Complexity of Modular Circuits

Paweł M. Idziak
Department of Theoretical Computer Science, Jagiellonian University
Kraków, Poland
pawel.idziak@uj.edu.pl

Piotr Kawałek
Department of Theoretical Computer Science, Jagiellonian University
Kraków, Poland
piotr.kawalek@doctoral.uj.edu.pl

Jacek Krzaczkowski
Department of Computer Science, Maria Curie-Skłodowska University
Lublin, Poland
krzacz@poczta.umcs.lublin.pl

ABSTRACT

We study how the complexity of modular circuits computing AND depends on the depth of the circuits and the prime factorization of the modulus they use. In particular our construction of subexponential circuits of depth 2 for AND helps us to classify (modulo Exponential Time Hypothesis) modular circuits with respect to the complexity of their satisfiability. We also study a precise correlation between this complexity and the sizes of modular circuits realizing AND. In particular we use the superlinear lower bound from [10] to check satisfiability of CC^0 circuits in probabilistic $2^{O(n/\varepsilon(n))}$ time, where ε is some extremely slowly increasing function. Moreover we show that AND can be computed by a polynomial size modular circuit of depth 2 (with $O(\log n)$ random bits) providing a probabilistic computational model that can not be derandomized.

We apply our methods to determine (modulo ETH) the complexity of solving equations over groups of symmetries of regular polygons with an odd number of sides. These groups form a paradigm for some of the remaining cases in characterizing finite groups with respect to the complexity of their equation solving.

CCS CONCEPTS

• **Theory of computation → Circuit complexity**; **Complexity theory and logic**; **Problems, reductions and completeness**; **Complexity classes**; • **Mathematics of computing** → *Combinatorial algorithms.*

KEYWORDS

modular circuits, circuit complexity, circuit satisfiability

ACM Reference Format:
Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. 2022. Complexity of Modular Circuits. In *37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '22), August 2–5, 2022, Haifa, Israel.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3531130.3533350

ACKNOWLEDGMENTS

This research is partially supported by Polish NCN Grant # 2014/14/A/ST6/00138

LICS '22, August 2–5, 2022, Haifa, Israel

ACM ISBN 978-1-4503-9351-5/22/08…$15.00
https://doi.org/10.1145/3531130.3533350

1 INTRODUCTION

Due to the pioneering work of Cook satisfiability of Boolean circuits is among the most celebrated problems in computer science. Although the problem itself is NP-complete, it becomes solvable in PTIME when restricted to circuits of special kinds, like monotone circuits or circuits with linear gates only. Here by linear gate we mean XOR of unbounded fan-in. Such a gate simply checks the parity of the sum of inputs. This has been generalized to the gates MOD_m^A that check if the sum of inputs, taken modulo m belongs to the set $A \subseteq \{0, \dots, m-1\}$. Note here that, traditionally only the sets $A = \{0\}$ (or dually, only $A = \{1, \dots, m-1\}$) are allowed. We will however always consider generalized modular gates (like e.g. in [7, 8, 10, 26]), i.e. MOD_m^A with arbitrary A and multiple wires between gates (including input gates). These gates are to be used to build modular circuits of bounded depth. More precisely for depth h and modulus m by $\mathrm{CC}_h[m]$ we mean a class of circuits built of gates MOD_m^A, possibly with different A for each gate. In Section 4 we also discuss modular circuits with possibly different moduli on different levels. Thus a $\mathrm{CC}[m_1; \dots; m_h]$-circuit admits only gates of the type $\mathrm{MOD}_{m_i}^A$ on the i-th level.

Our results start with the following full characterization of parameters h and m for which satisfiability of $\mathrm{CC}_h[m]$-circuits ($\mathrm{CC}_h[m]$-SAT for short) is in PTIME. In what follows, for a positive integer m by $\omega(m)$ we denote the number of different prime factors of m.

Theorem 1.1. *Let h and m be positive integers. Then under the assumption of ETH the problem of satisfiability for $\mathrm{CC}_h[m]$-circuits is in PTIME iff $h = 1$ or $\omega(m) = 1$.*

Our nonpolynomial (but subexponential) lower bounds are based on the construction of relatively small $\mathrm{CC}_2[m]$-circuits computing AND_n, i.e. of size $2^{O(n^{1/\omega}\log n)}$, where $\omega = \omega(m) \geqslant 2$, and the circuit size is defined as the number of its gates. Our construction of two levels circuits employs the idea of relatively short symmetric polynomials over finite fields used by Barrington, Beigel and Rudich in [1] to build 3-level circuit of a similar size, but with A always set to $\{0\}$.

In fact our construction is flexible enough for building not only small $\mathrm{CC}_2[m]$-, i.e. $\mathrm{CC}[m; m]$-circuits, but also for small $\mathrm{CC}\left[p^k; m\right]$-circuits, where $\omega(m) \geqslant 2$ but the prime p does not divide m. On the other hand from the papers [2, 26] we know that $\mathrm{CC}\left[m; p^k\right]$-circuits expressing AND need at least $2^{\Omega(n)}$ gates. But the general case of $\mathrm{CC}[m_1; m_2]$-circuits has been completely open. To squeeze 3 levels into 2 we heavily use our algebraic results from [18] which provide a sufficient control of internal value of MOD_m^A gate (i.e. the value of the sum modulo m before deciding if it belongs to the set A). This allows us to work directly on $\mathrm{CC}_h[m]$-circuits to provide

relatively simple and natural constructions. Moreover, this may be important in further applications of our methods.

It is also worth to notice here that the expressive power of modular circuits with 2 levels is also very sensitive to the sets A used in MOD_m^A. Indeed in [6] Caussinus shows the very same lower bound $2^{\Omega(n)}$ for AND if on the second level only the set $A = \{1, \dots, m-1\}$ is allowed.

Next we show that not only the width of the modulus m, i.e. $\omega(m)$, but also the circuit depth may substantially contribute to reduce the size of $\mathrm{CC}_h[m]$-circuits realizing AND. Surprisingly also the number $\omega'(m)$ of large prime factors of m plays some role. By a large prime divisor of m we mean each one that is at least $\omega(m)$.

Theorem 1.2. *For $h \geqslant 3$ and a positive integer m with $\omega = \omega(m) \geqslant 2$ and $\omega' = \omega'(m)$ there are $\mathrm{CC}_h[m]$-circuits of size $2^{O(n^{1/((\omega-1)(h-2)+\omega')}\log n)}$, computing n-ary* AND.

Note here that if the modulus m satisfies $\omega(m) = \omega'(m)$ then Theorem 1.2 gives circuits of size $2^{O(n^{1/((\omega-1)(h-1))}\log n)}$ that compute n-ary AND. Since with $\omega(m) \to \infty$ we have $\frac{\omega'(m)}{\omega(m)} \to 1$, the asymptotical upper bounds for the circuits realizing AND is also $2^{O(n^{1/((\omega-1)(h-1))}\log n)}$. Very recently Chapman and Williams [9] have built $\mathrm{CC}_h[m]$-circuits for all Boolean functions that are symmetric. The asymptotic behaviour (with respect both to h and $\omega(m)$) of sizes of their circuit is given by $2^{O(n^{c/((\omega-1)(h-c))})}$, but a constant c that is involved is rather large. Note that every n-ary symmetric function is a sum of several (at most $n+1$) functions $\mathrm{EXACT}_k(x_1, \dots, x_n)$ that returns 1 iff exactly k among the x_i's are set to 1. In fact our proof of Proposition 3.1 can be easily adapted to show that each of the above functions $\mathrm{EXACT}_k(x_1, \dots, x_n)$ is computable by $\mathrm{CC}_2[m]$-circuits of size $2^{O(n^{1/\omega}\log n)}$. Thus, by summing up (at most $n+1$ summands), we can represent each n-ary symmetric function by $\mathrm{CC}_3[m]$-circuits of size $2^{O(n^{1/\omega}\log n)}$ which is slightly better than the mentioned bound of Chapman and Williams. We do hope that our methods can be also employed to improve their bounds for $\mathrm{CC}_h[m]$-circuits of bigger depth h.

Although the only known lower bound for the size of modular circuits computing AND is slightly better than linear (see [10]), Barrington, Straubing and Thérien [2] conjectured that it has to be exponential. In fact, after the paper [1], the bound $2^{\Omega(n^\delta)}$ with some $\delta > 0$ is a popular belief. In contrast to this conjecture there are constructions [13, 14] of (quasi)polynomial size probabilistic modular circuits computing AND. The construction in [13] is of quasipolynomial size and uses polylog(n) random bits. The one from [14] fixes the depth to be constant (but a substantial one), reduces the size to be polynomial and cuts down the number of random bits to $O(\log n)$. Our techniques applied in the proof of Theorem 1.2 have proved to be useful also in this probabilistic setting. First, while keeping only $O(\log n)$ random bits, we reduce the depth of the circuits realizing AND to be only 2. Note here that our construction is slightly more transparent but at the expense of uniformity present in [14]. We believe however that using expanders and universal hashing functions (again as in [14]) one might be able to uniformize the circuits.

Theorem 1.3. *For the modulus m with $\omega(m) \geqslant 2$ the n-ary* AND functions can be realized by $\mathrm{CC}_2[m]$-circuits of polynomial size with $O(\log n)$ random bits. In fact the realization is done by $\mathrm{CC}[p;q]$-circuits, where p, q are different primes.

Again, the mentioned lower bound $2^{\Omega(n)}$ for the size of deterministic $\mathrm{CC}[p;q]$-circuits computing AND, blocks any derandomization here. Thus to confirm the suggestion made in [14] that if AND can be computed by small CC^0 circuits one would need to increase the depth.

We also show how superpolynomial lower bounds for sizes of modular circuits computing AND would give rise to subexponential algorithms checking satisfiability (or equivalence) of such circuits. In fact this connection, as well as the converse one, (due to their technicality) is presented only in Section 6, in particular in Theorem 6.1. As a result of these considerations we obtain (see Theorem 6.4) an upper bound for satisfiability of CC^0-circuits that is asymptotically lower than the one ETH permits for AC^0-circuits.

Our methods proved themselves to be powerful enough to be applied in some other contexts. In particular in Theorem 7.1 we give a characterization of dihedral groups $\mathbf{D}_{2k+1}$, i.e. groups of symmetries of regular polygon with odd number of sides, for which the problem of solving equation is tractable. In fact for odd m we show that this happens only if $\omega(m) = 1$, or ETH fails.

The complexity of solving equations over nilpotent groups has been shown in [12] to be tractable. On the other hand nonsolvable groups have been shown there to be NP-complete with this respect. The problem for the remaining, but very broad, realm of solvable but not nilpotent groups is open for over 20 years. It was conjectured that solving equations for these groups is tractable (see e.g. [15]). This conjecture was supported by a number of examples (e.g. [11, 16, 17]). All those examples have a nilpotent normal subgroup modulo which the group is nilpotent as well. Actually, under ETH, our paper [20] shows that this is a necessary condition for tractability. On the other hand the examples of dihedral groups provided in Theorem 7.1 show that this natural condition is not sufficient.

Another, in fact pretty similar, application of our methods is done for satisfiability of multivalued circuits (CSat), as we defined in [22]. In Section 7 we show examples of 2-nilpotent algebras for which this problem is not in P (unless ETH fails). Again, these are the first examples of 2-nilpotent algebras for which CSat is not in P.

2 SHALLOW OR NARROW MAY APPLY

In this section we analyze the expressive power of $\mathrm{CC}_h[m]$-circuits with $h = 1$ or $\omega(m) = 1$, which are to be called shallow and narrow, respectively. We start with stating that in such realm $\mathrm{CC}_h[m]$-circuits can compute AND only of bounded arity. Although one can find proofs of very similar statements in the literature (e.g [2, 3]), we have decided to sketch the proof in Section 8.

