{N}$ fixed, the choice of $\boldsymbol{k}$ which optimizes the size of an $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric MOD_m-circuit for AND_n is $k_1(n) = k_2(n) = \cdots = k_h(n) = n^{1/h}$. For this $\boldsymbol{k}$, there exists such a depth-$2h$ circuit of size $2^{\Theta(n^{1/(h\cdot r)}\cdot\log n)}$.*

Proof. Since $\prod_{i\in[h]} k_i(n) = n$ for all $n \in \mathbb{N}$, the smallest value that $k_{\max}(n)$ can attain is $n^{1/h}$, in case that all $k_i(n)$ are equal. ◀

Strikingly, the $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric construction from [24, Proposition 4.3] is slightly larger than this. It achieves only size (approximately) $2^{\Theta(n^{1/(h\cdot(r-1))}\cdot\log n)}$ instead of $2^{\Theta(n^{1/(h\cdot r)}\cdot\log n)}$ as in Corollary 3. This is because [24, Proposition 4.3] aims at an optimized depth: Indeed, the authors manage to compress the circuits down to only $h+1$ layers, which is essentially just one layer per nesting depth of the symmetry group and likely optimal under $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetry. In view of our Corollary 3, we consider it an important open problem to improve the size of this depth-$(h+1)$ construction so that it matches our lower bound, or to show that this is impossible.

Our results confirm the $2^{\Omega(n^{\varepsilon})}$ lower bound conjectured by ESH for fully symmetric and nested block symmetric circuits. Since much of the literature focuses on the AND_n-function, we have chosen to do the same, but our results can be proved for every symmetric function that is *aperiodic* (for a definition, see Section 2.3), such as OR_n or MAJ_n.

Our work raises the following immediate **open questions:**

1. Is there a *size-depth tradeoff* for nested block symmetric circuits?
   The current knowledge suggests this: We have size-optimal constructions that are at least a factor of 2 away from the optimal depth (Corollary 3), and we have (likely) depth-optimal circuits of slightly suboptimal size [24, Proposition 4.3].
2. What is the smallest possible symmetry group for which our method works? Do lower bounds for symmetric circuits inform the search for general (non-symmetric) lower bounds? For example, can every non-symmetric CC^0-circuit for AND_n be symmetrized at only a small cost?

Related work on modular circuits

The complexity of low-depth CC^0-circuits has particular relevance because of its various connections to other questions, of which we can only mention a few here. For example, it is known that strong lower bounds against low-depth CC^0-circuits would imply faster algorithms for solving equations over solvable groups [26], for circuit satisfiability problems over algebras [23, 25], and for constraint satisfaction problems with global modular constraints [5]. Another surprising application is in coding theory. Techniques from the construction of small CC^0-circuits for Boolean functions have been used to obtain explicit Ramsey-style graphs [17, 18]. These are crucial for example in the design of particularly good locally decodable error-correcting codes [16, 13] and private information retrieval schemes [14].

Related work on symmetric circuits

The idea to study symmetric circuits to facilitate lower bounds has been employed successfully in many different contexts in recent years. To name only a few examples, there are symmetry-based lower bounds for constant depth formulas with Boolean and majority gates [22], lower bounds for uniform symmetric Boolean (threshold) circuit families via a connection to fixed-point logics from finite model theory [1], and a recent research strand on symmetry in algebraic complexity theory [10, 11, 9, 15], notably proving the permanent polynomials to be exponentially hard for symmetric circuits.

For our purposes, the most relevant related work is a very recent paper by Kawałek and Weiß [27]. They seem to be the first to consider symmetry in the context of CC^0, but with a scope limited to $\mathrm{MOD}_q \circ \mathrm{MOD}_p \circ \mathrm{AND}_d$-circuits (where d is some fixed integer), and $\mathbf{Sym}_n$ as the symmetry group. Our lower bounds cover that case (when m has p and q as prime divisors) and apply to a much more general circuit model. In particular, Kawałek and Weiß suggest that to obtain smaller symmetric circuits for AND_n, one may have to increase the depth. Our Theorem 1 surprisingly refutes this, and shows that a depth greater than 2 yields no size improvement, unless also the symmetry constraint is relaxed as in Theorem 2.

Our techniques

The key challenge we solve is to overcome the obstacles that prevent the technique in [27] from generalising to symmetric circuits of arbitrary depth and with MOD_m-gates for composite numbers m. We accomplish this by using a very different technical framework, based on the

group-theoretic notion of *supports*. This is the central tool in the aforementioned articles by Dawar and others, and we use it in an inductive approach vaguely similar to the one in [9]. The lower bound for AND_n is then based on a periodicity argument as in [27]: Our main technical result, Lemma 12, shows that symmetric MOD_m-circuits of small (support) size necessarily compute periodic functions – but AND_n is aperiodic.

Another contribution of this work is the extension of the group-theoretic toolkit for dealing with the smaller symmetry groups $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$. Thus far, the literature has mostly focused on direct products of symmetric or alternating groups, which are technically easier to handle.

2 Preliminaries

We write $[n] = \{1, \dots, n\}$. For a tuple $\bar{x} = (x_1, \dots, x_n)$ indexed by $[n]$, and a subset $I \subseteq [n]$ of the indices, we use the notation $\bar{x}_I$ to refer to the subtuple $(x_i)_{i \in I}$ of $\bar{x}$ consisting of the entries with indices in I. In this notation, we also sometimes merge subtuples: For $I, J \subseteq [n]$, we denote by $\bar{x}_I \bar{x}_J$ the subtuple $\bar{x}_{I \cup J}$ of $\bar{x}$. In all these cases, the ordering of the respective subtuple of $\bar{x}$ is inherited from $\bar{x}$.

The *depth* of a rooted tree or DAG is the maximum number of edges on any path from the root to a leaf.

2.1 Permutation groups and supports

We write $\mathbf{Sym}_n$ for the symmetric group acting on the set $[n]$, and if A is a set, then $\mathbf{Sym}(A)$ denotes the symmetric group acting on A. For $S \subseteq [n]$, we write $\mathbf{Stab}(S) \leq \mathbf{Sym}_n$ for the setwise stabiliser subgroup of S in $\mathbf{Sym}_n$, and $\mathbf{Stab}^\bullet(S) \leq \mathbf{Sym}_n$ for the pointwise stabiliser of S. The setwise stabiliser is the subgroup of permutations π such that $\pi(S) = S$, whereas the pointwise stabiliser is the subgroup consisting of all $\pi \in \mathbf{Sym}_n$ such that $\pi(s) = s$ for every $s \in S$. This notion can be restricted to subgroups $\Gamma \leq \mathbf{Sym}_n$ as well: $\mathbf{Stab}_\Gamma(S) \leq \Gamma$ and $\mathbf{Stab}^\bullet_\Gamma(S) \leq \Gamma$ denote the respective subgroups of Γ that fix S setwise or pointwise.

For a group $\Gamma \leq \mathbf{Sym}_n$, and an element $a \in [n]$, the Γ-*orbit* of a is the set $\mathbf{Orb}_\Gamma(a) \coloneqq \{\pi(a) \mid \pi \in \Gamma\}$ of all possible images of a. By the well-known Orbit-Stabiliser Theorem, $|\mathbf{Orb}_\Gamma(a)| = \frac{|\Gamma|}{|\mathbf{Stab}_\Gamma(a)|}$. The notion of an orbit also applies to subsets of $[n]$, or more generally, other objects, such as gates of a circuit, that Γ may be acting on.

A set $S \subseteq [n]$ is a *support* of a group $\Gamma \leq \mathbf{Sym}_n$ if $\mathbf{Stab}^\bullet(S) \leq \Gamma$. It is known that every $\Gamma \leq \mathbf{Sym}_n$ that has a support of size $< n/2$ has a *unique minimal* support, denoted $\mathrm{sup}(\Gamma)$ [4, Lemma 26]. This notion will be of central importance in our analysis of symmetric circuits: In symmetric circuits of bounded size, also the minimal supports of the stabiliser groups of the gates will have bounded size. The function computed by a gate can then be described in terms of this small support.

Nested symmetric groups

Besides $\mathbf{Sym}_n$, we deal with nested symmetric groups. These are best described as the automorphism groups of rooted trees. Fix a depth $h \in \mathbb{N}$ and let $\boldsymbol{k} = (k_1(n), \dots, k_h(n))$, where for each $i \in [h]$, $k_i \colon \mathbb{N} \to \mathbb{N}$ is a function such that $\prod_{i \in [h]} k_i(n) = n$ for every $n \in \mathbb{N}$. The tuple $\boldsymbol{k}$ defines a family of rooted symmetric trees, one for each n: The n-th tree $\mathcal{T}_n^{\boldsymbol{k}}$ has h levels, and $k_i(n)$ is the number of children of every node on level i. The n leaf nodes are on level 0, and the root of the tree is the only node on level h. The automorphism group of $\mathcal{T}_n^{\boldsymbol{k}}$ is denoted $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$. It acts on the vertex set $V(\mathcal{T}_n^{\boldsymbol{k}})$. Each $\pi \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$

maps every subtree of $\mathcal{T}_n^{\boldsymbol{k}}$ to a subtree rooted at the same level, possibly permuting subtrees further down. Because in each level, all nodes have the same number of children, every node can be mapped to every other node on the same level by an automorphism in $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$. Formally, $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ is isomorphic to an iterated wreath product of symmetric groups, defined inductively as follows. Let T be a subtree of $\mathcal{T}_n^{\boldsymbol{k}}$ of depth 1, having $k_1(n)$ leaves. Then $\mathbf{Aut}(T) \cong \mathbf{Sym}_{k_1(n)}$. Now assume T is a subtree of $\mathcal{T}_n^{\boldsymbol{k}}$ of depth $i > 1$. Then the root v of T has $k_i(n)$ children, and $\mathbf{Aut}(T) \cong \mathbf{Aut}(T') \wr \mathbf{Sym}_{k_i(n)}$, where T' is isomorphic to the subtree rooted at any/every child of v.