Proposition 2.1. *For positive integers m, h, k and a prime p, the arity of* AND computable by

- $\mathrm{CC}_1[m]$-circuits is bounded by $m-1$,
- $\mathrm{CC}_h\left[p^k\right]$-circuits is bounded by a constant depending only on h and $m = p^k$. □

From the above bound we can infer one implication in Theorem 1.1. To do this an easy observation is required, that we can simulate $\mathrm{CC}_h[m]$ circuit with some inputs fixed to be constant, just by slightly modifying the structure of the circuit, without inflating its size. For this reason, in the following, we simply allow some inputs to be constant

Corollary 2.2. *Satisfiability of $\mathrm{CC}_h[m]$-circuits is in PTIME whenever $h = 1$ or $\omega(m) = 1$.*

Proof. The only property of $\mathrm{CC}_1[m]$ and $\mathrm{CC}_h\left[p^k\right]$ circuits we are going to use is that they have a bound, say c, for the arity of AND they can express. This bound allows us to reduce our search for an n-tuple a in the large set $\{0,1\}^n$ satisfying the circuit Γ to a smaller subset of size at most n^c containing only tuples with at most c ones. Indeed, let a be a satisfying tuple with minimal possible $\left|a^{-1}(1)\right|$. By this minimality we know that Γ with all the inputs with indices outside $a^{-1}(1)$ set to 0 behaves like the $\left|a^{-1}(1)\right|$-ary AND. Thus $\left|a^{-1}(1)\right| \leqslant c$, so that the tuple a has at most c ones, as claimed. □

The function AND_n is an example of a nonconstant extremely unbalanced boolean function, i.e, one value is taken exactly once. Our next goal is to show that shallow or narrow modular circuits can only compute functions with rather balanced piles, i.e. preimages $f^{-1}(0)$ and $f^{-1}(1)$. The smaller of these two piles is to be denoted by $D(f)$. Formally the balance of the n-ary boolean function f is defined to be

$$\mathrm{bal}(f) = 1 - \frac{\left|\left|f^{-1}(0)\right| - \left|f^{-1}(1)\right|\right|}{2^n} = \frac{|D(f)|}{2^{n-1}}.$$

Thus, the constant functions have balance 0. The functions AND_n and OR_n have the smallest possible non-zero balance 2^{1-n}. In what follows any function f with $|D(f)| = 1$, or alternatively with the smallest possible non-zero balance is to be called a spike.

Remark 2.3. *Each nonconstant n-ary boolean function f can be turned to be $(n-k)$-ary spike by fixing $k \leqslant \log|D(f)|$ of its variables to be constant from $\{0,1\}$.*

Proof. To fix the notation we use the symbol $f[x_i/c]$ for the function obtained from f by fixing the variable x_i to be $c \in \{0,1\}$.

Now, as long as $|D(f)| \geqslant 2$ we iteratively reduce the size of $D(f)$ by at least a half, by fixing the value of one of the variables without making f constant. To do that we start with picking $a^0, a^1 \in D(f) = f^{-1}(b)$ and a coordinate i so that $a^0_i = 0$ and $a^1_i = 1$. Obviously $\left|f^{-1}(b)\right| = \left|f[x_i/0]^{-1}(b)\right| + \left|f[x_i/1]^{-1}(b)\right|$ and we pick $c \in \{0,1\}$ so that $\left|f[x_i/c]^{-1}(b)\right| \leqslant \left|f[x_i/1-c]^{-1}(b)\right|$. Thus we have $|D(f[x_i/c])| \leqslant |D(f)|/2$, as required. To see that $f[x_i/c]$ is not constant, note that a^c, with its i-th coordinate removed, belongs to $D(f[x_i/c])$ so that $1 \leqslant |D(f[x_i/c])| \leqslant |D(f)|/2 < 2^n/2 = 2^{n-1}$. □

We note here that all spikes (of the same arity) are interdefinable. Indeed, if $\delta^{\varepsilon}_{\overline{a}}$ denotes the spike which takes the value $\varepsilon \in \{0,1\}$ only on the tuple $\overline{a} \in \{0,1\}^n$ then

- $\delta^{1-\varepsilon}_{\overline{a}}(\overline{x}) = 1 - \delta^{\varepsilon}_{\overline{a}}(\overline{x})$,
- $\delta^{\varepsilon}_{\overline{b}}(x_1,\ldots,x_n) = \delta^{\varepsilon}_{\overline{a}}(x_1 + a_1 - b_1,\ldots,x_n + a_n - b_n)$.

This interdefinability can be realized by modifying only the sets A in the gates MOD^A_m on the last and/or first level, so that the sizes of the corresponding circuits remain unchanged.

Now, if the arity of spikes computable by $\mathrm{CC}_h[m]$-circuits is bounded, like in Proposition 2.1, we use Remark 2.3 to turn the circuit into the one computing a spike with arity at least $n - \log|D(f)| = n - (n-1)\log\mathrm{bal}(f)$. This gives the following lower bound on the balance of $\mathrm{CC}_h[m]$-circuits independently of its arity and size.

Corollary 2.4. *For $h = 1$ or $\omega(m) = 1$ the balance of non-constant functions computable by $\mathrm{CC}_h[m]$-circuits is at least 2^{1-c}, where c bounds the arity of $\mathrm{CC}_h[m]$-computable conjunctions.*

From Corollary 2.4 we immediately get the following.

Corollary 2.5. *Let $L \subseteq \{0,1\}^*$ be a language recognizable by $\mathrm{CC}_h[m]$ circuits, where $h = 1$ or $\omega(m) = 1$. Then the number of words in L of the length n is either 0 or is rather large, i.e. at least 2^{n-c}, where c bounds the arity of $\mathrm{CC}_h[m]$-computable conjunctions.*

3 DEEP OR WIDE NEED NOT APPLY

In this section we will show the converse to Corollary 2.2 but under the assumption of the Exponential Time Hypothesis. As we have already noted this is done by constructing a conjunction of subexponential size. The next Proposition formulates this fact in more details.

Proposition 3.1. *For a positive integer m with exactly ω different prime divisors we have:*

1. *for each 3-CNF-SAT formula Φ with ℓ clauses there is a $\mathrm{CC}_2[m]$ circuit of size at most $2^{O(\ell^{1/\omega}\log\ell)}$ representing Φ,*
2. *in particular unbounded fan-in* AND can be computed by $\mathrm{CC}_2[m]$ circuits of size $2^{O(n^{1/\omega}\log n)}$, where n is the number of variables (input gates).

The above bounds on the circuits size also bound the time needed to obtain them.

Combining this Proposition with Exponential Time Hypothesis (and Sparsification Lemma [23, Thm. 1]) we immediately get the following Corollary.

Corollary 3.2. *If $h \geqslant 2$ and $\omega(m) \geqslant 2$ then satisfiability for $\mathrm{CC}_h[m]$-circuits is not in PTIME, unless ETH fails.*

As we have mentioned in the Introduction our proof of Proposition 3.1 is modelled after the idea of Barrington, Beigel and Rudich [1] where the AND_n had been shown to be computable by modular circuits of the same subexponential size as described in Proposition 3.1(2) but on 3 levels. In squeezing this to 2 levels we need the concepts of $\mathbb{Z}[p,q]$-expressions and the circuits realizing them.

In the papers [18, 19] we have been studied action of the group $\mathbb{Z}_p$ on the group $\mathbb{Z}_q$ via the function $\mathrm{b} : \mathbb{Z}_p \longrightarrow \mathbb{Z}_q$ defined by $\mathrm{b}(0) = 0$ and $\mathrm{b}(x) = 1$ for all other $x \in \mathbb{Z}_p$. With the help of this action we define $\mathbb{Z}[p,q]$-expression to be the n-ary expression over the variables $\overline{x} = (x_1,\ldots,x_n)$:

$$t(\overline{x}) = \sum_{\substack{\beta\in Z_p^n \\ c\in Z_p}} \alpha_{\beta,c} \cdot \mathrm{b}\left(\sum_{i=1}^{n} \beta_i x_i + c\right),$$

where the $\alpha_{\beta,c} \in Z_q$ while $\beta = (\beta_1, \ldots, \beta_n) \in Z_p^n$ and $c \in Z_p$, the outer sum and the multiplications by the $\alpha_{\beta,c}$'s are taken modulo q, while the inner sum and the multiplications by the β_i's are taken modulo p. Obviously the $\mathbb{Z}[p,q]$-expression $t(\overline{x})$ is determined by the sequence $\left\langle \alpha_{\beta,c} : \beta \in Z_p^n, c \in Z_p \right\rangle$ of coefficients from $\mathbb{Z}_q$. This sequence may have the exponential size p^{n+1}. However only the nonzero $\alpha_{\beta,c}$'s contribute to the length of $t(\overline{x})$ and consequently to the size of a circuit that models $t(\overline{x})$. In fact, if $t(\overline{x})$ returns always boolean values on boolean inputs $\overline{x}$, $t(\overline{x})$ may be realized by a circuit, called $\Gamma(t)$, of size $1 + |L(t)|$, where

$$L(t) = \left\{(\beta, c) \in Z_q^n \times Z_q : \alpha_{\beta,c} \neq 0\right\}.$$

Indeed, the subexpression $\mathrm{b}\left(\sum_{i=1}^n \beta_i x_i + c\right)$ can be realized by a single $\mathrm{MOD}_p^{\mathbb{Z}_p - \{-c\}}$ gate (denoted $\Gamma_{\beta,c}(t)$), and then combining the outputs of all the $\Gamma_{\beta,c}(t)$ (with (β, c) ranging over $L(t)$) by the MOD_q-like gate. For this reason by the size of the circuit $\Gamma(t)$, as well as of the $\mathbb{Z}[p,q]$-expression $t(\overline{x})$, we simply mean $1 + |L(t)|$.

In our further consideration we will also use the bunch $\Theta(t) = \left\{\Gamma_{\beta,c}(t)\right\}_{(\beta,c)\in L(t)}$ of the above gates with $|L(t)|$ outputs. These outputs are going to be treated as a single bundle (without ordering, but with copying the output of each $\Gamma_{\beta,c}(t)$ the corresponding, i.e. $\alpha_{\beta,c}$, number of times) as they always will be used as inputs to other MOD-gates, so that they will be summed up first. To keep track of the modulus used to sum up this bundle we will say that the bundle is of type q and that the bunch $\Theta(t)$ is of type $[p,q]$.

The importance of the $\mathbb{Z}[p,q]$-expressions lies in the next Fact that has been originally shown in [18] as Lemma 3.1 (but with $\mathrm{b}(x)$ replaced by $\widehat{\mathrm{b}}(x) = 1 - \mathrm{b}(1-x)$).

Fact 3.3. *With two different primes p, q we can represent every n-ary function $g : Z_p^n \longrightarrow Z_q$ by a $\mathbb{Z}[p,q]$-expression of length bounded by $2^{O(n)}$.* □

The next fact is borrowed from [1], but we include its more transparent proof in Section 8.