The group $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ embeds into $\mathbf{Sym}_n$ by identifying every $\pi \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ with the permutation that it induces on the leaves of the tree. Note that every $\sigma \in \mathbf{Sym}_n$ is induced by at most one $\pi \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$.

For the analysis, it will be helpful to speak of the action of $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ on *blocks*: For a node $v \in V(\mathcal{T}_n^{\boldsymbol{k}}), B(v) \subseteq V(\mathcal{T}_n^{\boldsymbol{k}})$ denotes the block of v, by which we mean the set of all nodes (including v) that have the same parent as v. Every $\pi \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ stabilises a block setwise or moves it to another block on the same level. The set of all blocks is denoted $\mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}}) \coloneqq \{B(v) \mid v \in V(\mathcal{T}_n^{\boldsymbol{k}})\}$.

For $i \in [h] \cup \{0\}$, we denote by $L_i \subseteq V(\mathcal{T}_n^{\boldsymbol{k}})$ the nodes in the i-th level of the tree. For a set of nodes $W \subseteq V(\mathcal{T}_n^{\boldsymbol{k}})$, we denote by $L_0(W) \subseteq L_0$ the set of all leaves w that have an ancestor (i.e. node on a path from the root to w) in W.

Supports for nested symmetric groups

Any group $\Gamma \leq \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ can be embedded into $\mathbf{Sym}_n$ (via its action on the leaves) and thus admits a notion of support in the sense described previously. However, when Γ is explicitly presented as a subgroup of $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$, we use a more refined notion, that we call *blockwise support*. It breaks up the support according to the different copies of symmetric groups in $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$. For a block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, let $\Gamma|_B \leq \mathbf{Sym}(B)$ denote the permutation group on B consisting of all $\pi \in \mathbf{Sym}(B)$ such that there exists a $\sigma \in \Gamma$ that fixes B setwise and satisfies $\sigma|_B = \pi$, i.e. σ permutes the nodes in B like π does.

A set $S \subseteq B$ is a *B-support* of $\Gamma \leq \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ if $\mathbf{Stab}^{\bullet}_{\mathbf{Sym}(B)}(S) \leq \Gamma|_B$. In other words, this means that for every $\pi \in \mathbf{Stab}^{\bullet}_{\mathbf{Sym}(B)}(S)$, there exists at least one permutation $\sigma \in \Gamma$ that acts like π on B.

2.2 (Symmetric) modular circuits

A circuit is a DAG, possibly with multiedges, and a single designated root. Its nodes are called gates and its edges are also called wires. Edges are directed from a gate where a value is computed towards the next gate that uses this value as an input. The nodes with no incoming edges are called input gates, and each input gate g is labelled with a variable $\ell(g) \in \{x_1, \ldots, x_n\}$, where n is the arity of the function to be computed by the circuit. Each internal gate is labelled with an operation. In this paper, we only consider circuits with modular counting gates: For $m \in \mathbb{N}$, and $R \subseteq \mathbb{Z}_m$, the operation MOD_m^R is of arbitrary fan-in k, and satisfies

$$\mathrm{MOD}_m^R(x_1, \ldots, x_k) = \begin{cases} 1 & \text{if } (\sum_{i \in [k]} x_i \mod m) \in R \\ 0 & \text{otherwise} \end{cases}$$

A MOD_m-*circuit* is one where every internal gate is labelled with the operation MOD_m^R for an arbitrary $R \subseteq \mathbb{Z}_m$. The computation result is the Boolean value that is computed at the root.

We assume throughout that for every variable x_i, there exists exactly one input gate with label x_i. This is not a restriction since distinct input gates labelled with the same variable can always be identified.

The *size* of a circuit C is $|C| \coloneqq |V(C)| + |E(C)|$, the total number of gates plus wires, counted with multiplicities. For a gate g, we write $g(\bar{x})$ to denote the function from $\{x_1, \dots, x_n\}$ to $\{0,1\}$ that is computed by the subcircuit of C rooted at g.

Symmetric circuits

Let $n \in \mathbb{N}$, and $\Gamma \leq \mathbf{Sym}_n$ be a subgroup of $\mathbf{Sym}_n$. A MOD_m-circuit C is called Γ-*symmetric* if its set of input variables is $\{x_1, \dots, x_n\}$ and every $\pi \in \Gamma$ acting on the input gates *extends to an automorphism* of C. That is, for every $\pi \in \Gamma$, there exists a $\sigma \in \mathbf{Sym}(V(C))$ such that $\pi(\ell(g)) = \ell(\sigma(g))$ for all inputs gates $g \in V(C)$, and σ is an automorphism of C. This means that for every internal gate g, $\ell(g) = \ell(\sigma(g))$ (i.e., the gates compute the same operation MOD_m^R, for the same R), and for any two gates $g, h \in V(C)$, the multiplicity of the directed edge (g,h) is the same as of $(\sigma(g), \sigma(h))$. We call C *rigid* if for every $\pi \in \Gamma$, the circuit automorphism σ extending π is unique. This is equivalent to C not having any non-trivial automorphism that fixes every input gate. Fortunately, w.l.o.g. we can always assume our symmetric circuits to be rigid (see e.g. [8, Lemma 4.3] or Lemma 25 in the appendix).

The advantage of working with rigid Γ-symmetric circuits is that for every $\pi \in \Gamma$, and $g \in V(C)$, we may write $\pi(g)$ to mean the well-defined gate $\sigma(g)$, for the unique circuit automorphism σ that extends π. In this sense, if C is rigid, then Γ has a well-defined (faithful) action on $V(C)$.

With respect to this action, we can speak about the orbits of gates. Their size is an important complexity measure of Γ-symmetric rigid circuits, called *orbit size*, by which we mean the maximum size of a Γ-orbit of any gate:

$$\mathrm{maxOrb}_\Gamma(C) \coloneqq \max_{g \in V(C)} |\mathbf{Orb}_\Gamma(g)|.$$

Note that for any gate g, $\mathbf{Orb}_\Gamma(g) \subseteq V(C)$, so to establish lower bounds on the circuit size $|V(C)|$, it is sufficient to prove lower bounds for $\mathrm{maxOrb}_\Gamma(C)$.

By induction on the circuit structure, it is not difficult to verify the following fact about the interplay between symmetry and the semantics of the gates (see Appendix A).

▶ **Lemma 4.** *Let C be a Γ-symmetric rigid circuit, for $\Gamma \leq \mathbf{Sym}_n$. Let $\delta \colon \{x_1, \dots, x_n\} \to \{0,1\}$ be an assignment to the variables, let $g \in V(C)$ be a gate, and let $\pi \in \Gamma$. Then*

$$g(\delta(x_1), \dots, \delta(x_n)) = \pi(g)(\delta(\pi^{-1}(x_1)), \dots, \delta(\pi^{-1}(x_n))).$$

Supports in symmetric circuits

We apply the previously introduced notions of supports to stabiliser groups of gates in symmetric circuits. We have already seen that the concept of a support can be defined differently for different permutation groups. Therefore, from here on, we restrict the symmetry group Γ of our circuits and always assume that $\Gamma \in \{\mathbf{Sym}_n, \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})\}$, for some n or some symmetric n-leaf tree $\mathcal{T}_n^{\boldsymbol{k}}$.

We are interested in the *support of a gate* g in a given Γ-symmetric rigid circuit C. Let $\mathbf{Stab}(g) \leq \Gamma$ be the stabiliser group of the gate, that is, $\mathbf{Stab}(g) \coloneqq \{\pi \in \Gamma \mid \pi(g) = g\}$.

▶ **Definition 5** (Supports of gates). *Let $g \in V(C)$ be a gate in a Γ-symmetric rigid circuit, for $\Gamma \in \{\mathbf{Sym}_n, \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})\}$.*

- *If $\Gamma = \mathbf{Sym}_n$, then the support of g is $\sup(g) \coloneqq \sup(\mathbf{Stab}_{\mathbf{Sym}_n}(g)) \subseteq [n]$, i.e., the unique minimal support of $\mathbf{Stab}_{\mathbf{Sym}_n}(g)$ in $\mathbf{Sym}_n$.*
- *If $\Gamma = \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$, then we consider the* blockwise support *of g: For every $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, we denote by $\sup_B(g) \subseteq B$ the unique minimal B-support of the group $\mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})}(g)$.*

In either case, the support of g is undefined if the respective minimal (B-)support of $\mathbf{Stab}(g)$ is not uniquely defined.

We still need to argue that in the scenarios we study in this paper, the respective minimal supports exist and are unique, so that $\sup(g)$ and $\sup_B(g)$ are indeed always defined when we need them.

▶ **Lemma 6.**

1. *Let $n > 8$ and C be a $\mathbf{Sym}_n$-symmetric rigid circuit. Let k be such that $1 \leq k \leq \frac{n}{4}$ and $\mathrm{maxOrb}_{\mathbf{Sym}_n}(C) \leq \binom{n}{k}$. Then for every $g \in V(C)$, $\mathbf{Stab}_{\mathbf{Sym}_n}(g)$ has a support of size less than k.*
2. *Let C be an $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric rigid circuit, for some h-tuple $\boldsymbol{k}$ such that $k_{min} \coloneqq \min_{i\in[h]} k_i(n) > 8$. For every block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, let k_B be such that $1 \leq k_B \leq \frac{|B|}{4}$ and $\mathrm{maxOrb}_{\mathbf{Stab}(B)}(C) \leq \binom{|B|}{k_B}$. Then for every $g \in V(C)$, $\mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})}(g)$ has a B-support $S \subseteq B$ with $|S| < k_B$.*

The first part of the lemma is shown in [12, Theorem 14] for symmetric Boolean circuits, and the set-up here is very similar. The second item follows from the first with an additional argument. The proof details are given in Appendix A.

As already mentioned, by [4, Lemma 26], every subgroup of $\mathbf{Sym}_n$ that has a support of size $< n/2$ also has a unique minimal support. This holds in particular whenever the conditions of the above lemma are satisfied, both in case (1) and (2). In the set-up in the following sections, this will always be the case because if the conditions of the lemma are not satisfied for a circuit C, then $\mathrm{maxOrb}(C)$, and hence $|V(C)|$, is anyway at least as large as the circuit size lower bounds we have to prove for Theorem 1 and Theorem 2. For a circuit C where the respective supports are defined, we write

$$\mathrm{maxSup}(C) \coloneqq \max_{g\in V(C)} |\sup(g)|, \text{ if } C \text{ is } \mathbf{Sym}_n\text{-symmetric.}$$

$$\mathrm{maxSup}_B(C) \coloneqq \max_{g\in V(C)} |\sup_B(g)|, \text{ if } C \text{ is } \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})\text{-symmetric and } B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}}).$$

Another known fact about supports of gates in symmetric circuits is that they are moved by permutations in the expected way:

▶ **Lemma 7** ([8, Lemma 4.2]). *Let C be a $\mathbf{Sym}_n$-symmetric rigid circuit in which the supports of all gates are defined. Let $g \in V(C)$ be a gate. Then for every $\pi \in \mathbf{Sym}_n$, $\sup(\pi(g)) = \pi(\sup(g))$.*

Analogously, in every $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric circuit C, $\sup_{\pi(B)}(\pi(g)) = \pi(\sup_B(g))$ for every gate g, every block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, and every $\pi \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$.

2.3 Periodic functions

A function $f\colon \mathbb{N} \to \mathbb{N}$ is called *periodic* with a period of length ℓ if $f(x) = f(x + \ell)$ for every $x \in \mathbb{N}$. An analogous definition applies if f is only defined on an initial segment of $\mathbb{N}$. A particular periodic function that will be of high importance for us is the binomial coefficient modulo m. Its period length is known exactly:

▶ **Theorem 8** ([28, Theorem 2.3]). *Let $m \in \mathbb{N}$ be fixed, and let $p_1, \dots, p_r$ be its prime divisors. Fix $x \in \mathbb{N}$. The function $a(n) \coloneqq \binom{n}{x} \bmod m$ has a period of minimal length $\ell(m,x) = m \cdot \prod_{i\in[r]} p_i^{\lfloor \log_{p_i}(x) \rfloor}$.*

Let $f\colon \{0,1\}^n \to \{0,1\}$ be an n-ary Boolean function. If for every $\pi \in \mathbf{Sym}_n$, and every $\bar{x} \in \{0,1\}^n$, $f(x_1,\dots,x_n) = f(\pi(\bar{x})) \coloneqq f(x_{\pi^{-1}(1)},\dots,x_{\pi^{-1}(n)})$, then we say that f is $\mathbf{Sym}_n$*-symmetric*. The value $f(\bar{x})$ of a $\mathbf{Sym}_n$-symmetric n-ary function f only depends on the number of 1-entries in $\bar{x}$, denoted $|\bar{x}|_1$. Thus, we may view $f(\bar{x})$ as a unary function $f(|\bar{x}|_1)$ defined on $\{0,\dots,n\}$, and as such, we can speak of its period.

▶ **Lemma 9.** *The $\mathbf{Sym}_n$-symmetric Boolean function AND_n does not have a period of any length $\leq n$.*

Proof. For a contradiction, assume that $0 < \ell \leq n$ was the period length of AND_n. Then $\mathrm{AND}_n(\bar{x}) = 1$ also in case that $|\bar{x}|_1 = n - \ell$. But $0 \leq n - \ell < n$, so $\mathrm{AND}_n(\bar{x}) = 0$ in this case. Contradiction. ◀

3 Size lower bound for fully symmetric circuits

In this section, we prove the size lower bound from Theorem 1 for $\mathbf{Sym}_n$-symmetric MOD_m-circuits computing AND_n. The technical core, from which the size lower bound follows, is a lower bound on the *support size* $\mathrm{maxSup}(C)$ required to compute AND_n:

▶ **Theorem 10.** *Fix a positive integer $m > 3$ and let r be the number of distinct prime divisors of m. Let $(C_n)_{n\in\mathbb{N}}$ be a family of $\mathbf{Sym}_n$-symmetric rigid MOD_m-circuits. If $\mathrm{maxSup}(C_n) < (n/m)^{1/r}$ for all $n \in \mathbb{N}$, then C_n does not compute AND_n.*

▶ **Corollary 11.** *In the setting of the theorem, if C_n computes AND_n for an $n > 8$, then $|V(C_n)| \geq \binom{n}{(n/m)^{1/r}}$.*

Proof. If C_n computes AND_n, then by Theorem 10, $\mathrm{maxSup}(C_n) \geq (n/m)^{1/r}$. Suppose for a contradiction that $|V(C_n)| < \binom{n}{(n/m)^{1/r}}$. Then in particular, $\mathrm{maxOrb}(C_n) < \binom{n}{(n/m)^{1/r}}$. Because $n > 8$ and $m > 3$, the conditions of Lemma 6 (1) are fulfilled for $k = (n/m)^{1/r}$, so $\mathrm{maxSup}(C_n) < (n/m)^{1/r}$, which is a contradiction. ◀

By replacing factorials with their Stirling approximations, we can compute that $\binom{n}{(n/m)^{1/r}} \geq (f_1(m,r) \cdot n^{f_2(m,r)})^{n^{1/r}}$, where f_1, f_2 are functions depending on m, r, so we can treat them as constants. Thus, Corollary 11 indeed yields the asymptotic circuit size lower bound of $2^{\Omega(n^{1/r}\cdot \log n)}$ that is claimed in Theorem 1.

It remains to prove Theorem 10. This is done by showing that if $\mathrm{maxSup}(C_n) < (n/m)^{1/r}$, then the function computed by C_n is periodic with a period of length $< n$.

The proof rests on the upper bound on the period length shown in Corollary 13. Before we can state and prove that corollary, we need to introduce a refined notion of period, for not fully symmetric functions:

Let $S \subseteq [n]$, and let $f(x_1,\dots,x_n)$ be a $\mathbf{Stab}^\bullet(S)$-symmetric function. Let $X_S \coloneqq \{x_i \mid i \in S\}$. Then the value of f is fully determined by the assignment $\alpha\colon X_S \to \{0,1\}$ and by the number of 1-entries in the variables $\{x_i \mid i \in [n] \setminus S\}$. Formally, let us write $f_\alpha\colon X_{[n]\setminus S} \to \{0,1\}$ for the function obtained from f by fixing the variables in X_S to the values given by α. That is,

$$f_\alpha(\bar{x}_{[n]\setminus S}) \coloneqq f(\alpha(\bar{x}_S)\bar{x}_{[n]\setminus S}).$$

When we say that a $\mathbf{Stab}^\bullet(S)$-symmetric function f has a period, we mean that for every $\alpha\colon X_S \to \{0,1\}$, the function $f_\alpha(\bar{x}_{[n]\setminus S})$, whose value only depends on the number of 1-entries in its input, has a period. This period may be of different length for different assignments $\alpha\colon X_S \to \{0,1\}$, but when we say that f has a period of length at most ℓ, then we mean that for each $\alpha\colon X_S \to \{0,1\}$, the period length of f_α is at most ℓ.

Now let $g \in V(C)$ be a gate. The function $g(\bar{x})$ computed by g is always $\mathbf{Stab}(g)$-symmetric by Lemma 4. Because $\mathbf{Stab}^\bullet(\sup(g)) \leq \mathbf{Stab}(g)$, it is also $\mathbf{Stab}^\bullet(\sup(g))$-symmetric. So when we say that $g(\bar{x})$ has a period of length at most ℓ, we mean that $g_\alpha(\bar{x})$ has such a period for every $\alpha\colon X_{\sup(g)} \to \{0,1\}$. Now the technical lemma that we want to show reads as follows.