Fact 3.4. *Let p be a prime and $k \geqslant 1$ be an integer. Then there is a polynomial $w(\overline{x}) \in \mathbb{Z}_p[\overline{x}]$ of degree at most $p^k - 1$, such that for $\overline{x} \in \{0,1\}^n$ we have*

$$w(\overline{x}) = \begin{cases} 0, & \text{if } \left|\overline{x}^{-1}(0)\right| \equiv 0 \text{ modulo } p^k, \\ 1, & \text{else.} \end{cases}$$

With the help of Facts 3.3 and 3.4 we can represent 3-CNF formulas by a relatively short $\mathbb{Z}[p,q]$-expressions in the following sense:

Lemma 3.5. *Let p, q be two different primes and $\nu \geqslant 1$ be an integer. Then for each 3-CNF-SAT formula $\Phi(\overline{x})$ with n variables $\overline{x} = (x_1, \ldots, x_n)$ and ℓ clauses there is a $\mathbb{Z}[p,q]$-expression $t^{\Phi}_{p,q}(\overline{x})$ of size at most $2^{O(q^\nu \cdot \log \ell)}$ such that for all $\overline{a} \in \{0,1\}^n$ we have*

$$t^{\Phi}_{p,q}(\overline{a}) = \begin{cases} 0, & \text{if the number of unsatisfied (by } \overline{a}) \text{ clauses} \\ & \text{in } \Phi \text{ is divisible by } q^\nu \\ 1, & \text{else.} \end{cases}$$

Proof. To fix our notation let $\Phi(\overline{x}) = \bigwedge_{i=1}^{\ell} C_i$ be a 3-CNF formula with the clauses $C_i = C_i(z_i^1, z_i^2, z_i^3)$. Fact 3.4 supplies us with an ℓ-ary polynomial $w(c_1, \ldots, c_\ell) \in GF(q)[\overline{c}]$ of degree at most $q^\nu - 1$. We want to feed up the polynomial w by substituting $C_i(z_i^1, z_i^2, z_i^3)$ for the variable c_i to get a total function $w^* : Z_p^n \longrightarrow Z_q$. In order to do that we first extend each clause C_i to be a total function $Z_p^3 \longrightarrow Z_q$ (instead of $\{0,1\}^3 \longrightarrow \{0,1\}$) by putting arbitrary values on the set $Z_p^3 - \{0,1\}^3$. Now the function

$$w^*(\overline{z}) = w\left(C_1(z_1^1, z_1^2, z_1^3), \ldots, C_\ell(z_\ell^1, z_\ell^2, z_\ell^3)\right)$$

behaves on the boolean values exactly as we need, i.e. for $\overline{a} \in \{0,1\}^n$ we have

$$w^*(\overline{a}) = \begin{cases} 0, & \text{if the number of unsatisfied (by } \overline{a}) \text{ clauses} \\ & \text{in } \Phi \text{ is divisible by } q^\nu \\ 1, & \text{else.} \end{cases}$$

All we need is to turn w^* into a relatively short $\mathbb{Z}[p,q]$-expression. Instead of applying Fact 3.3 directly to w^* we will do it for each its monomial separately. Note that the monomials of w, after our substitution of the C_i's for c_i's have the form

$$C_{i_1}(z_{i_1}^1, z_{i_1}^2, z_{i_1}^3) \cdot \ldots \cdot C_{i_s}(z_{i_s}^1, z_{i_s}^2, z_{i_s}^3) \text{ with } s < q^\nu,$$

so that there are at most $3q^\nu$ variables involved into each such "monomial". Because of that, Fact 3.3 allows us to represent each summand in w^* by a $\mathbb{Z}[p,q]$-expression of size $O(2^{cq^\nu})$. Since ℓ^{q^ν} bounds the number of monomials of degree at most $q^\nu - 1$ it also bounds the number of summands in w^*, so that we end up with the bound $O\left(2^{cq^\nu} \cdot \ell^{q^\nu}\right) \leqslant 2^{O(q^\nu \log \ell)}$ for our $\mathbb{Z}[p,q]$-expression representing w^*. □

Our Claim shows also that for p, q, ν as above we also have a relatively short $\mathbb{Z}[p,q]$-expression $t_{p,q}(x_1, \ldots, x_n)$ that behaves almost like an n-ary AND. That is, its size is at most $2^{O(q^\nu \cdot \log n)}$ and for all $\overline{a} \in \{0,1\}^n$ we have

$$t_{p,q}(\overline{a}) = \begin{cases} 0, & \text{if the number of zeros among the } a_i\text{'s} \\ & \text{is divisible by } q^\nu, \\ 1, & \text{else.} \end{cases} \tag{1}$$

We will also use the symbols $\Gamma_{p,q}, \Gamma^{\Phi}_{p,q}$ to denote the circuits $\Gamma(t_{p,q}), \Gamma(t^{\Phi}_{p,q})$ computing the $\mathbb{Z}[p,q]$-expressions $t_{p,q}$ and $t^{\Phi}_{p,q}$, respectively. Also the symbols $\Theta_{p,q}, \Theta^{\Phi}_{p,q}$ will be used to denote the bunch of initial MOD_p-gates in the circuits $\Gamma_{p,q}, \Gamma^{\Phi}_{p,q}$.

Now we are ready to prove Proposition 3.1.

Proof. To start we let $m = p_1^{\alpha_1} \cdot \ldots p_\omega^{\alpha_\omega}$ be the prime decomposition of m. Each of the groups $\mathbb{Z}_{p_j}$'s can be identified with a subgroup of $\mathbb{Z}_m$ generated by $\frac{m}{p_j}$, by simply sending z to $\frac{m}{p_j} \cdot z$. After such identification we know that the sum $\sum_{j=1}^{\omega} \frac{m}{p_j} \cdot \mathbb{Z}_{p_j}$ is in fact a direct sum, so that each element of this sum has a unique decomposition.

To construct a $\mathrm{CC}_2[m]$ circuit computing the 3-CNF formula Φ with ℓ clauses we first fix integers $\nu_1, \ldots, \nu_\omega$ to satisfy $p_j^{\nu_j - 1} \leqslant \sqrt[\omega]{\ell} < p_j^{\nu_j}$. Also, for convenience we identify the index 0 with ω so that we can refer to the indices of the primes p_j's cyclically. Now, for each $j = 1, \ldots, \omega$, Lemma 3.5 supplies us with a $\mathbb{Z}\left[p_{j-1}, p_j\right]$-expression $t^{\Phi}_j(\overline{x})$ of the length at most $O(2^{c_j \cdot \ell^{1/\omega} \cdot \log \ell})$ so that for

$\overline{a} \in \{0,1\}^n$ we have

$$t_j^{\Phi}(\overline{a}) = \begin{cases} 0, & \text{if the number of unsatisfied (by } \overline{a}) \text{ clauses} \\ & \text{in } \Phi \text{ is divisible by } p_j^{\nu_j} \\ 1, & \text{else.} \end{cases}$$

Our identification of the direct sum $\bigoplus_{j=1}^{\omega} \frac{m}{p_j} \cdot \mathbb{Z}_{p_j}$ with a subgroup of $\mathbb{Z}_m$ allows us to sum up (modulo m) all the t_j^{Φ} to get

$$T^{\Phi}(\overline{a}) = \sum_{j=1}^{\omega} \frac{m}{p_j} \cdot t_j^{\Phi}(\overline{a}). \tag{2}$$

We argue now that for $\overline{a} \in \{0,1\}^n$

$$T^{\Phi}(\overline{a}) = 0 \text{ iff } \Phi \text{ is satisfied by } \overline{a}.$$

Indeed, to see the 'if' direction note that the number ℓ_0 of unsatisfied (by $\overline{a}$) clauses is zero so that Lemma 3.5 gives that each of the $t_j^{\Phi}(\overline{a})$'s, and consequently the sum $T^{\Phi}(\overline{a})$, is zero. Conversely, if $\overline{a}$ does not satisfy Φ then $1 \leqslant \ell_0$, which together with $\ell_0 \leqslant \ell < p_1^{\nu_1} \cdot \ldots \cdot p_{\omega}^{\nu_{\omega}}$ gives that at least one of the $p_j^{\nu_j}$'s does not divide ℓ_0. Thus for this j the summand $\frac{m}{p_j} \cdot t_j^{\Phi}(\overline{a})$ is non-zero and – by the unique decomposition – the entire sum $T^{\Phi}(\overline{a}) \neq 0$.

Now, the circuit required in Proposition 3.1(1) is not supposed to calculate separately each of the $\Gamma(t_j^{\Phi})$'s by summing up the subexpressions b $\left(\sum_{i=1}^{n} \beta_i x_i + c\right)$ of t_j^{Φ}. Instead, each such subexpression is calculated by the gate $\Gamma_{\beta,c}(t_j^{\Phi})$ and then sent to the $\mathrm{MOD}_m^{\{0\}}$-gate $\left(\frac{m}{p_j} \cdot \alpha_{\beta,c}\right)$-times. Due to the properties of T^{Φ}, this last gate, after collecting all the bundles $\Theta(t_j^{\Phi})$'s, calculates the boolean value of $\Phi(\overline{a})$. Moreover the entire circuit consists of $1 + \sum_{j=1}^{\omega} \left|L(t_j)\right|$ gates:

- the final gate $\mathrm{MOD}_m^{\{0\}}$,
- the gates $\Gamma_{\beta,c}(t_j^{\Phi})$ of the form $\mathrm{MOD}_{p_j}^{\mathbb{Z}_{p_j}-\{-c\}}$, one for each $(\beta, c) \in L(t_j)$.

From Lemma 3.5 we know that the sizes of the t_j's (and therefore of the $\Theta(t_j^{\Phi})$'s) can be uniformly bounded by $O(2^{c\ell^{1/\omega}} \log \ell)$. Thus this also bounds the size of the circuit. □

Note here that in our construction of subexponential size $\mathrm{CC}_2[m]$-circuit computing AND the final gate is $\mathrm{MOD}_m^{\{0\}}$. This contrasts the result of Caussinus [6] where the lower bound $2^{\Omega(n)}$ is shown if the final gate is $\mathrm{MOD}_m^{\{1,\ldots,m-1\}}$.

4 MAKING THE CIRCUITS SMALLER

We start with observing that composing $\mathrm{CC}_2[m]$ circuits by using 2 separate groups of 2 levels we can keep the size $2^{O(k^{1/\omega} \log k)}$ to compute AND_{k^2}. Indeed we can simply feed the k inputs of the last two levels computing AND_k by the outputs of k-ary independent conjunctions built on the 2 starting levels. Repeating this recursively $\lfloor h/2 \rfloor$-many times on h levels we get the following Proposition.

Proposition 4.1. *For $h \geqslant 2$ and a positive integer m with $\omega = \omega(m) \geqslant 2$ there are $\mathrm{CC}_h[m]$-circuits of size $2^{O(n^{1/(\omega\lfloor h/2 \rfloor)} \log n)}$, computing n-ary* AND.

Our next step is to use both the depth h of the circuit and the width $\omega(m)$ of the modulus to make our $\mathrm{CC}_h[m]$-circuits for AND much smaller. But before doing that we warm up with the following easy observation. The idea of its proof has been already explored in [19, 20, 27].

Proposition 4.2. *For $h \geqslant 2$ and a sequence of alternating primes $p_1 \neq p_2 \neq p_3 \neq \ldots \neq p_h$ there are $\mathrm{CC}[p_1; \ldots; p_h]$-circuits of size $2^{O(n^{1/(h-1)})}$, computing n-ary* AND.

Proof. Obviously we may assume that $n = k^{h-1}$ for some k. For each $j = 1, \ldots, h-1$ Fact 3.3 supplies us with a k-ary $\mathbb{Z}[p_j, p_{j+1}]$-expression C_j of size $2^{O(k)}$ that on $\overline{a} \in \{0,1\}^k \subseteq \mathbb{Z}_{p_j}^k$ behaves as AND_k. On the starting level of our circuit we group $n = k^{h-1}$ inputs into n/k groups of k inputs each. Then each group is passed through the bunch $\Theta(C_1)$ so that we end up with n/k bundles B_i. Note that if B_i was passed through $\mathrm{MOD}_{p_2}^{\{1\}}$ gate we would get the conjunction of k inputs of B_i. Instead we again group the bundles $B_1, \ldots, B_{n/k}$ into n/k^2 groups with k bundles each and pass each such a group through the bunch $\Theta(C_2)$. Again, the sum of each of the n/k^2 resulting bundle (modulo p_3) coincide with AND of k^2 on the initial inputs that fall into that bundle. After repeating this $h-1$ times we end up with a single bundle of type p_h. At this point we actually use $\mathrm{MOD}_{p_h}^{\{1\}}$ gate to sum this bundle up and get AND of all the inputs.