▶ **Lemma 12.** *Let $s\colon \mathbb{N} \to \mathbb{N}$ be a function in $o(n)$. Fix a number $m \in \mathbb{N}$. Let $(C_n)_{n\in\mathbb{N}}$ be a family of $\mathbf{Sym}_n$-symmetric rigid MOD_m-circuits in which all supports have size at most $s(n)$. Fix $n \in \mathbb{N}$. Let g be a gate in C_n. Then the $\mathbf{Stab}^\bullet(\sup(g))$-symmetric function $g(\bar{x})$ has a period of length at most $q(m,s) \coloneqq m \cdot \prod_{i\in[r]} p_i^{\lfloor \log_{p_i}(s(n)) \rfloor}$, where the product ranges over the prime factors $p_1, \ldots, p_r$ of m.*

Note that this bound on the period length does not depend on the depth of the gate – this also explains why it is possible already for depth-2 circuits to achieve the optimal size. All that matters is the size of the supports.

▶ **Corollary 13.** *In the setting of Lemma 12, the $\mathbf{Sym}_n$-symmetric function computed at the output gate of C_n has a period of length at most $m \cdot s(n)^r$, where r denotes the number of distinct prime divisors of m.*

The proof of Lemma 12 is given in the full version [29]. It proceeds by induction from the input gates to the root of the circuit. The periodic behaviour of the function computed at each gate is essentially shown by reducing the evaluation of the gate to an expression that involves binomial coefficients modulo m. This is possible because the children of the gate are symmetric to each other in a certain precise sense. The period length claimed in Lemma 12 comes from the periodicity of the binomial coefficients, as given by Theorem 8.

Proof of Theorem 10. If $\mathrm{maxSup}(C_n) < (n/m)^{1/r}$, then by Corollary 13, the $\mathbf{Sym}_n$-symmetric function computed by C_n has a period of length at most $m \cdot ((n/m)^{1/r})^r = n$. But then, this function cannot be AND_n by Lemma 9. ◀

4 Size lower bound for nested block symmetry

In this section, fix $h \in \mathbb{N}$ and a tuple $\boldsymbol{k} = (k_1(n), \ldots, k_h(n))$ such that $\prod_{i\in[h]} k_i(n) = n$ for each $n \in \mathbb{N}$. For every $n \in \mathbb{N}$, let $k_{\min}(n) \coloneqq \min_{i\in[h]} k_i(n)$ and $k_{\max}(n) \coloneqq \max_{i\in[h]} k_i(n)$ denote the smallest and largest block sizes in the tree $\mathcal{T}_n^{\boldsymbol{k}}$.

We now show how to adapt the proof from the previous section to obtain the circuit size lower bound claimed in Theorem 2 for $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric circuits. The main technical result from which the lower bound can be derived is the following variation of Theorem 10:

▶ **Theorem 14.** *Fix a positive integer $m > 3$ and let r be the number of distinct prime divisors of m. Let $(C_n)_{n\in\mathbb{N}}$ be a family of $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric MOD_m-circuits. Let $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$ be a block that has size $|B| = k_j(n)$, for some $j \in [h]$. [2] If $\mathrm{maxSup}_B(C_n) < (k_j(n)/m)^{1/r}$, then C_n does not compute AND_n.*

[2] Note that we can fix the block B independently of n since the structure of $\mathcal{T}_n^{\boldsymbol{k}}$ only depends on $\boldsymbol{k}$, and n just controls the size of B.

▶ **Corollary 15.** *In the setting of the theorem, if C_n computes AND_n for an n such that $k_{min}(n) > 8$, then $|V(C_n)| \geq \binom{k_{max}(n)}{(k_{max}(n)/m)^{1/r}}$.*

Proof. If $C \coloneqq C_n$ computes AND_n, then by Theorem 14, for every block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, $\mathrm{maxSup}_B(C) \geq (|B|/m)^{1/r}$. Then for every $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, $\mathrm{maxOrb}_{\mathbf{Stab}(B)}(C) \geq \binom{|B|}{(|B|/m)^{1/r}}$ by Lemma 6 (2). Since $|\mathrm{maxOrb}_{\mathbf{Stab}(B)}(C)| \leq |V(C)|$, for every $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, we can pick a block of maximal size and obtain $|V(C)| \geq \binom{k_{\max}(n)}{(k_{\max}(n)/m)^{1/r}}$. ◀

Just like in the last section, the size lower bound from this corollary translates into

$$|V(C_n)| \geq 2^{\Omega(k_{\max}(n)^{1/r} \cdot \log(k_{\max}(n)))},$$

which is what is stated in Theorem 2. Similarly as in the previous section, Theorem 14 is proved by showing that if the supports are not big enough, then the functions computed by the gates have a certain periodic behaviour. However, the notion of period is different now because it has to match the different notion of symmetry.

Recall from Section 2.1 that for a tree $\mathcal{T}_n^{\boldsymbol{k}}$, and a set $W \subseteq V(\mathcal{T}_n^{\boldsymbol{k}})$ of nodes, $L_0(W)$ denotes the set of leaves in subtrees rooted at nodes in W.

▶ **Definition 16** (Block-periodic functions)**.** *Let $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$. Let $\Gamma \leq \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ be a group such that for every $\pi \in \mathbf{Sym}(B)$, there is a $\sigma \in \Gamma$ that fixes B setwise and satisfies $\sigma|_B = \pi$. Let $f(x_1, \ldots, x_n)$ be a Γ-symmetric function. Let $\beta \colon \{x_1, \ldots, x_n\} \to \{0, 1\}$ be an assignment such that for each $v \in B$, $\beta(\bar{x}_{L_0(v)}) \in \{\bar{0}, \bar{1}\}$, i.e., β is constant on each set $X_{L_0(v)}$, for each $v \in B$. Then by Γ-symmetry of f, the value of $f(\beta(\bar{x}))$ depends only on $\beta(\bar{x}_{[n]\setminus L_0(B)})$ and on the number*

$$|\beta(\bar{x})|_1^B \coloneqq |\{v \in B \mid \beta(\bar{x}_{L_0(v)}) = \bar{1}\}|.$$

*We say that f has a B-*period *of length ℓ if*

$$f(\beta(\bar{x})) = f(\beta'(\bar{x})),$$

for any two assignments β, β' that are constant on $X_{L_0(v)}$, for each $v \in B$, and satisfy $|\beta'(\bar{x})|_1^B = |\beta(\bar{x})|_1^B + \ell$, and $\beta(\bar{x}_{[n]\setminus L_0(B)}) = \beta'(\bar{x}_{[n]\setminus L_0(B)}) \in \{\bar{0}, \bar{1}\}$.

▶ **Lemma 17.** *Let B, Γ and f be as in Definition 16. If $f(x_1, \ldots, x_n)$ has a B-period of length $1 \leq \ell \leq |B|$, then $f \neq \mathrm{AND}_n$.*

Proof. Suppose for a contradiction that $f = \mathrm{AND}_n$. Then $f(\bar{1}) = 1$. Consider an assignment $\beta \colon \{x_1, \ldots, x_n\} \to \{0, 1\}$, which is 1 everywhere except that precisely for ℓ nodes $v \in B$, $\beta(\bar{x}_{L_0(v)}) = \bar{0}$. Then $f(\beta(\bar{x})) = 1$ because f has a B-period of length ℓ. But this is a contradiction because $\mathrm{AND}_n(\beta(\bar{x})) \neq 1$. ◀

Again, it remains to prove the key technical ingredient, Corollary 19, and again, this requires to adjust the notion of periodicity to stabiliser groups of gates that fix the support pointwise.

Let $S \subseteq B$, let $\Gamma \leq \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ be a group such that for every $\pi \in \mathbf{Stab}^{\bullet}_{\mathbf{Sym}(B)}(S)$, there is a $\sigma \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ that fixes B setwise and satisfies $\sigma|_B = \pi$. We now consider $\mathbf{Stab}^{\bullet}_{\Gamma}(S) \leq \Gamma$, the pointwise stabiliser of S in Γ. Let $f(x_1, \ldots, x_n)$ be a $\mathbf{Stab}^{\bullet}_{\Gamma}(S)$-symmetric function. Fix an assignment $\alpha \colon X_{[n]\setminus L_0(B\setminus S)} \to \{0, 1\}$ which is constant on each set $X_{L_0(v)}$ for each $v \in S$, and constant on $X_{[n]\setminus L_0(B)}$. Now when α is regarded as fixed, then for assignments $\beta \colon X_{L_0(B\setminus S)} \to \{0, 1\}$ that are constant on each set $X_{L_0(v)}$ for each $v \in B \setminus S$, the value of $f(\alpha\beta(\bar{x}))$ only depends on $|\beta(\bar{x})|_1^B$.

Analogously to the previous section, we write $f_\alpha \colon X_{L_0(B\setminus S)} \to \{0,1\}$ for the function obtained from f by fixing the variables in $X_{[n]\setminus L_0(B\setminus S)}$ to the values given by α. That is,

$$f_\alpha(\bar{x}_{L_0(B\setminus S)}) \coloneqq f(\alpha(\bar{x}_{[n]\setminus L_0(B\setminus S)})\bar{x}_{L_0(B\setminus S)}).$$

When we say that a $\mathbf{Stab}^\bullet_\Gamma(S)$-symmetric function f has a B-period, we mean that for every $\alpha \colon X_{[n]\setminus L_0(B\setminus S)} \to \{0,1\}$ that is constant on $X_{L_0(v)}$ for each $v \in S$, and constant on $X_{[n]\setminus L_0(B)}$, the function $f_\alpha(\bar{x}_{L_0(B\setminus S)})$ has a B-period.

Now let $g \in V(C)$ be a gate and let $\Gamma \coloneqq \mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}^{\boldsymbol{k}}_n)}(g)$. By the definition of $\sup_B(g)$, we know that for every $\pi \in \mathbf{Stab}^\bullet_{\mathbf{Sym}(B)}(S)$, there is a $\sigma \in \mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}^{\boldsymbol{k}}_n)}(g)$ that fixes the set B setwise and satisfies $\sigma|_B = \pi$. Therefore, Γ has the properties we are assuming in the above paragraph. The function $g(\bar{x})$ computed by g is always Γ-symmetric by Lemma 4. Because $\mathbf{Stab}^\bullet_\Gamma(\sup_B(g)) \leq \Gamma$, it is also $\mathbf{Stab}^\bullet_\Gamma(\sup_B(g))$-symmetric. So when we say that $g(\bar{x})$ has a B-period of length at most ℓ, we mean that $g_\alpha(\bar{x}_{L_0(B\setminus S)})$ has such a period for every $\alpha \colon X_{[n]\setminus L_0(B\setminus \sup_B(g))} \to \{0,1\}$ that is constant on each $X_{L_0(v)}$ for each $v \in \sup_B(g)$, and constant on $X_{[n]\setminus L_0(B)}$. Now the technical lemma that we want to show reads as follows.

▶ **Lemma 18.** *Let $B \in \mathcal{B}(\mathcal{T}^{\boldsymbol{k}}_n)$ be a block of size $k_j(n)$, for some $j \in [h]$. Let $s \colon \mathbb{N} \to \mathbb{N}$ be a function in $o(n)$. Fix a number $m \in \mathbb{N}$. Let $(C_n)_{n\in\mathbb{N}}$ be a family of $\mathbf{Aut}(\mathcal{T}^{\boldsymbol{k}}_n)$-symmetric rigid MOD_m-circuits such that $\mathrm{maxSup}_B(C_n) < s(k_j(n))$ for all $n \in \mathbb{N}$. Let g be a gate in C_n and let $\Gamma \coloneqq \mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}^{\boldsymbol{k}}_n)}(g)$. Then the $\mathbf{Stab}^\bullet_\Gamma(\sup_B(g))$-symmetric function $g(\bar{x})$ has a B-period of length at most*

$$q(m,s) \coloneqq m \cdot \prod_{i\in[r]} p_i^{\lfloor \log_{p_i}(s(k_j(n)))\rfloor},$$

where the product ranges over the prime factors $p_1, \ldots, p_r$ of m.

▶ **Corollary 19.** *In the setting of Lemma 18, the $\mathbf{Aut}(\mathcal{T}^{\boldsymbol{k}}_n)$-symmetric function computed at the output gate of C_n has a B-period of length at most $m \cdot s(k_j(n))^r$, where r denotes the number of distinct prime divisors of m.*

The proof of Lemma 18 is analogous to the proof of Lemma 12, where we essentially "zoom in" on the group $\mathbf{Sym}(B)$ instead of performing the calculation for the symmetry group $\mathbf{Sym}_n$. Details can be found in [29].

Proof of Theorem 14. Assume the setting of Theorem 14, so in particular, $\mathrm{maxSup}_B(C_n) < (|B|/m)^{1/r}$ for some block $B \in \mathcal{B}(\mathcal{T}^{\boldsymbol{k}}_n)$. By Corollary 19, the $\mathbf{Aut}(\mathcal{T}^{\boldsymbol{k}}_n)$-symmetric function computed by C_n has a B-period of length at most $m \cdot ((|B|/m)^{1/r})^r = |B|$. But then, this function cannot be AND_n by Lemma 17. ◀

5 Symmetric circuit upper bounds

In this section, we present the upper bound constructions matching the lower bounds in Theorem 1 and Theorem 2. The fully symmetric depth-2 construction for Theorem 1 is entirely due to [24] and we include a summary of their proof for completeness in Section 5.1. The construction for Theorem 2 in Section 5.2 does not appear elsewhere in the literature but it is simply a recursive application of the depth-2 construction. In [24], a similar but more sophisticated recursive construction is presented, leading to a smaller depth at the cost of greater asymptotic size. Our construction here is a more naive variant of this, which is possibly not depth-optimal, but matches the size lower bound from Theorem 2.

5.1 Fully symmetric depth-2 construction

▶ **Theorem 20** ([24, Proposition 3.1]). *Fix $m \in \mathbb{N}$ with at least $r \geq 2$ distinct prime divisors of m. For every $n \in \mathbb{N}$, there is a $\mathbf{Sym}_n$-symmetric depth-2 MOD_m-circuit with $2^{\mathcal{O}(n^{1/r} \cdot \log n)}$ gates which computes AND_n.*

This symmetric construction was first provided for depth-3 circuits by Barrington, Beigel and Rudrich [2], and improved to depth 2 by Idziak, Kawałek, Krzaczkowski [24], and independently by Chapman and Williams [6]. We outline the proof from [24]. The main building block is what the authors call $\mathbb{Z}_{pq}$*-expressions*.

▶ **Definition 21** ($\mathbb{Z}_{pq}$-expressions). *Let p, q be two distinct primes. Let $n \in \mathbb{N}$. Let $b\colon \mathbb{Z}_p \to \mathbb{Z}_q$ be the function that maps 0 to 0 and every $x \in \mathbb{Z}_p \setminus \{0\}$ to $1 \in \mathbb{Z}_q$. An n-ary $\mathbb{Z}_{pq}$-*expression *is of the form*

$$\sum_{\beta \in \mathbb{Z}_p^n, c \in \mathbb{Z}} \alpha_{\beta,c} \cdot b\Big(\sum_{i=1}^{n} \beta_i x_i + c \mod p \Big) \mod q,$$

where the coefficients $\alpha_{\beta,c}$ are in $\mathbb{Z}_q$, the outer sum and the multiplications with the $\alpha_{\beta,c}$ are evaluated in $\mathbb{Z}_q$, while the expression inside b is evaluated in $\mathbb{Z}_p$.

It is straightforward to see that $\mathbb{Z}_{pq}$-expressions can be computed by modular counting circuits of depth 2, in a certain sense: Since the output of any MOD_m^R-gate is always Boolean, and the result of a $\mathbb{Z}_{pq}$-expression is in $\mathbb{Z}_q$, the only way in which we can realise such expressions as MOD_m-circuits is to have multiple output wires. The semantics is that the sum over the output wires modulo q is equal to the result of the $\mathbb{Z}_{pq}$-expression. Such a MOD_m-circuit with a set of designated output wires that are to be interpreted as a sum in $\mathbb{Z}_q$ is called a MOD_m-circuit of *output type* q henceforth. The depth of a circuit of output type q refers to the maximum number of wires along any path, so the layer consisting of the output wires counts towards the depth. With this definition it is straightforward to write $\mathbb{Z}_{pq}$-expressions as modular circuits:

▶ **Lemma 22.** *Let $p, q, m \in \mathbb{N}$ be integers such that m has p and q as prime factors. Every $\mathbb{Z}_{pq}$-expression can be realised by a depth-2 MOD_m-circuit of output type q.*

For a Boolean assignment β to variables $x_1, \ldots, x_n$, we write $\beta(\bar{x})$ for the tuple $(\beta(x_1), \ldots, \beta(x_n))$. By $|\beta(\bar{x})|_0$, we denote the number of 0s in this tuple. A particular $\mathbb{Z}_{pq}$-expression that is central for the proof of Theorem 20 is the following.

▶ **Lemma 23.** *Let $\nu \in \mathbb{N}$ and let q be a prime. The function $t_{q^\nu}(x_1, \ldots, x_n)$ which satisfies for all $\beta\colon \{x_1, \ldots, x_n\} \to \{0, 1\}$:*

$$t_{q^\nu}(\beta(\bar{x})) \coloneqq \begin{cases} 0 & \text{if } |\beta(\bar{x})|_0 \text{ is divisible by } q^\nu \\ 1 & \text{else} \end{cases}$$

is expressible as a $\mathbb{Z}_{pq}$-expression, for every prime $p \neq q$. Moreover, for every $m \in \mathbb{N}$ that has p and q as prime factors, this $\mathbb{Z}_{pq}$-expression can be realised as a depth-2 $\mathbf{Sym}_n$-symmetric circuit of output type q and of size at most $2^{\mathcal{O}(q^\nu \cdot \log n)}$.