It should be clear that the size of the entire circuit is bounded by $2^{O(k)} = 2^{O(n^{1/(h-1)})}$. □

Now we are in a position to prove Theorem 1.2. However we will start with its slightly weaker version.

Proposition 4.3. *For $h \geqslant 3$ and a positive integer m with $\omega = \omega(m) \geqslant 2$ and $\omega' = \omega'(m)$ there are $\mathrm{CC}_h[m]$-circuits of size $2^{O(n^{1/((\omega-1)(h-2)+(\omega'-1))} \log n)}$, computing n-ary* AND.

Proof. As in the proof of Proposition 3.1 we start with the prime decomposition $m = p_1^{\alpha_1} \cdot \ldots p_{\omega}^{\alpha_{\omega}}$ and assume that $p_1 > \ldots > p_{\omega'} \geqslant \omega > p_{\omega'+1} > \ldots > p_{\omega}$. Moreover, without loss of generality we assume that $n = k^{(\omega-1)(h-2)+(\omega'-1)}$ for some integer k and put $k_{\omega} = (\omega-1)k^{\omega-1}$ and $k_{\omega'} = (\omega-1)k^{\omega'-1}$. Finally we pick integers

$$\begin{aligned} &\nu_1, \ldots, \nu_{\omega} && \text{satisfying } p_j^{\nu_j - 1} \leqslant k_{\omega}^{1/(\omega-1)} < p_j^{\nu_j}, \\ & && \text{so that } \textstyle\prod_{j \neq i} p_j^{\nu_j} > k_{\omega}, \text{ for } i = 1, \ldots, \omega, \\ &\overline{\nu}_1, \ldots, \overline{\nu}_{\omega'} && \text{satisfying } p_j^{\overline{\nu}_j - 1} \leqslant k_{\omega'}^{1/(\omega'-1)} < p_j^{\overline{\nu}_j}, \\ & && \text{so that } \textstyle\prod_{j \neq i} p_j^{\overline{\nu}_j} > k_{\omega'}, \text{ for } i = 1, \ldots, \omega'. \end{aligned}$$

Also for two different prime divisors p, q of m we modify k_{ω}-ary and $k_{\omega'}$-ary $\mathbb{Z}[p, q]$-expressions of the form t_{pq} that satisfy (1) to $t'_{pq} = 1 - t_{pq}$ with the arity that later should be clear from the context. Note here that, except their arities, the $t_{p_i p_j}$'s depend not only on the primes p_i, p_j but also on the integers ν_j (or $\overline{\nu}_j$, whatever applies).

By Γ'_{pq} and Θ'_{pq} we denote the circuit $\Gamma(t'_{pq})$ and the bunch $\Theta(t'_{pq})$ of type $[p, q]$, respectively.

Note that for fixed p_i and $z_1, \ldots, z_{k_{\omega}} \in \{0,1\}$ we have

$$\mathrm{AND}\left\{t'_{p_i p_j}(z_1, \ldots, z_{k_{\omega}}) : j \neq i\right\} = \mathrm{AND}\left\{z_1, \ldots, z_{k_{\omega}}\right\}. \tag{3}$$

Indeed, Lemma 3.5 assure us that the left hand side in the above display is 1 if and only if for all $j \neq i$ the number of zeros among the z's is divisible by $p_j^{\nu_j}$. This in turn means that the number of zeros among the z's is divisible by $\prod_{j\neq i} p_j^{\nu_j} > \prod_{j\neq i} k_\omega^{1/(\omega-1)} = k_\omega$. But there are only k_ω places for such zeros so that there are no zeros among the z's at all.

Now for each $h' = 0, 1, 2, \ldots, h-2$ we recursively built a circuit $\nabla_{h'}$ of depth h'

(i) with n inputs $x_1, \ldots, x_n$, (repeated $\omega(\omega-1)$ times by ∇_0)
(ii) and with $b_{h'} = \omega(\omega-1) \cdot n/k^{(\omega-1)h'} = \omega(\omega-1) \cdot k^{(\omega-1)(h-2-h')+(\omega'-1)}$ bundles of outputs.

For $h' > 0$ each bundle mentioned in (ii) is the result of some bunch of the form Θ_{pq}. Thus each bundle has one of the types $p_1, \ldots, p_\omega$ and all the bundles are evenly divided into these types so that

(iii) there are $b_{h'}/\omega = (\omega-1) \cdot k^{(\omega-1)(h-2-h')+(\omega-1)}$ bundles of each type.

Moreover enlarging $\nabla_{h'}$ to $\nabla_{h'+1}$ we will keep the following properties:

(iv) summing up (modulo q) a bundle B of type q (and denoting this sum by $s_B(\overline{x})$) only the boolean values 0 or 1 may appear,
(v) the conjunction of all $b_{h'}$ values $s_B(\overline{x})$ (i.e. with B ranging over all bundles produced by $\nabla_{h'}$) coincides with $\mathsf{AND}(x_1, \ldots, x_n)$.

We start with artificially adding level 0 just to multiply variables so that it does not contribute to the depth of our circuits. In fact this starting circuit ∇_0 (of depth 0) takes n inputs $x_1, \ldots, x_n$ and makes $b_0 = \omega(\omega-1) \cdot n$ bundles, each of which consisting of one typed variable, i.e. each variable x_i is repeated $\omega - 1$ times in each type. It should be (more than) obvious that (i)-(v) hold.

Now to go from $\nabla_{h'}$ to $\nabla_{h'+1}$ we first group $\frac{b_{h'}}{\omega}$ bundles of a given type, say p, into $\frac{b_{h'}}{\omega k_\omega}$ groups of size k_ω (i.e. each such a group consists of k_ω bundles of type p). Next, all k_ω bundles in one group are passed through $\omega - 1$ bunches Θ'_{pq}, one for each $q \neq p$, to produce $\omega - 1$ bundles, again one for each type $q \neq p$. Thus $b_{h'}$ bundles (that go to the gates on level $h'+1$) are replaced by $b_{h'+1} = (\omega-1)\frac{b_{h'}}{k_\omega} = \frac{b_{h'}}{k^{\omega-1}}$ new bundles, as required in (i)-(iii). To pass the k_ω-element group $B_1, \ldots, B_{k_\omega}$ of bundles through the bunch Θ'_{pq} of gates we inflate each single input (say the r-th one) of Θ'_{pq} into the number of outputs in B_r so that in fact Θ'_{pq} is fed by $s_{B_1}, \ldots, s_{B_{k_\omega}}$.

To see (iv), say for a bundle B of type q, note that $s_B(\overline{x}) = t'_{pq}(s_{B_1}(\overline{x}), \ldots, s_{B_{k_\omega}}(\overline{x}))$, where $B_1, \ldots, B_{k_\omega}$ form the k_ω element group of bundles (of type p) that were passed through Θ'_{pq}. Since t'_{pq} returns boolean values on boolean arguments, we get (iv).

To prove (v) let $C_1, \ldots, C_{\omega-1}$ be the bundles resulting from passing the k_ω-element group $B_1, \ldots, B_{k_\omega}$ of bundles of type p through $\omega - 1$ bunches Θ'_{pq} (with $q \neq p$). If C_r is of type q then $s_{C_r}(\overline{x}) = t'_{pq}(s_{B_1}(\overline{x}), \ldots, s_{B_{k_\omega}}(\overline{x}))$ and consequently

$$\mathsf{AND}(s_{C_1}(\overline{x}), \ldots, s_{C_{\omega-1}}(\overline{x})) = \mathsf{AND}\Big\{t'_{pq}(s_{B_1}(\overline{x}), \ldots, s_{B_{k_\omega}}(\overline{x})) : q \neq p\Big\}.$$

Due to the equation (3) the last conjunction is equal to $\mathsf{AND}(s_{B_1}(\overline{x}), \ldots, s_{B_{k_\omega}}(\overline{x}))$. Thus the two conjunctions of all the sums of the form $s_B(\overline{x})$: one before processing the bundles through a given level and the other one after processing them, are equal. This shows (v).

After arriving at the level $h-2$, our circuit ∇_{h-2} produces $b_{h-2} = \omega(\omega-1) \cdot k^{\omega'-1} = \omega k_{\omega'}$ bundles, i.e. $k_{\omega'}$ bundles in each type. Now we put all these $k_{\omega'}$ bundles of one type, say p, into one group and proceed this group through $\omega' - 1$ bunches Θ'_{pq} with q ranging over some $\omega' - 1$ element subset $Q_p \subseteq \{q \neq p : q \geqslant \omega\}$. Again, as in the proof of invariant (v), we argue that $\mathsf{AND}(s_{C_1}(\overline{x}), \ldots, s_{C_{\omega'-1}}(\overline{x})) = \mathsf{AND}(s_{B_1}(\overline{x}), \ldots, s_{B_{k_{\omega'}}}(\overline{x}))$ where $C_1, \ldots, C_{\omega'-1}$ are the bundles resulting from passing $k_{\omega'}$-element group $B_1, \ldots, B_{k_{\omega'}}$ of bundles of type p through the bunches Θ'_{pq} with $q \in Q_p$. The output of the $(h-1)$-th level consists of $\omega(\omega'-1)$ bundles, as each of the ω groups is passed through $\omega' - 1$ bunches Θ'_{pq} with large primes q. On the other hand for a fixed large q at most $\omega - 1$ primes $p \neq q$ may contribute to the bunches Θ'_{pq} that are actually used on level $h-1$. To distinguish those primes we put $Z_j = \Big\{i : \Theta'_{p_i p_j} \text{ is used on level } h-1\Big\}$ for $j \leqslant \omega'$. Note that $\{1, \ldots, \omega'\} - \{j\} \subseteq Z_j \subseteq \{1, \ldots, \omega\} - \{j\}$, i.e. in particular $|Z_j| \leqslant \omega - 1 < p_j$. In this notation we enumerate all the $|Z_1| + \ldots + |Z_{\omega'}|$ bundles resulting from level $h-1$ by $C_1^1, \ldots, C_1^{|Z_1|}, C_2^1, \ldots, C_2^{|Z_2|}, \ldots\ldots, C_{\omega'}^1, \ldots, C_{\omega'}^{|Z_{\omega'}|}$. Denoting $s_{C_j^i}(\overline{x})$ simply by $s_j^i(\overline{x})$ we now express the property (v) as

$$\mathsf{AND}(x_1, \ldots, x_n) = \mathsf{AND}\Big\{s_j^i(\overline{x}) : j \leqslant \omega' \text{ and } i \in Z_j\Big\}. \tag{4}$$

Now, at the very last level we put all $|Z_1| + \ldots + |Z_{\omega'}|$ bundles, with C_j^i being repeated $\frac{m}{p_j}$ times, into the gate $\mathsf{MOD}_m^{\{\sigma\}}$, where $\sigma = \sum_{j\leqslant\omega'} \frac{m}{p_j} \cdot |Z_j| \mod m$. This gate computes (modulo m) the sum

$$S(\overline{x}) = \sum_{j\leqslant\omega'} \sum_{i\in Z_j} \frac{m}{p_j} \cdot s_j^i(\overline{x})$$

and turns it to 1 if $S(\overline{x}) = \sigma$ and to 0 otherwise. Thus, due to (4), we are left with showing that $S(\overline{x}) = \sigma$ iff $s_j^i(\overline{x}) = 1$ for all $j \leqslant \omega'$ and $i \in Z_j$. Obviously if all the $s_j^i(\overline{x})$'s are 1 then the sum $S(\overline{x})$ is σ. Conversely, as in the proof of Proposition 3.1, we first identify the direct sum $\bigoplus_{j=1}^{\omega'} \frac{m}{p_j} \cdot \mathbb{Z}_{p_j}$ with a subgroup of $\mathbb{Z}_m$. Then the assumption that $\sigma = S(\overline{x}) = \sum_{j\leqslant\omega'} \frac{m}{p_j} \cdot \sum_{i\in Z_j} s_j^i(\overline{x})$ together with the fact that $0 \leqslant \sum_{i\in Z_j} s_j^i(\overline{x}) \leqslant |Z_j| \leqslant \omega - 1 < p_j$ gives, by the unique decomposition in the direct sum, that for each $j \leqslant \omega'$ we have $\sum_{i\in Z_j} s_j^i(\overline{x}) = |Z_j| \mod p_j$. But now $s_j^i(\overline{x}) \in \{0, 1\}$ and $|Z_j| < p_j$ yield that all the $s_j^i(\overline{x})$'s are 1.