Proof. This follows from [24, Lemma 3.5]. The fact that the circuit can be realised in a $\mathbf{Sym}_n$-symmetric way is not stated explicitly there, but can be seen by inspection of the proof. To be precise, the proof of [24, Fact 3.4] shows that t_{q^ν} is effectively expressed as a linear combination of elementary symmetric polynomials. Each of these polynomials is by definition $\mathbf{Sym}_n$-symmetric, and this carries over to the $\mathbb{Z}_{pq}$-expression representing them. ◀

To prove the upper bound result, we summarise the proof of [24, Proposition 3.1]:

Proof of Theorem 20. Let $p_1, \dots, p_r$ be the prime factors of m. Fix integers $\nu_1, \dots, \nu_r$ such that for each $j \in [r]$, we have $p_j^{\nu_j - 1} \leq n^{1/r} < p_j^{\nu_j}$. Let

$$T(\bar{x}) \coloneqq \sum_{j=1}^{r} \frac{m}{p_j} \cdot t_{p_j^{\nu_j}}(\bar{x}) \mod m.$$

One can show that $T(\beta(\bar{x})) = 0$ if and only if $\beta(x_i) = 1$ for every $i \in [n]$: If all $\beta(x_i)$ are equal to 1, then $|\beta(\bar{x})|_0 = 0$ is divisible by every prime power, so each $t_{p_j^{\nu_j}}$ will evaluate to 0, and hence $T(\beta(\bar{x})) = 0$. Conversely, assume that $T(\beta(\bar{x})) = 0$. This can only be the case if for all $j \in [r]$, $t_{p_j^{\nu_j}}(\beta(\bar{x})) = 0$. Then $|\beta(\bar{x})|_0$ is divisible by $\prod_{j \in [r]} p_j^{\nu_j} > n$. Since $|\beta(\bar{x})|_0 \leq n$, it follows that $|\beta(\bar{x})|_0 = 0$, which is what we had to show.

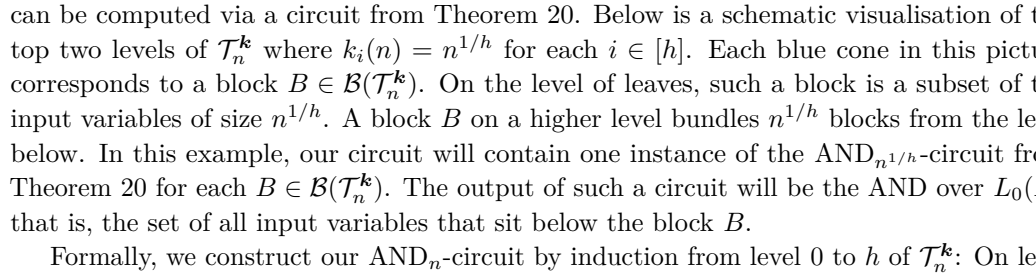
By Lemma 23, each $t_{p_j^{\nu_j}}$ can be expressed as a depth-2 $\mathbf{Sym}_n$-symmetric MOD_m-circuit C_j of output type p_j and of size at most $2^{\mathcal{O}(n^{1/r} \cdot \log n)}$. Thus, to compute AND_n, we connect the outgoing wires of the depth-2 symmetric circuits $C_1, \dots, C_r$ to an output gate $\mathrm{MOD}_m^{\{0\}}$ that sums up the values $t_{p_j^{\nu_j}}$ modulo m, with the respective coefficients $\frac{m}{p_j}$ realised by appropriate wire multiplicities, and outputs 1 if and only if $T(\beta(\bar{x})) = 0$. Let this circuit be C. Recall that we defined the depth of a modular circuit of output type q in such a way that it includes its outgoing wires. Thus, the outgoing wires of the C_j are already accounted for in their depth, and adding one more output gate on top does not increase the depth of the resulting circuit. Hence, C also has depth 2. ◀

5.2 Nested block-symmetric construction

Now we present the construction that achieves the upper bound in Theorem 2. It simply applies Theorem 20 to recursively compute the AND over each block defined by the tree $\mathcal{T}_n^{\boldsymbol{k}}$.

▶ **Theorem 24.** *Let $m \in \mathbb{N}$ be a number with $r \geq 2$ distinct prime factors. Fix an $h \in \mathbb{N}$ and an h-tuple $\boldsymbol{k} = (k_1(n), \dots, k_h(n))$ such that $\prod_{i \in [h]} k_i(n) = n$ for all $n \in \mathbb{N}$. Let $k_{max}(n) \coloneqq \max_{i \in [h]} k_i(n)$. For every $n \in \mathbb{N}$ there is an $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric MOD_m-circuit C_n of size $2^{\mathcal{O}(k_{max}(n)^{1/r} \cdot \log k_{max}(n))}$ and depth $2h$ that computes AND_n.*

Proof. The inductive circuit construction follows the structure of $\mathcal{T}_n^{\boldsymbol{k}}$. Recall that the tree $\mathcal{T}_n^{\boldsymbol{k}}$ defines a set $\mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$ of blocks of siblings in the tree. The AND over each such block can be computed via a circuit from Theorem 20. Below is a schematic visualisation of the top two levels of $\mathcal{T}_n^{\boldsymbol{k}}$ where $k_i(n) = n^{1/h}$ for each $i \in [h]$. Each blue cone in this picture corresponds to a block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$. On the level of leaves, such a block is a subset of the input variables of size $n^{1/h}$. A block B on a higher level bundles $n^{1/h}$ blocks from the level below. In this example, our circuit will contain one instance of the $\mathrm{AND}_{n^{1/h}}$-circuit from Theorem 20 for each $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$. The output of such a circuit will be the AND over $L_0(B)$, that is, the set of all input variables that sit below the block B.



Formally, we construct our AND_n-circuit by induction from level 0 to h of $\mathcal{T}_n^{\boldsymbol{k}}$: On level 0, every block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$ is a subset of leaves. For each such B, we invoke Theorem 20 to obtain a $\mathbf{Sym}(B)$-symmetric circuit C_B that computes the AND over all variables x_i with $i \in B$.

Next, we consider an arbitrary level $i > 0$ and assume by induction that for all blocks $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$ with $B \subseteq L_{i-1}$, a circuit C_B with the following properties has been constructed:

1. C_B computes the AND over all variables x_i such that $i \in L_0(B)$.
2. C_B is symmetric under the subgroup of $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ that stabilises B setwise.

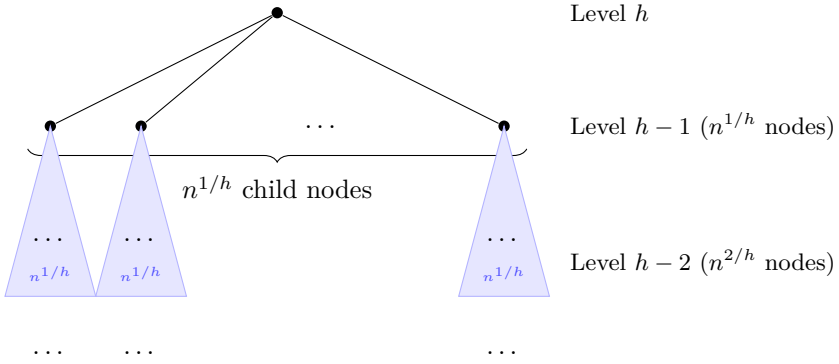


Figure 1 An $n^{1/h}$-ary tree of depth h.

Now let $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$ be a block with $B \subseteq L_i$. To obtain the circuit C_B for this block, we invoke Theorem 20 on the outputs of the circuits $C_{B'}$ for all blocks $B' = B(v)$ for every node v that is a child of some node in B. That is, C_B simply computes the AND over the results of all the blocks on level $i-1$ that are bundled in B. It is not difficult to check that this circuit C_B again satisfies the two properties above (using the fact that the construction from Theorem 20 is symmetric). For the unique block B on level $h-1$, the circuit C_B is the desired $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric circuit that computes the AND over all n input variables. The total depth of the construction is $2h$ because the circuit from Theorem 20 has depth 2, and we use this on h levels. For each block, the subcircuit that Theorem 20 gives us has size at most $2^{\mathcal{O}(k_{\max}(n)^{1/r} \cdot \log k_{\max}(n))}$. The number of blocks is $|\mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})| \leq n$, so the total size of the constructed circuit is at most $n \cdot 2^{\mathcal{O}(k_{\max}(n)^{1/r} \cdot \log k_{\max}(n))} = 2^{\mathcal{O}(k_{\max}(n)^{1/r} \cdot \log k_{\max}(n) + \log n)}$. The additive $\log n$ term vanishes in the $\mathcal{O}$ because $k_{\max}(n) \geq n^{1/h}$. ◀

6 Concluding remarks

Using a clean group-theoretic framework, we have determined the exact size complexity of AND_n for fully symmetric and nested block-symmetric CC^0-circuits. For fully symmetric circuits, it turns out that the depth-2 construction from [24] is already optimal. For nested block-symmetric circuits, the optimal size is achieved by recursively nesting that construction. This approach is of course somewhat naive, and we know from [24, Proposition 4.3] that one can in fact compress its depth down to $h+1$. This is done via a trick that lets the authors chain consecutive $\mathbb{Z}_{pq}$-expressions together without explicitly having to compute the AND over each block. Strangely, the implementation of this trick in [24, Proposition 4.3] achieves only $1/(r-1)$ in the exponent of the circuit size, rather than $(1/r)$, which we have shown to be optimal for symmetric circuits. After a thorough examination of the depth reduction trick, it seems that this increase in size is perhaps inherent and cannot be avoided if one wishes to achieve a lower depth than $2h$. Thus, our results motivate further efforts to improve the size of the depth-$(h+1)$ construction, or to show that this is impossible (at least under symmetry assumptions).

Beyond that, we hope that our techniques will provide a basis for proving lower bounds under even less restrictive symmetry assumptions in the future, so that one day, the 30 year old problem CC^0 versus ACC^0 may be settled.

References

1 Matthew Anderson and Anuj Dawar. On Symmetric Circuits and Fixed-Point Logics. *Theory of Computing Systems*, 60(3):521–551, 2017. doi:10.1007/s00224-016-9692-2.

2 David A. Mix Barrington, Richard Beigel, and Steven Rudich. Representing Boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4:367–382, 1994. doi:10.1007/BF01263424.

3 David A. Mix Barrington, Howard Straubing, and Denis Thérien. Non-uniform automata over groups. *Information and Computation*, 89(2):109–132, 1990. doi:10.1016/0890-5401(90)90007-5.

4 Andreas Blass, Yuri Gurevich, and Saharon Shelah. Choiceless polynomial time. *Annals of Pure and Applied Logic*, 100(1-3):141–187, 1999. doi:10.1016/S0168-0072(99)00005-6.

5 Joshua Brakensiek, Sivakanth Gopi, and Venkatesan Guruswami. Constraint Satisfaction Problems with Global Modular Constraints: Algorithms and Hardness via Polynomial Representations. *SIAM Journal on Computing*, 51(3):577–626, 2022. doi:10.1137/19M1291054.

6 Brynmor Chapman and R. Ryan Williams. Smaller ACC^0 circuits for symmetric functions. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215 of *LIPIcs*, pages 38:1–38:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.38.

7 Arkadev Chattopadhyay, Navin Goyal, Pavel Pudlak, and Denis Therien. Lower bounds for circuits with MOD_m gates. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 709–718, 2006. doi:10.1109/FOCS.2006.46.

8 Anuj Dawar, Benedikt Pago, and Tim Seppelt. Symmetric Algebraic Circuits and Homomorphism Polynomials, 2025. doi:10.48550/arXiv.2502.06740.

9 Anuj Dawar, Benedikt Pago, and Tim Seppelt. Symmetric Algebraic Circuits and Homomorphism Polynomials. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, 2026. doi:10.4230/LIPIcs.ITCS.2026.46.

10 Anuj Dawar and Gregory Wilsenach. Symmetric Arithmetic Circuits. In *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, volume 168 of *LIPIcs*, pages 36:1–36:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.36.

11 Anuj Dawar and Gregory Wilsenach. Lower Bounds for Symmetric Circuits for the Determinant. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 52:1–52:22, 2022. doi:10.4230/LIPIcs.ITCS.2022.52.

12 Anuj Dawar and Gregory Wilsenach. Symmetric Arithmetic Circuits, 2024. arXiv:2002.06451.

13 Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching Vector Codes. *SIAM Journal on Computing*, 40(4):1154–1178, 2011. doi:10.1137/100804322.

14 Zeev Dvir and Sivakanth Gopi. 2-Server PIR with Sub-Polynomial Communication. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 577–584, 2015. doi:10.1145/2746539.2746546.

15 Prateek Dwivedi, Benedikt Pago, and Tim Seppelt. Lower Bounds in Algebraic Complexity via Symmetry and Homomorphism Polynomials, 2026. To appear at STOC 2026. doi:10.48550/arXiv.2601.09343.

16 Klim Efremenko. 3-Query Locally Decodable Codes of Subexponential Length. *SIAM Journal on Computing*, 41(6):1694–1703, 2012. doi:10.1137/090772721.

17 Parikshit Gopalan. Constructing Ramsey graphs from Boolean function representations. *Comb.*, 34(2):173–206, 2014. doi:10.1007/S00493-014-2367-1.

18 Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20(1):71–86, 2000. doi:10.1007/s004930070032.

19 Vince Grolmusz. A degree-decreasing lemma for (MOD_p-MOD_m) circuits. *Discrete Mathematics and Theoretical Computer Science*, 4(2):247–254, 2001. doi:10.46298/dmtcs.289.

20 Vince Grolmusz and Gábor Tardos. Lower bounds for (MOD_p-MOD_m) circuits. *SIAM Journal on Computing*, 29(4):1209–1222, 2000. doi:10.1137/S0097539798340850.

21 Johan Håstad. *Computational limitations for small depth circuits*. PhD thesis, Massachusetts Institute of Technology, 1986.

22 William He and Benjamin Rossman. Symmetric formulas for products of permutations. In *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 68:1–68:23, 2023. doi:10.4230/LIPIcs.ITCS.2023.68.

23 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Intermediate Problems in Modular Circuits Satisfiability. In *Proceedings of LICS'20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 578–590, 2020. doi:10.1145/3373718.3394780.

24 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Complexity of modular circuits. In *Proceedings of LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 32:1–32:11. ACM, 2022. doi:10.1145/3531130.3533350.

25 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Nonuniform Deterministic Finite Automata over Finite Algebraic Structures. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 161:1–161:14, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2025.161.

26 Paweł M. Idziak, Piotr Kawałek, Jacek Krzaczkowski, and Armin Weiß. Satisfiability Problems for Finite Groups. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *LIPIcs*, pages 127:1–127:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.127.

27 Piotr Kawałek and Armin Weiß. Violating Constant Degree Hypothesis Requires Breaking Symmetry. In *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, volume 327 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:21, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2025.58.

28 Alexandre Laugier and Manjil Saikia. Periodic sequences modulo m, 2015. arXiv:1209.2371.

29 Benedikt Pago. Optimal Lower Bounds for Symmetric Modular Circuits, 2026. doi:10.48550/arXiv.2604.04760.

30 Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 77–82, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28404.

31 Howard Straubing and Denis Thérien. A Note on MOD_p-MOD_m Circuits. *Theory of Computing Systems*, 39(5):699–706, 2006. doi:10.1007/s00224-004-1210-2.

A Details on symmetry groups and circuits

We first give a detailed proof of the claim that symmetric MOD_m-circuits can always be assumed to be rigid.

▶ **Lemma 25** (Rigidification). *Let $\Gamma \leq \mathbf{Sym}_n$. If C is a Γ-symmetric circuit MOD_m-circuit, then there exists a rigid Γ-symmetric MOD_m-circuit C' that computes the same function as C and satisfies $|C'| \leq |C|$.*

Proof. We construct C' by merging equivalent gates in C. It may be necessary to repeat the following procedure more than once to accomplish rigidity. Formally, we define C' and a surjective map $\delta\colon V(C) \to V(C')$ inductively from the input gates of C to the root. The map δ keeps track of which gates of C have been merged into which gates of C' and just

simplifies the presentation. Let $\mathbf{Stab}_{\text{in}} \leq \mathbf{Sym}(V(C))$ denote the group consisting of all circuit automorphisms of C that fix every input gate of C. As long as C is not rigid, there is at least one $\mathbf{Stab}_{\text{in}}$-orbit of gates that is not a singleton set.

By our convention, for every variable, there is a unique input gate with that label in C, so input gates never violate rigidity. Thus, we let the input gates of C' be the same as in C, and define δ as the identity map on them. Now assume by induction that we have constructed C' and δ up to layer d. We describe the construction on layer $d+1$. Let $V_{d+1} \subseteq V(C)$ be the set of gates in layer $d+1$. For each $\mathbf{Stab}_{\text{in}}$-orbit $O \subseteq V_{d+1}$, we introduce a new gate g_O in layer $d+1$ of C', and we let $\delta(g) \coloneqq g_O$ for every $g \in O$. The operation type of g_O is the same as that of each gate in O.

To define the connections between layer $d+1$ and layer d in C', we first note:

▷ Claim 26. Let $g, g' \in V(C)$ be in the same $\mathbf{Stab}_{\text{in}}$-orbit. Then there is a bijection $\gamma\colon gE(C) \to g'E(C)$ such that for each $h \in gE(C)$, h and $\gamma(h)$ are in the same $\mathbf{Stab}_{\text{in}}$-orbit.

Proof. Since g, g' are in the same orbit, there exists $\pi \in \mathbf{Stab}_{\text{in}}$ such that $\pi(g) = g'$, and the action of π on $gE(C)$ defines a bijection $\gamma\colon gE(C) \to g'E(C)$ with the claimed property. ◁

By the claim, for any two gates $g, g' \in O$, it holds that $\{\delta(h) \mid h \in gE(C)\} = \{\delta(h) \mid h \in g'E(C)\}$. Therefore we can pick an arbitrary $g \in O$ and define the set of children of g_O in C' as

$$g_O E(C') \coloneqq \{\delta(h) \mid h \in gE(C)\}.$$

For each child $\delta(h)$ of g_O in C', we let the multiplicity of the edge between g_O and $\delta(h)$ be defined as follows. Let $m(g,h)$ denote the multiplicity of the edge between g and h in C. Then in C', the multiplicity of the edge $(g_O, \delta(h))$ is

$$\sum_{h' \in \delta^{-1}(\delta(h)) \cap gE} m(g, h').$$

This finishes the construction of C'. Note that by construction, two gates $g_1, g_2 \in V(C)$ are in the same $\mathbf{Stab}_{\text{in}}$-orbit if and only if $\delta(g_1) = \delta(g_2)$. Clearly, $|C'| \leq |C|$.

▷ Claim 27. C' computes the same function as C.