It remains to calculate the size of the entire circuit. Each of the first $h-2$ levels has $2^{O(p_i^{\nu_i} \log k_\omega)}$ gates in each bunch of type p_i. There are at most $O(n)$ bunches of each type. Using $p_i^{\nu_i} \leqslant p_i k_\omega^{1/(\omega-1)} \in O(k)$ we bound the size of each bunch by $2^{O(k \log k)}$. The same holds on the level $h-1$. Summing up we bound the size of entire circuit by $2^{O(k \log n)} \leqslant 2^{O(n^{1/((\omega-1)(h-2)+(\omega'-1))} \log n)}$. □

Now we are ready to show Theorem 1.2 that, in comparison to Proposition 4.3, increases the degree of the root just by one.

Proof. Our circuits here are based on those from the proof of Proposition 4.3 by modifying only two levels: ∇_0 and ∇_1. This time we start with assuming that $n = k^{(\omega-1)(h-2)+\omega'}$ for some integer k. Also, additionally to the ν_j's and the $\overline{\nu}_j$'s (exactly as in the proof of Proposition 4.3) we pick ν_j^0 to satisfy $p_j^{\nu_j^0-1} \leqslant k < p_j^{\nu_j^0}$ for all the j's. The starting circuit ∇_0 takes n inputs $x_1, \ldots, x_n$ and makes $b_0 = \omega \cdot n$ bundles, each of which consisting of one typed variable, so that there are exactly n bundles in each type. To proceed these bundles through the gates of ∇_1 we will group n bundles in each type into groups of size $k_0 = k^\omega$, but in a synchronized way. By this synchronization we mean that first the set $\{1, \ldots, n\}$ is split into n/k_0 groups G_i of size k_0 and then in each type, say p, we form a group of bundles $G_i^p = \{x_j : j \in G_i\}$. Next, each such group G_i^p is passed though all the Θ'_{pq}'s (with $q \neq p$) to get the bundles $\Theta'_{pq}(G_i^p)$. As previously we want to have that $\mathsf{AND}(x_1, \ldots, x_n)$ coincides with the conjunction of all the $s_B(\overline{x})$'s with B ranging over all the bundles produced by ∇_1, i.e. that:

$$\mathsf{AND}(x_1, \ldots, x_n) = \mathsf{AND}\left\{s_{\Theta_{pq}(G_i^p)}(\overline{x}) : \ p \neq q, \ i = 1, \ldots, n/k_0\right\}.$$

We get this by observing that $\mathsf{AND}(G_i^p)$ can be replaced by the conjunction of $s_B(\overline{x})$ for (at least ω) bundles B of all ω different types $p_1, \ldots, p_\omega$. This however is witnessed by

$$\mathsf{AND}(G_i^p) = \mathsf{AND}\left(\left\{s_{\Theta_{pq}(G_i^p)}(\overline{x}) : \ q \neq p\right\} \cup \left\{s_{\Theta_{qp}(G_i^q)}(\overline{x}) : \ q \neq p\right\}\right),$$

due to the fact that for a fixed i our synchronization spans the sets G_i^p and G_i^q on the very same variables.

In this process ∇_1 replaces each group of $k_0 = k^\omega$ bundles by $\omega-1$ new bundles. This means that ∇_1 produces $b_1 = (\omega-1)\cdot\frac{b_0}{k^\omega} = (\omega-1)\omega \cdot k^{(\omega-1)(h-3)+(\omega'-1)}$ bundles, which is exactly the number of bundles produced by ∇_1 in the proof of Proposition 4.3. This allows us to put these bundles into the consecutive levels of the circuit described in that proof.

As previously our choice of $k_0, k_\omega, k_{\omega'}$ (for determining the sizes of the groups of bundles) yields that, on each level, the sizes of the bunches used in our circuit are bounded by $2^{O(k \log k)}$. Combining this with the fact that on each level at most $O(n)$ bunches are used and with $n = k^{(\omega-1)(h-2)+\omega'}$ we can bound the size of our circuit by $2^{O(n^{1/((\omega-1)(h-2)+\omega')} \log n)}$. □

5 PROBABILISTIC CIRCUITS

In this section we prove Theorem 1.3, i.e. we construct polynomial size $\mathrm{CC}[p;q]$-circuits Γ_n computing AND_n with the help of $l = 6 + \log n$ additional random bits. This means that Γ_n has $n + l$ inputs and for each n-tuple $\overline{a} \in \{0,1\}^n$ for at least $\frac{2}{3}$ possible tuples $\overline{b} \in \{0,1\}^l$ we have $\Gamma_n(\overline{a}, \overline{b}) = \mathsf{AND}_n(\overline{a})$. These circuits will be based on $O(l)$-ary special $\mathbb{Z}[p,q]$-expressions so that we can control their size to be polynomial in n, i.e. $2^{O(l)}$.

To start our construction define Λ to be the set of all tuples $\lambda = \left(\lambda_{\overline{c},j}\right)_{j=1,\ldots,p^*l}^{\overline{c}\in\{0,1\}^l}$ of length $2^l p^* l$, where $p^* = \lceil \log_{\frac{p}{p-1}} 2 \rceil$ and each $\lambda_{\overline{c},j}$ is an $GF(p)$-affine combination of the x_i's satisfying $\lambda_{\overline{c},j}(1, \ldots, 1) = 1$. Define $\mathrm{b}'(z) = 1 - \mathrm{b}(z)$ so that for $\lambda \in \Lambda$ we put

$$t_\lambda(\overline{x}, \overline{b}) = \sum_{\overline{c}\in\{0,1\}^l} \prod_{i=1}^{l} \mathrm{b}'(b_i - c_i) \cdot \prod_{j=1}^{p^*l} \mathrm{b}(\lambda_{\overline{c},j}(\overline{x})),$$

to show that

- each $t_\lambda(\overline{x}, \overline{b})$ can be turned into $\mathbb{Z}[p,q]$-expression with $2^{O(l)}$ summands (corresponding to the number of gates in the circuits realizing this expression),
- for at least one $\lambda \in \Lambda$ the expression $t_\lambda(\overline{x}, \overline{b})$ calculates $\mathsf{AND}_n(\overline{x})$ for at least $\frac{2}{3}$ of the $\overline{b}$'s in $\{0,1\}^l$.

For the first item note that each summand in $t_\lambda(\overline{x}, \overline{b})$ can be obtained by an appropriate substitution in a $(l+p^*l)$-ary function $Z_p^l \times Z_p^{p^*l} \ni (\overline{u}, \overline{z}) \longmapsto \mathrm{b}'(u_1) \cdot \ldots \cdot \mathrm{b}'(u_l) \cdot \mathrm{b}(z_1) \cdot \ldots \cdot \mathrm{b}(z_{p^*l}) \in Z_q$. By Fact 3.3 such function can be represented by a $\mathbb{Z}[p,q]$-expression with $O(p^{(p^*+1)l})$ summands. Now, summing up (modulo q) over the $\overline{c}$'s we end up with a $\mathbb{Z}[p,q]$-expression with $O(2^l p^{(p^*+1)l}) = 2^{O(l)} = poly(n)$ summands.

Before showing the second item note that for fixed $\overline{b} \in \{0,1\}^l$ the expression $t_\lambda(\overline{x}, \overline{b})$ reduces to only one summand, namely $\prod_{j=1}^{p^*l} \mathrm{b}(\lambda_{\overline{b},j}(\overline{x}))$. Now, for a fixed $\overline{a} \in \{0,1\}^n$ and $\overline{b} \in \{0,1\}^l$ the random variable $X_{\overline{a},\overline{b}}$ checks for a particular tuple $(\lambda_{\overline{b},j})_{j=1,\ldots,p^*l}$ if the value $\prod_{j=1}^{p^*l} \mathrm{b}(\lambda_{\overline{b},j}(\overline{a}))$ coincides with $\mathsf{AND}_n(\overline{a})$. Thus the sum $X_{\overline{a}} = \sum_{\overline{b}\in\{0,1\}^l} X_{\overline{a},\overline{b}}$, defined now on entire Λ, simply counts the number of the $\overline{b}$'s for which $t_\lambda(\overline{a}, \overline{b}) = \mathsf{AND}_n(\overline{a})$. We conclude our argument with showing that $\Pr\left[\bigwedge_{\overline{a}\in\{0,1\}^n} X_{\overline{a}} \geqslant \frac{2}{3} \cdot 2^l\right] \neq 0$. Note that for fixed $\overline{a} \neq \overline{1}$ and randomly chosen $\lambda_{\overline{c},j}$ we have $\Pr\left[\lambda_{\overline{c},j}(\overline{a}) \neq 0\right] = \frac{p-1}{p}$ so that $\Pr\left[X_{\overline{a},\overline{b}} = 0\right] = \left(\frac{p-1}{p}\right)^{p^*l} = 2^{-l}$ and $E(X_{\overline{a},\overline{b}}) = 1 - 2^{-l}$. Consequently $E(X_{\overline{a}}) = 2^l(1 - 2^{-l}) = 2^l - 1$. Fixing δ so that $(1-\delta)E(X_{\overline{a}}) = \frac{2}{3} \cdot 2^l$ we apply Chernoff's inequality for the lower tail to get $\Pr\left[X_{\overline{a}} \leqslant \frac{2}{3} \cdot 2^l\right] \leqslant \exp\left(-\frac{E(X_{\overline{a}})\cdot\delta^2}{2}\right) \leqslant \exp\left(-\frac{64n-1}{32}\right) < 2^{-n}$. Consequently probability of the fact that no $\lambda \in \Lambda$ leads to t_λ with desired property is bounded by $\Pr\left[\bigvee_{\overline{a}\in\{0,1\}^n} X_{\overline{a}} \leqslant \frac{2}{3}2^l\right] < 2^n \cdot 2^{-n} = 1$, as required.

6 ALGORITHMS

In Sections 3 and 4 we have seen how to construct subexponential conjunctions and how it helps to encode 3-CNF SAT in satisfiability of modular circuits. Obviously better upper bounds for the size of circuits realizing AND (and consequently 3-CNF formulas) give rise to higher complexity of $\mathrm{CC}_h[m]$-SAT. In particular a polynomial upper bound for the size of AND would show NP-completeness of $\mathrm{CC}_h[m]$-SAT. Although, in Section 5 we have shown that AND can be realized by a probabilistic $\mathrm{CC}_h[m]$-circuits of polynomial size (provided $h, \omega(m) \geqslant 2$), we strongly believe that this cannot be done without those random bits.

In this section we analyze how the lower (superpolynomial) bound for the size of circuits realizing AND can be used to (subexponentially) bound the complexity of $\mathrm{CC}_h[m]$-SAT from above.

To this end for fixed depth h and modulus m by $\gamma_{h,m}(n)$ we denote the size of the smallest possible $CC_h[m]$-circuit computing AND_n. Note first that (according to Proposition 2.1) if $h = 1$ or $\omega(m) = 1$ the values $\gamma_{h,m}(n)$ are defined only for finitely many first integers n. However, independently of h and m, Fact 3.3 ensures us that $\gamma_{h,m}$ is at most exponentially large and therefore computable in 2-EXPTIME. In our considerations we need much better bound for the time needed to compute $\gamma_{h,m}(n)$. Note that the functions bounding sizes of the circuit constructed in Propositions 3.1(2), 4.2, 4.1, 4.3 and Theorem 1.2 are of the form $2^{O(n^{1/\delta}\log n)}$ and can be computed in PTIME. Although we cannot guarantee that $\gamma_{h,m}$ is PTIME-computable, it would be enough for us to bound it from below by such a function (which is still close enough to $\gamma_{h,m}$).

Now we provide two algorithms for satisfiability of $CC_h[m]$-circuits, a deterministic one and a slightly faster randomized one with running times depending on the growth rate of $\gamma_{h,m}$, or rather it inverse. For a partial increasing function $f : \mathbb{N} \longrightarrow \mathbb{N}$ by $f^{-1}(k)$ we mean the largest n with $f(n) \leqslant k$.

Theorem 6.1. *Suppose that $\gamma_{h,m}$ has PTIME-computable increasing lower bound f. Then there are two algorithms for checking if an n-ary $CC_h[m]$-circuit is satisfiable:*

- *a deterministic one with the running time: $O\left(poly\,|\Gamma| + 2^{f^{-1}|\Gamma|\cdot\log n}\cdot|\Gamma|\right)$,*
- *a randomized one with the running time: $O\left(poly\,|\Gamma| + 2^{f^{-1}|\Gamma|}\cdot|\Gamma|\right)$.*

Proof. Our deterministic algorithm is based on a brute-force search for a satisfying tuple in a relatively small set S of size $n^{f^{-1}|\Gamma|}$ consisting of all the tuples $a \in \{0,1\}^n$ with at most $f^{-1}|\Gamma|$ ones. To determine this set we first need to know the value $f^{-1}|\Gamma|$. But since f is PTIME-computable this can be done in $poly\,|\Gamma|$ steps. This together with checking whether S contains a satisfying tuple takes $poly\,|\Gamma| + O\left(|\Gamma|\cdot n^{f^{-1}|\Gamma|}\right)$ steps, as claimed.

Since $\gamma^{-1}_{h,m} \leqslant f^{-1}$ we are left with showing that if Γ is satisfiable then it can be satisfied by a tuple $a \in \{0,1\}^n$ with $\left|a^{-1}(1)\right| \leqslant \gamma^{-1}_{h,m}|\Gamma|$. Suppose then that a is a non-zero satisfying tuple with the minimal number of ones. By this minimality we know that Γ with all the inputs with indices outside $a^{-1}(1)$ set to 0 behaves like the $\left|a^{-1}(1)\right|$-ary AND. Thus $\gamma_{h,m}\left|a^{-1}(1)\right| \leqslant |\Gamma|$ so that the tuple a has at most $\gamma^{-1}_{h,m}|\Gamma|$ ones.

On the other hand our second algorithm, the probabilistic one, is based on randomly choosing sufficiently many inputs so that the probability of having a satisfying one among them exceeds $1/2$, if there is any such satisfying tuple at all. We claim that $2^{\gamma^{-1}_{h,m}|\Gamma|}$ samples suffices. Indeed, if Γ is constant then any single sample witnesses its (un)satisfiability. Remark 2.3 allows us to modify a nonconstant circuit Γ to get $(n-k)$-ary spike circuit Γ' for some $k \leqslant \log|D(\Gamma)|$, so that $|\Gamma'| \geqslant \gamma_{h,m}(n-k)$. Consequently $|\Gamma| \geqslant |\Gamma'| \geqslant \gamma_{h,m}(n-\log|D(\Gamma)|)$, which together with $\left|\Gamma^{-1}(1)\right| \geqslant |D(\Gamma)|$ gives $\left|\Gamma^{-1}(1)\right|/2^n \geqslant 2^{-\gamma^{-1}_{h,m}|\Gamma|}$. This simply means that we will find a tuple from $\Gamma^{-1}(1)$ among $2^{\gamma^{-1}_{h,m}|\Gamma|}$ samples. But again, to calculate how long we need to sample we increase $2^{\gamma^{-1}_{h,m}|\Gamma|}$ to $2^{f^{-1}|\Gamma|}$ and use the fact that f is PTIME-computable. □

From our proof of Theorem 6.1 we get the following generalization of Corollary 2.4.

Corollary 6.2. *The balance of a $CC_h[m]$-circuit Γ is at least $2^{1-\gamma^{-1}_{h,m}|\Gamma|}$.*

Observe here, that like in Corollary 2.5, we can use the function $\gamma_{h,m}$ to bound from below the number of words (of a given length) in a language recognizable by polynomial size $CC_h[m]$-circuits. In particular the suspected lower bound for $\gamma_{h,m}$ of the form $2^{\Omega(n^\delta)}$ translates into the bound $2^{n-O(\log^{1/\delta}n)}$.

Although our random sampling algorithm RanSam described in the proof of Theorem 6.1 is not involved, the proof itself tells that a bigger lower bound for $\gamma_{h,m}$ allows us to reduce the number of samples in RanSam. Below we show that this connection is two-sided.

Proposition 6.3. *If RanSam works (with probability at least $1/2$) with at most $2^{f|\Gamma|}$ samples for some increasing computable function f, then $f^{-1}(n) \leqslant \gamma_{h,m}(n+1)$.*

Proof. We run RanSam on the circuit AND_n with $2^{f\gamma_{h,m}(n)}$ samples, so that the expected number of satisfying tuples is $2^{f\gamma_{h,m}(n)}/2^n$. This procedure however has to find, with probability at least $1/2$ the unique satisfying tuple. Thus, Markov inequality yields $2^{f\gamma_{h,m}(n)}/2^n \geqslant 1/2$ so that $f^{-1}(n-1) \leqslant \gamma_{h,m}(n)$. □

Combining Theorem 6.1 and Proposition 6.3 we get that the suspected lower bound $2^{\Omega(n^\delta)} \leqslant \gamma_{h,m}$ is equivalent to the upper bound $2^{O(\log^{1/\delta}|\Gamma|)}$ for the running time of RanSam. Actually any superpolynomial lower bound for $\gamma_{h,m}$ mutually translates into substantially subexponential (i.e. at most $2^{|\Gamma|^{o(1)}}$) number of samples.

As for now only slightly superlinear lower bounds $\Omega(n\cdot\varepsilon(n))$ are known for the number of wires in $CC_h[m]$-circuits computing AND_n, where $\varepsilon(n)$ is an extremely slowly increasing function (see [10]). Although the functions $\varepsilon(n)$ depend on h and m, a careful inspection of their description shows that the inverse to $n\cdot\varepsilon(n)$ is always bounded by $O(n/\varepsilon(n))$. Thus, arguing as in the proof of Theorem 6.1, but with $|\Gamma|$ replaced by the number $w(\Gamma)$ of wires in the circuit Γ, we get the following.

Theorem 6.4. *Satisfiability of $CC_h[m]$-circuits Γ is solvable in probabilistic $2^{O(w(\Gamma)/\varepsilon(w(\Gamma)))}$ time.*

This Theorem stays in a big contrast to the lower bound $2^{\Omega(w(\Gamma))}$ (provided by the randomized version of ETH [5]) for probabilistic algorithms for satisfiability of AC^0-circuits.

We conclude this section with arguing that (under some additional assumption about effective coding of 3-CNF formulas by modular circuits) the running time of RanSam is hard to beat. Our heuristic assumption simply says that there is a PTIME algorithm that turns 3-CNF formulas Φ with ℓ clauses into $CC_h[m]$-circuits of size bounded by $O(\gamma_{h,m}(c\ell))$. Assuming also that $\gamma_{h,m}$ (or some of its Θ-equivalents) is PTIME-computable we know that RamSam

runs with $2^{O(\gamma_{h,m}^{-1}|\Gamma|)}$ samples. On the other hand ETH, applied to the circuit Γ produced from 3-CNF formula by the algorithm supplied by our heuristic assumption, gives an integer $d > 0$ so that $\mathrm{CC}_h[m]$-SAT cannot be solved in $O(2^{\frac{1}{d}\gamma_{h,m}^{-1}|\Gamma|})$. Thus the best imaginable algorithm solving $\mathrm{CC}_h[m]$-SAT has the running time bounded by a polynomial applied to the running time of RanSam.

7 CONCLUDING REMARKS AND APPLICATIONS

In view of the results in Section 4, in particular a spectacular role played by $\omega'(m)$, as well as the easiness of increasing the degree of the root just by 1, it seems to be really hard to state reasonable conjectures for the asymptotic behaviour of the $\gamma_{h,m}$'s. As for now, for $h = 2$ the degree of the root (occurring in the exponent) is at least $\omega(m)$ (Proposition 3.1) and for $h \geqslant 3$ at least $(\omega - 1)(h-2)+\omega'$ (Theorem 1.2). However for the 'majority' of potential moduli m we know that $\omega'(m)$ is pretty close to $\omega(m)$, so that this degree is almost $(\omega - 1)(h-1)+1$ (and coincides with $\omega(m)$, whenever $h = 2$). Due to the fact that prime factorization (i.e. the number $\omega(m)$) may contribute fully into this degree and the depth h contributes by the factor $h - 1$, it seems natural to suspect that the bound for $\log \gamma_{h,m}(n)$ could be of the form $n^{1/(\omega(h-1))} \log n$.

Another remark we want to make here is the difference between the circuits of the form $\mathrm{CC}_h[p;q;p;q;\ldots]$ (with $p \neq q$) and $\mathrm{CC}_h[p \cdot q]$. In the later case we have $\omega' = \omega = 2$ so that the bound for the considered degree is h, while Proposition 4.2 gives degree $h - 1$ in the first case. Moreover, it seems that there is no room for improving this $h - 1$ in this case.

This difference in locations of primes on different levels is even more striking for $\mathrm{CC}_2[p;m]$ and $\mathrm{CC}_2[m;p]$, whenever m has $r \geqslant 2$ prime divisors except p. In the first case we can actually argue, as in the proof of Proposition 3.1, to get the upper bound $2^{n^{1/r}\log n}$ for $\gamma_{h,m}$, while [2, 26] give $2^{\Omega(n)}$ lower bound in case of $\mathrm{CC}_2[m;p]$.