Proof. We show by induction that for every gate $g \in V(C)$, $\delta(g)$ computes the same function as g. For the input gates this is clear. Now consider the inductive step for layer $d+1$. Let $g \in V(C)$ be a gate on layer $d+1$, labelled with the operation MOD_m^R, and let $h_1, \ldots, h_k$ be its children in C. Then it computes

$$g(\bar{x}) = \begin{cases} 1 & \text{if } (\sum_{i \in [k]} m(g,h_i) \cdot h_i(\bar{x}) \mod m) \in R \\ 0 & \text{otherwise} \end{cases}$$

Now the children of $\delta(g)$ in C' are defined as the δ-images of the children of some gate g^* in the same orbit of g that was used in the construction. Hence, there exists a $\pi^* \in \mathbf{Stab}_{\text{in}}$ that maps g to g^* and the children of g to the children of g^* (preserving edge multiplicities). It is generally true for every $\pi \in \mathbf{Stab}_{\text{in}}$ and any $h \in V(C)$ that $\pi(h)$ and h compute the same function. Thus, we have

$$\begin{aligned} \sum_{i \in [k]} m(g,h_i) \cdot h_i(\bar{x}) &= \sum_{i \in [k]} m(g,h_i) \cdot \pi^*(h_i)(\bar{x}) \\ &= \sum_{i \in [k]} m(g,h_i) \cdot \delta(\pi^*(h_i))(\bar{x}), \end{aligned}$$

where the last equality holds by induction hypothesis. In the construction of C', the edge multiplicities between $\delta(g^*) = \delta(g)$ and its children $\{\delta(\pi^*(h_i)) \mid i \in [k]\}$ are chosen such that $\delta(g)$ indeed computes the above sum modulo m, and checks for membership in R. This finishes the inductive step. ◁

▷ Claim 28. Every $\pi \in \Gamma$ extends to a circuit automorphism of C', that is, C' is Γ-symmetric.

Proof. Let $\pi \in \Gamma$. Since C is Γ-symmetric, there is an automorphism σ of C that π extends to. The σ-image of every $\mathbf{Stab}_{\text{in}}$-orbit $O \subseteq V(C)$ is again a $\mathbf{Stab}_{\text{in}}$-orbit. Thus, we can define a bijection $\sigma' \colon V(C') \to V(C')$ by setting $\sigma'(g_O) \coloneqq g_{\sigma(O)}$ for every $\mathbf{Stab}_{\text{in}}$-orbit $O \subseteq V(C)$. By construction of C' and because σ is an automorphism of C, σ' is an automorphism of C'. ◁

The construction is iterated until the resulting circuit C' is rigid, which has to happen at some point, because as long as there is a non-singleton $\mathbf{Stab}_{\text{in}}$-orbit, the construction strictly reduces the number of gates. ◀

▶ **Lemma 4.** *Let C be a Γ-symmetric rigid circuit, for $\Gamma \leq \mathbf{Sym}_n$. Let $\delta \colon \{x_1, \dots, x_n\} \to \{0,1\}$ be an assignment to the variables, let $g \in V(C)$ be a gate, and let $\pi \in \Gamma$. Then*

$$g(\delta(x_1), \dots, \delta(x_n)) = \pi(g)(\delta(\pi^{-1}(x_1)), \dots, \delta(\pi^{-1}(x_n))).$$

Proof. By induction on the circuit structure. Let g be an input gate labelled with a variable x_i. Let $j \coloneqq \pi(i)$. Then $g' \coloneqq \pi(g)$ is an input gate labelled with x_j. It holds $g(\delta(x_1), \dots, \delta(x_n)) = \delta(x_i)$ and $g'(\delta(x_1), \dots, \delta(x_n)) = \delta(x_j)$. In other words, $g(x_1, \dots, x_n)$ is the i-th projection, and $g'(x_1, \dots, x_n)$ is the j-th projection. So, as desired, we have

$$g(\delta(x_1), \dots, \delta(x_n)) = \delta(x_i) = g'(\delta(\pi^{-1}(x_1)), \dots, \delta(\pi^{-1}(x_n))).$$

For the inductive step, let g be an internal gate of the circuit and assume that the statement holds for all children of g. Let $h_1, \dots, h_k$ denote the children of g. Let f denote the operation computed by g, which is the same as the operation of $\pi(g)$. The proof works for every fully symmetric operation f, in particular for $f = \mathrm{MOD}_m^R$. Then

$$\begin{aligned} g(\delta(x_1), \dots, \delta(x_n)) &= f(h_1(\delta(x_1), \dots, \delta(x_n)), \dots, h_k(\delta(x_1), \dots, \delta(x_n))) \\ &= f(\pi(h_1)(\delta(\pi^{-1}(x_1)), \dots, \delta(\pi^{-1}(x_n))), \dots, \pi(h_k)(\delta(\pi^{-1}(x_1)), \dots, \\ &\quad \delta(\pi^{-1}(x_n)))) = \pi(g)(\delta(\pi^{-1}(x_1)), \dots, \delta(\pi^{-1}(x_n))). \end{aligned}$$

The second equality is the induction hypothesis for $h_1, \dots, h_k$. The third equality is true because the gate $\pi(g)$ computes f applied to the outputs of the gates $\pi(h_1), \dots, \pi(h_k)$, and the order of the arguments of f is irrelevant by symmetry. ◀

▶ **Lemma 6.**

1. *Let $n > 8$ and C be a $\mathbf{Sym}_n$-symmetric rigid circuit. Let k be such that $1 \leq k \leq \frac{n}{4}$ and $\mathrm{maxOrb}_{\mathbf{Sym}_n}(C) \leq \binom{n}{k}$. Then for every $g \in V(C)$, $\mathbf{Stab}_{\mathbf{Sym}_n}(g)$ has a support of size less than k.*
2. *Let C be an $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric rigid circuit, for some h-tuple $\boldsymbol{k}$ such that $k_{min} \coloneqq \min_{i \in [h]} k_i(n) > 8$. For every block $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$, let k_B be such that $1 \leq k_B \leq \frac{|B|}{4}$ and $\mathrm{maxOrb}_{\mathbf{Stab}(B)}(C) \leq \binom{|B|}{k_B}$. Then for every $g \in V(C)$, $\mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})}(g)$ has a B-support $S \subseteq B$ with $|S| < k_B$.*

Proof. The first result is stated in [12, Theorem 14] for Boolean circuits symmetric under the action of $\mathbf{Sym}_n$ on variables $\mathcal{X} \coloneqq \{x_{ij} \mid i, j \in [n]\}$. The symmetric circuits we consider here can be viewed as circuits in the variables $\{x_{ii} \mid i \in [n]\} \subseteq \mathcal{X}$, and thus, [12, Theorem 14] also applies here (the particular operation type of the gates is irrelevant for this).

With some more work, (2) follows from (1). Fix a gate $g \in V(C)$. We can assume that g is an internal gate, as for input gates, there always exists a B-support of size 0 or 1, depending whether the index of the variable that g is labelled with is in B or not. Let $B \in \mathcal{B}(\mathcal{T}_n^{\boldsymbol{k}})$. Define a new circuit C_B from C as follows. Remove all input variables x_i such that $i \notin L_0(B)$. Then, for every $v \in B$, identify all input gates labelled with variables x_i, for every $i \in L_0(v)$. This leaves us with a circuit with one input variable for every $v \in B$. By Lemma 25, we may again assume that it is rigid. This is the circuit C_B. The original circuit C is in particular symmetric under $\mathbf{Stab}(B)$, the setwise stabiliser of B in $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$; hence C_B is also $\mathbf{Stab}(B)$-symmetric. The action of $\mathbf{Stab}(B)$ on B is that of $\mathbf{Sym}(B)$, so C_B can really be seen as a $\mathbf{Sym}(B)$-symmetric circuit. By the assumption on orbit size in (2), $\operatorname{maxOrb}_{\mathbf{Sym}(B)}(C_B) \leq \binom{|B|}{k_B}$, and we have assumed that $1 \leq k_B \leq \frac{|B|}{4}$. Moreover, we are assuming that $|B| > 8$. Hence, (1) can be applied to C_B, where we just rename B to $[n]$. This means that in C_B, the gate g has a support $S \subseteq B$ of size at most k_B. We now show that this is also a B-support of g in C. We have to show that

$$\mathbf{Stab}^{\bullet}_{\mathbf{Sym}(B)}(S) \leq \mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})}(g)|_B.$$

So let $\pi \in \mathbf{Stab}^{\bullet}_{\mathbf{Sym}(B)}(S)$ and choose an arbitrary $\sigma \in \mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$ that fixes B setwise and permutes the elements of B according to π. Since C is $\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})$-symmetric, σ extends to a circuit automorphism θ_C of C. This also induces a circuit automorphism $\theta_{C_B} \in \mathbf{Sym}(V(C_B))$ of C_B, which behaves like θ_C on the internal gates (noting that except for the input gates and the connections to them, C and C_B are identical circuits). Now by definition of support, θ_{C_B} fixes g: Indeed, θ_{C_B} acts on the inputs of C_B as π, and π fixes the support S of g pointwise. But since θ_{C_B} and θ_C agree on g, also $\theta_C(g) = g$. Hence, $\sigma \in \mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})}(g)$, and its restriction to B is π, so $\pi \in \mathbf{Stab}_{\mathbf{Aut}(\mathcal{T}_n^{\boldsymbol{k}})}(g)|_B$, which is what we had to show. ◀