The technique we have developed for proving Proposition 3.1 can be used to determine (modulo ETH) the complexity of solving equations over the dihedral groups $\mathbf{D}_{2k+1}$, i.e. groups of symmetries of regular polygons with odd number of sides. Some of the variables in these equations are already preevaluated (as otherwise every equation has a trivial solution with all the variables set to the neutral element of the group). This is equivalent to consider polynomials (instead of terms) over groups. The decision version of this problem for the group $\mathbf{G}$ is denoted by PolSat($\mathbf{G}$). Analogously by PolEqv($\mathbf{G}$) we mean the problem of deciding whether two polynomials over $\mathbf{G}$ define the same function. Note here that from the paper [12] of Goldmann and Russell we know that PolSat is NP-complete for nonsolvable groups and in PTIME for nilpotent groups. Moreover our paper [20] partially fills this gap by showing that (modulo ETH) PolSat($\mathbf{G}$) is not in PTIME unless $\mathbf{G}$ has Fitting length at most 2, e.g. if $\mathbf{G}$ is a wreath product of two nilpotent groups. This paper refutes a long standing belief that PolSat for all solvable groups is in PTIME. The conjecture was based on many examples of groups that are in fact 2-nilpotent. The very recent paper of Földvári and Horváth [11] summarizes most of these examples by showing that PolSat(G) is in PTIME whenever $\mathbf{G}$ is a semidirect product of a p-group and an abelian group. Note here that the dihedral groups $\mathbf{D}_{p^k}$, with prime p, fall into this realm. On the other hand our characterization below dismisses such a speculation about tractability of PolSat for groups of Fitting length 2 (unless ETH fails).

Theorem 7.1. *If ETH holds then for each odd integer $m \geqslant 3$ the problem PolSat($\mathbf{D}_m$) is in PTIME iff $\omega(m) = 1$.*

Proof. Remind that the dihedral group $\mathbf{D}_m$ is generated by two elements, a rotation ρ (with angle $2\pi/m$) and a reflection σ satisfying $\rho^m = 1, \sigma^2 = 1$ and $\sigma\rho = \rho^{-1}\sigma$. This means that $\mathbf{D}_m$ has $2m$ elements: m rotations $\rho^0, \rho^1, \rho^2, \ldots, \rho^{m-1}$ and m reflections $\sigma, \sigma\rho, \sigma\rho^2, \ldots, \sigma\rho^{m-1}$.

If $\omega(m) = 1$ then we have already noted that [11] puts PolSat($\mathbf{D}_m$) into PTIME.

Now suppose $m = p_1^{\alpha_1} \cdot \ldots p_\omega^{\alpha_\omega}$, where the p_j's are pairwise different odd primes and $\omega \geqslant 2$. Since the rotations form a cyclic group isomorphic to $\mathbb{Z}_m$ for each $j = 1, \ldots, \omega$ there is a rotation, say ρ_j, generating a cyclic subgroup of order p_j.

Define unary polynomials (with $j = 1, \ldots, \omega$) by putting

$$\begin{aligned} \mathrm{e}(x) &= \sigma(\sigma x^m)^m, \\ \mathrm{e}_j(x) &= x^{2m/p_j}, \\ \mathrm{b}_j(x) &= (\rho_j \mathrm{e}(x)\rho_j^{-1}\mathrm{e}(x)^{-1})^{\frac{m+1}{2}}, \end{aligned}$$

and observe that the range of e is $\{1, \sigma\}$, i.e. the group isomorphic to $\mathbb{Z}_2$, while e_j maps the group $\mathbf{D}_m$ onto its cyclic subgroup $\left\{1, \rho_j, \rho_j^2, \ldots, \rho_j^{p_j-1}\right\}$ isomorphic to $\mathbb{Z}_{p_j}$. Moreover the polynomial b_j maps the group $\{1, \sigma\}$ onto $\{1, \rho_j\} \subseteq \left\{1, \rho_j, \rho_j^2, \ldots, \rho_j^{p_j-1}\right\}$ and therefore b_j can be used to build $\mathbb{Z}[2, p_j]$-expressions as polynomials of $\mathbf{D}_m$.

Now we adapt the proof of Proposition 3.1(1) to our setting. For a 3-CNF formula Φ we borrow the $\mathbb{Z}[2, p_j]$-expressions by putting $t_j^\Phi = t_{2,p_j}^\Phi$ to build the polynomial T^Φ, but we modify the original definition (2) to be read

$$T^\Phi(x_1, \ldots, x_n) = \sum_{j=1}^{\omega} t_j^\Phi(x_1, \ldots, x_n)$$

where the sum is computed in the direct sum $\bigoplus_{j=1}^{\omega} \mathbb{Z}_{p_j}$ identified with a subgroup of the group $\mathbb{Z}_m$ of all rotations. Now we simply transform 3-CNF formula Φ into the equation $T^\Phi(\overline{x}) = 0$, with 0 being the neutral element of both $\mathbb{Z}_m$ and $\mathbf{D}_m$. To see that Φ is satisfiable iff the corresponding equation has a solution in $\mathbf{D}_m$, we simply go back and forth between the boolean values and the elements of $\mathbf{D}_m$ by identifying the rotations, i.e. elements of $\mathrm{e}_0^{-1}(1)$ with the boolean value true and the reflections, i.e. elements of $\mathrm{e}_0^{-1}(\sigma)$ with the boolean value false.

Obviously, as previously, the length of T^Φ is bounded by $2^{O(\ell^{1/\omega}\log \ell)}$ where ℓ is the number of clauses in Φ. Thus ETH yields that PolSat($\mathbf{D}_m$) cannot be in PTIME. □

Very recently the authors together with A.Weiß [21] have done a full analysis of the complexity for PolSat over all dihedral groups $\mathbf{D}_m$. In particular our method used in Theorem 7.1 is substantially extended to cover, among other things, a situation where m is even but has at least two different odd prime divisors. Moreover a

randomized polynomial time algorithm is provided for PolSat($\mathbf{D}_m$) with m having at most one odd prime divisor.

Another feature of the dihedral groups $\mathbf{D}_m$ is that the problem PolEqv($\mathbf{D}_m$) is in PTIME for all m, see [4]. Thus Theorem 7.1 provides the first examples of finite groups with tractable PolEqv and untractable PolSat (modulo ETH). Note here that every group with tractable PolSat has tractable PolEqv, as to decide whether two polynomials t, s are equal we simply check that none of the $|G|-1$ equations of the form $ts^{-1} = a$ (with a ranging over $G - \{1\}$) has a solution.

Almost the same argument can be used in the setting of multivalued circuit satisfiability CSat and circuit equivalence CEqv, as defined in [22]. Such multivalued circuits are built over a fixed finite algebra $\mathbf{A}$ so that the gates here simply compute the basic operations of the algebra. In fact, our paper [22] initiated a systematic project of characterizing finite algebras $\mathbf{A}$ with CSat($\mathbf{A}$) in PTIME and provided a partial characterization for algebras from congruence modular varieties. However a somehow similar (to PolSat for groups) gap was left open, namely the unsolved complexity of CSat and CEqv for nilpotent but not supernilpotent algebras. Another of our papers [19] presents algebras $\mathbf{D}[p_1, \ldots, p_h]$ built over the alternating chain of primes $p_1 \neq p_2 \neq p_3 \neq \ldots \neq p_h$ with CSat and CEqv outside PTIME, provided $h \geqslant 3$ and ETH holds. Later the paper [25] developed these methods to actually force nilpotent algebras with CSat or CEqv in PTIME to be wreath products of two supernilpotent algebras. On the other hand, in [18] we provided examples of such wreath products (that are actually 2-nilpotent) with CSat and CEqv in PTIME. Although CEqv for all 2-nilpotent algebras has been confirmed [24] to be in PTIME, an analogue for CSat is blocked by the following example, the proof of which simply repeats the argument for Theorem 7.1.

Example 7.2. *CSat for the following 2-nilpotent algebras is outside PTIME, modulo ETH:*
For any sequence $p_0, p_1, p_2, \ldots, p_\omega$ of pairwise different primes the algebra $\mathbf{D}[p_0; p_1 \cdot \ldots \cdot p_\omega]$ is defined to be the group $\mathbb{Z}_{p_0} \times \mathbb{Z}_{p_1} \times \ldots \times \mathbb{Z}_{p_\omega}$ endowed with $2\omega + 1$ unary operations that for $\overline{x} = (x_0, x_1, \ldots, x_\omega) \in Z_{p_0} \times Z_{p_1} \times \ldots \times Z_{p_\omega}$ are defined by

$$\begin{aligned} \mathrm{e}_j(\overline{x}) &= (0, \ldots, 0, x_j, 0, \ldots, 0), && \text{for } j = 0, 1, \ldots, \omega, \\ \mathrm{b}_j(\overline{x}) &= (0, \ldots, 0, b_j^*(x_0), 0, \ldots, 0), && \text{for } j = 1, \ldots, \omega, \end{aligned}$$

where $b_j^* : Z_{p_0} \longrightarrow Z_{p_j}$ is the function given by $b_j^*(0) = 0$ and $b_j^*(a) = 1$ otherwise. □

8 EASY STUFF

Proof of Proposition 2.1. To warm up, note that for any modulus m and $n \geqslant m$ each sequence $\alpha_1, \ldots, \alpha_n$ of integers contains a nonempty subsequence $\alpha_{i_1}, \ldots, \alpha_{i_k}$ that modulo m sums up to 0. Indeed, either there is 0 among the sums $\alpha_1, \alpha_1+\alpha_2, \ldots, \alpha_1+\ldots+\alpha_n$ or at least two of them are equal, making their difference , i.e. a shorter nonempty sum, to be 0.

Now, the only (say n-ary) gate MOD_m^A in the $\mathrm{CC}_1[m]$-circuit (that takes α_i times the input x_i), after checking if $\sum_{i=1}^n \alpha_i x_i$ belongs to A, returns the very same value on the constant sequence $x_1 = \ldots = x_n = 1$ and its modification obtained by switching $x_{i_1}, \ldots, x_{i_k}$ to 0. This destroys the possibility for MOD_m^A with $n \geqslant m$ inputs to serve as AND_n.

For $\mathrm{CC}_h\left[p^k\right]$-circuits we induct on h to show how from a particular circuit Γ pass to a polynomial $w_\Gamma(\overline{x})$ over $GF(p)$ so that this polynomial:

- computes the circuit Γ,
- is presented in its sparse representation,
- contains monomials of degree d^h for some constant d depending on p^k only.

Having done that, we pick a monomial in $w_\Gamma(\overline{x})$ of minimal degree and evaluating all the x_i's occurring in this monomial by 1 and the other x_i's by 0, we know that $w_\Gamma(\overline{x}) \neq 0$. However $n > d^h$ ensures us that at least one of the x_i's is 0, contrary to the fact that Γ is supposed to compute AND_n.

To start our induction for $h = 1$ we refer to the paper [3] of Beigel and Tarui, where Lemma 2.1 supplies us with a polynomial $r_0(x_1, \ldots, x_n)$ over $GF(p)$ that on the boolean values of the x_i's behaves as the gate $\mathrm{MOD}_{p^k}^{\{0\}}$. In fact in the proof of that Lemma it is shown that the polynomial

$$r_0(x_1, \ldots, x_n) = \prod_{j=1}^{k-1} \left(1 - \left(\sum_{I \subseteq \{1, \ldots, n\}, |I| = p^j} \prod_{i \in I} x_i \right)^{p-1}\right)$$

does the job. It is easy to see that the degree d_0 of $r_0(x_1, \ldots, x_n)$ is bounded by p^{k+1}, independently of n. Since on the boolean values of the x_i's the polynomial $r_0(\overline{x})$ can be represented as

$$r_0(x_1, \ldots, x_n) = \prod_{j=1}^{k-1} \left(1 - \left(\frac{\sum_{i=1}^n x_i}{p^j} \right)^{p-1}\right)$$

we get that $r_c(x_1, \ldots, x_n) = r_0(x_1 - c, x_2, \ldots, x_n)$ computes the gate $\mathrm{MOD}_{p^k}^{\{c\}}$. Consequently $r_A(\overline{x}) = 1 - \prod_{c \in A}(1 - r_c(\overline{x}))$ computes $\mathrm{MOD}_{p^k}^{A}$. The degree of the polynomial $r(\overline{x})$ is bounded by $d = |A| \cdot d_0 \leqslant p^{2k+1}$.

Now assume that a $\mathrm{CC}_h\left[p^k\right]$-circuit Γ composes on the final level $\mathrm{CC}_{h-1}\left[p^k\right]$-circuits $\Gamma_1, \ldots, \Gamma_m$ by the gate $\mathrm{MOD}_{p^k}^A$. To get the required polynomial $w_\Gamma(\overline{x})$ we simply plug into m-ary $r_A(y_1, \ldots, y_m)$ the polynomials $w_{\Gamma_1}, \ldots, w_{\Gamma_m}$. Obviously the degree of $w_\Gamma(\overline{x})$ is bounded by the maximal degree of the w_{Γ_j}'s (i.e. by d^{h-1}) multiplied by the degree of $r_A(\overline{y})$ (i.e. by d) which gives the required bound of d^h. □

Proof of Fact 3.4. For an n-tuple of variables $\overline{x} = (x_1, \ldots, x_n)$ we define

$$v_j(\overline{x}) = \sum_{1 \leqslant i_1 < i_2 < \ldots < i_j \leqslant n} x_{i_1} \ldots x_{i_j}$$

to be the sum of all j-linear monomials over the variables $\overline{x}$. In particular $v_0(\overline{x}) = 1$. We will concentrate on their behaviour only for the boolean values 0, 1, so that we put $v_j' : \{0,1\}^n \longrightarrow \mathbb{Z}_p$ to be the appropriate restriction of v_j.

First observe that $v_0', v_1', \ldots, v_n'$ are linearly independent members of the vector space $\mathbb{Z}_p^{2^n}$. Indeed, if $\sum_{j=0}^n \alpha_j v_j' = 0$ then evaluating at $\overline{x} = (0, \ldots, 0)$ we get $\alpha_0 = 0$. Moreover inducting on j we evaluate on $\overline{x} \in \{0,1\}^n$ with 1 ocurring exactly j times to get $\alpha_j = 0$.

Now, fix $n \geqslant m = p^k$ and concentrate on the m dimensional subspace V_m of the 2^n dimensional space $\mathbb{Z}_p^{2^n}$ spanned over $v'_0, \ldots, v'_{m-1}$. One can easily see that each v'_j, and therefore each $v \in V_m$ is fully symmetric, i.e. $v(x_1, \ldots, x_n) = v(x_{\sigma 1}, \ldots, x_{\sigma n})$ for all permutation σ. This symmetry allows us to define $v[i]$ to be $v(1, \ldots, 1, 0, \ldots, 0)$ with exactly i ones.

Slightly more effort is required to show that each v'_j (and therefore each $v \in V_m$) is m-periodic, i.e. $v[i + m] = v[i]$. This reduces to show that $\binom{i+p^k}{j} = \binom{i}{j}$ modulo p, or in other words, that the prefixes of length p^k of the i-th and $i+p^k$-th rows of Pascal triangle coincide modulo p. However due to the fact that an entry in the $j+1$-th row depends only on the two values in j-th row, we are left with noticing that the mentioned coincidence holds for $i = 0$, or in other words that

$$\binom{p^k}{j} \underset{p}{\equiv} \begin{cases} 1, & \text{for } j = 0, \\ 0, & \text{for } j = 1, \ldots, p^k - 1. \end{cases}$$

To see that V_m actually consists of all fully symmetric m-periodic functions $\{0,1\}^n \longrightarrow \mathbb{Z}_p$, first note that each such a function can be obtained as a linear combination (over $\mathbb{Z}_p$) of $w_0, w_1, \ldots, w_{m-1}$, where

$$w_j[i] = \begin{cases} 1, & \text{for } i \equiv j \mod p^k, \\ 0, & \text{else.} \end{cases}$$

This shows that the vector space of all fully symmetric m-periodic functions has dimension at most m so that it has to coincide with V_m.

This observation allows us to represent the companion function $w' : \{0,1\}^n \longrightarrow \mathbb{Z}_p$, that behaves as w in the statement of the Fact, as a linear combination of the the v'_j's. This can be used to represent w itself as the very same linear combination of the v_j's (with $j = 0, 1, \ldots, m-1$) showing that the degree of w can be kept below p^k. □

REFERENCES

[1] David A. Mix Barrington, Richard Beigel and Steven Rudich. 1994. Representing boolean functions as polynomials modulo composite numbers. *Computational Complexity*, 4, 4 (Dec. 1994), 367–382. DOI: https://doi.org/10.1007/BF01263424

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

[3] Richard Beigel and Jun Tarui. 1991. On ACC. *In Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '91)*, October 1-4, 1991, San Juan, Puerto Rico, IEEE, 783—792. https://doi.org/10.1109/SFCS.1991.185449.

[4] Stanley Burris and John Lawrence. 2005. Results on the equivalence problem for finite groups. *Algebra Universalis*, 52 (Feb. 2005), 495–500. DOI: https://doi.org/10.1007/s00012-004-1895-8

[5] Chris Calabro, Russell Impagliazzo and Ramamohan Paturi. 2009. The Complexity of Satisfiability of Small Depth Circuits. *In Proceedings of the 4th International Workshop on Parameterized and Exact Computation (IWPEC'09)*, September 10-11, 2009, Copenhagen, Denmark, Lecture Notes in Computer Science, 5917, Springer, Berlin, Heidelberg, 75—85. DOI: https://doi.org/10.1007/978-3-642-11269-0_6

[6] Hervé Caussinus. 1996. A Note on a Theorem of Barrington, Straubing and Thérien. *Information Processing Letters*, 58, 1 (Apr. 1996), 31–33. DOI: https://doi.org/10.1016/0020-0190(96)00029-4

[7] Vince Grolmusz and Gábor Tardos. 2000. Lower bounds for (MODp-MODm) circuits. *SIAM Journal of Computing*, 29, 4 (2000), 1209–1222. DOI: https://doi.org/10.1137/S0097539798340850

[8] Vince Grolmusz. 2001. A Degree-Decreasing Lemma for (MOD-q - MOD-p) Circuits. *Discrete Mathematics and Theoretical Computer Science*, 4, 2 (Jan. 2001), 247–254. DOI: https://doi.org/10.46298/dmtcs.289

[9] Brynmor Chapman and R. Ryan Williams. 2022. Smaller ACC0 Circuits for Symmetric Functions. *In Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS 2022)* January 11-13, 2023, Cambridge, Massachusetts, LIPIcs, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Article 38, 19 pages. https://doi.org/10.4230/LIPIcs.ITCS.2022.38.

[10] Arkadev Chattopadhyay, Navin Goyal, Pavel Pudlak, Denis Therien. 2006. Lower bounds for circuits with MOD_m gates. *In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, Oktober 21-24, 2006 Berkeley, California, IEEE, 709—718. https://doi.org/10.1109/FOCS.2006.46

[11] Attila Földvári and Gábor Horváth. 2020. The complexity of the equation solvability and equivalence problems over finite groups. *International Journal of Algebra and Computation*, 30, 3 (May 2020), 1–17. DOI: https://doi.org/10.1142/S0218196720500137

[12] Mikael Goldmann and Alexander Russell. 1999. The complexity of solving equations over finite groups. *In Proceedings of the 14th Annual IEEE Conference on Computational Complexity (CCC'99)*, May 4-6, 1999, Atlanta, Georgia, IEEE, 80–86. https://doi.org/10.1109/CCC.1999.766266

[13] Kristoffer Arnsfelt Hansen. 2008. Constant width planar branching programs characterize ACC^0 in quasipolynomial size. *In Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC'08)*, June 23-26, 2008, College Park, Maryland, IEEE, 92—99. https://doi.org/10.1109/CCC.2008.30

[14] Kristoffer Arnsfelt Hansen and Michal Koucký. 2009. A New Characterization of ACC^0 and Probabilistic CC^0. *In Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC'09)*, July 15-18, 2009, Paris, France, IEEE, 27–34. https://doi.org/10.1109/CCC.2009.15

[15] Gábor Horváth. 2011. The complexity of the equivalence and equation solvability problems over nilpotent rings and groups. *Algebra Universalis*, 66 (Dec. 2011), 391—403. DOI: https://doi.org/10.1007/s00012-011-0163-y

[16] Gábor Horváth. 2015. The complexity of the equivalence and equation solvability problems over meta-Abelian groups. *Journal of Algebra* 433 (Jul. 2015), 208–230. DOI: https://doi.org/10.1016/j.jalgebra.2015.03.015

[17] Gábor Horváth and Csaba Szabó. 2006. The complexity of checking identities over finite groups. *Internat. J. Algebra Comput.*, 16, 5 (Oct. 2006), 931-939. DOI: https://doi.org/10.1142/S0218196706003256

[18] Paweł M. Idziak, Piotr Kawałek and Jacek Krzaczkowski. 2018. Expressive power, satisfiability and equivalence of circuits over nilpotent algebras. *In Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS'18)*, August 27–31, 2018, Liverpool, UK, LIPIcs, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Article 17, 15 pages. https://doi.org/10.4230/LIPIcs.MFCS.2018.17

[19] Paweł M. Idziak, Piotr Kawałek and Jacek Krzaczkowski. 2020. Intermediate Problems in Modular Circuits Satisfiability. *In Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'20)*, July 8–11, 2020, Saarbrücken, Germany, ACM, 578-–590. https://doi.org/10.1145/3373718.3394780

[20] Paweł M. Idziak, Piotr Kawałek, Jacek Krzaczkowski and Armin Weiß. 2020. Equation satisfiability in solvable groups. To appear in *Theory of Computing Systems*. arXiv:2010.11788. Retrieved from https://arxiv.org/abs/2010.11788.

[21] Paweł M. Idziak, Piotr Kawałek, Jacek Krzaczkowski and Armin Weiß. 2022. Satisfiability problems for finite groups. *In Proceedings of the 49th International Colloquium on Automata, Languages and Programming (ICALP'22)*, July 4-8, 2022, Paris, France, LIPIcs, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Article 133, 20 pages. https://doi.org/10.4230/LIPIcs.ICALP.2022.133

[22] Paweł M. Idziak and Jacek Krzaczkowski. 2022. Satisfiability in MultiValued Circuits. *SIAM Journal on Computing*, 51, 3 (2022), 337-378. DOI: https://doi.org/10.1137/18M1220194 Conference version: *In Proceedings of 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'18)*, July 9–12, 2018, Oxford, UK, ACM, 550–558. https://doi.org/10.1145/3209108.3209173.

[23] Russell Impagliazzoa, Ramamohan Paturia and Francis Zaneb. 2001. Which problems have strongly exponential complexity, *J. Comput. Syst. Sci.* 63, 4 (Dec. 2001), 512–530. DOI: https://doi.org/10.1006/jcss.2001.1774

[24] Piotr Kawałek, Michael Kompatscher and Jacek Krzaczkowski. 2019. Circuit equivalence in 2-nilpotent algebras. arXiv:1909.12256. Retrieved from https://arxiv.org/abs/1909.12256.

[25] Michael Kompatscher. 2021. CSAT and CEQV for nilpotent Maltsev algebras of Fitting length > 2. arXiv:2105.00689. Retrieved from https://arxiv.org/abs/2105.00689.

[26] Howard Straubing and Denis Thérien. 2006. A note on MODp–MODm circuits. *Theory of Computing Systems*, 39 (Sep. 2006), 699–706. DOI: https://doi.org/10.1007/s00224-004-1210-2

[27] A. Weiß. 2020 Hardness of Equations over Finite Solvable Groups Under the Exponential Time Hypothesis. *In Proceedings of the 47th International Colloquium on Automata, Languages and Programming (ICALP'20)*, July 8-11, 2020, Saarbrücken, Germany, LIPIcs, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Article 102, 19 pages. https://doi.org/10.4230/LIPIcs.ICALP.2020.102