

NONUNIFORM DETERMINISTIC FINITE AUTOMATA OVER FINITE ALGEBRAIC STRUCTURES

PAWEŁ M. IDZIAK

*Department of Theoretical Computer Science,
Jagiellonian University,
Kraków, Poland*

PIOTR KAWALEK

*Institute of Discrete Mathematics and Geometry,
TU Wien, Austria*

*Department of Theoretical Computer Science,
Jagiellonian University,
Kraków, Poland*

JACEK KRZACZKOWSKI

*Department of Computer Science,
Maria Curie-Skłodowska University,
Lublin, Poland*

ABSTRACT. Nonuniform deterministic finite automata (NUDFA) over monoids were invented by Barrington in [Bar85] to study boundaries of nonuniform constant-memory computation. Later, results on these automata helped to identify interesting classes of groups for which equation satisfiability problem (POLSAT) is solvable in (probabilistic) polynomial-time [GR02, IKKW22b]. Based on these results, we present a full characterization of groups, for which the identity checking problem (called POLEQV) has a probabilistic polynomial-time algorithm. We also go beyond groups, and propose how to generalise the notion of NUDFA to arbitrary finite algebraic structures. We study satisfiability of these automata in this more general setting. As a consequence, we present full description of finite algebras from congruence modular varieties for which testing circuit equivalence CEQV can be solved by a probabilistic polynomial-time procedure. In our proofs we use two computational complexity assumptions: randomized Exponential Time Hypothesis and Constant Degree Hypothesis.

E-mail addresses: pawel.idziak@uj.edu.pl, piotr.kawalek@tuwien.ac.at, krzacz@umcs.pl.

Piotr Kawalek: This research was funded in whole or in part by National Science Centre, Poland #2021/41/N/ST6/03907. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission. Funded by the European Union (ERC, POCOCOP, 101071674). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Jacek Krzaczkowski: This research was funded in whole or in part by National Science Centre, Poland #2022/45/B/ST6/02229. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

1. INTRODUCTION

There are many interactions between mathematics and (theoretical) computer science. Many branches of these sciences influence each other, sometimes in quite surprising ways. Relatively recent example of such an influence is so-called algebraic approach to Constraint Satisfaction Problem (CSP) which led to complete classification of computational complexity of CSP [Bul17, Zhu20]. It is really impressive how in this case (universal) algebra, combinatorics, logic, computational complexity and algorithmic work together to give new results in each of these fields.

Another example of synergy between different fields of mathematics and theoretical computer science can be observed on the borderline of circuit complexity, automata theory and (universal) algebra. The most significant example here is the role of monoids played in automata theory and formal languages.

Usually deterministic finite automaton (DFA) is determined by an alphabet Σ acting over a set Q of states by a function $\delta : \Sigma \times Q \ni (\sigma, q) \mapsto \sigma \cdot q \in Q$. This action can be extended (in an obvious way) to the action of the free monoid Σ^* . To decide if a word $w \in \Sigma^*$ is accepted by a particular DFA we need to endow it with a starting state q_0 and a set $F \subseteq Q$ of accepting states. Then w gets accepted if $w \cdot q_0 \in F$. For our purposes we prefer, first to treat the set Q^Q of functions as the monoid with $f \cdot g = g \circ f$, and then to treat the action δ as a function $a : \Sigma \rightarrow Q^Q$ given by $a(\sigma)(q) = \sigma \cdot q$. Now, the word $\sigma^1 \dots \sigma^n$ gets accepted if $a(\sigma^1) \dots a(\sigma^n) \in S$, where S consists of transitions determined by the words $w \in \Sigma^n$ satisfying $w \cdot q_0 \in F$.

Before restating Barrington's definition of Non-uniform Deterministic Finite Automata (NUDFA) over monoids [Bar86] we note that $a(\sigma^1) \dots a(\sigma^n)$ is nothing else but $\mathbf{t}_n(a(\sigma^1), \dots, a(\sigma^n))$, where $\mathbf{t}_n(x_1, \dots, x_n) = x_1 \cdot \dots \cdot x_n$. In NUDFA over the monoid \mathbf{M} to accept the word from Σ^n we are going to relax the term $x_1 \cdot \dots \cdot x_n$ to an arbitrary semigroup term $\mathbf{t}(x_1, \dots, x_k)$ and replace one action $a : \Sigma \rightarrow Q^Q$ by a bunch of functions $a^x : \Sigma \rightarrow M$, one for each variable of \mathbf{t} . Now a \mathbf{t} -program (with inputs from Σ^n represented by n -variable word $b_1 \dots b_n$) consists of:

- a set of k -instructions, one for each variable x of \mathbf{t} , of the form $\iota(x) = (b^x, a^x)$, where b^x is one of the variables b_i ,
- and a set $S \subseteq M$ of accepting values.

Finally, a NUDFA over \mathbf{M} is a sequence (possibly even nonrecursive) of programs $(\mathbf{t}_n, n, \iota_n, S_n)_{n \in \mathbb{N}}$ with $\mathbf{t}_n = (x_1, \dots, x_{k_n})$ being some terms of \mathbf{M} , $S_n \subseteq M$ and ι_n being the instructions for the variables of \mathbf{t}_n . A word $b^1 \dots b^n \in \Sigma^n$ gets accepted by such a NUDFA if $\mathbf{t}_n(a^{x_1}(b^{x_1}), \dots, a^{x_{k_n}}(b^{x_{k_n}})) \in S_n$. Originally Barrington considered mainly the Boolean case where $\Sigma = \{0, 1\}$ to study computational boundaries of non-uniform constant-memory computation. Such automata can be used to compute the Boolean functions of the form $\{0, 1\}^n \rightarrow \{0, 1\}$. They also give an interesting algebraic insight into the internal structure of the class NC^1 [BST90]. Note that in the Boolean case $\Sigma = \{0, 1\}$ the function a^x is given by the pair of values $a^x(0), a^x(1)$. In such the case, for simplicity we write $\iota(x) = (\mathbf{b}^x, a^x(0), a^x(1))$.

A natural question that arises here is whether the language accepted by a particular NUDFA over \mathbf{M} is nonempty. This problem reduces to:

PROGSAT(\mathbf{M}): Decide if a given program over \mathbf{M} accepts at least one word.

The problem PROGSAT proved itself to be extremely useful in studying groups (and monoids) for which determining if an equation has a solution (POLSAT) is in P. In fact the proof of Goldmann and Russell in [GR02] that each nilpotent group \mathbf{G} has tractable POLSAT(\mathbf{G}) modifies a polynomial time algorithm for PROGSAT(\mathbf{G}). A study of connections between POLSAT and PROGSAT for finite monoids is given in [BMM⁺00]. Recently a complete classification of finite groups \mathbf{G} with PROGSAT(\mathbf{G}) \in RP, together with its consequences for POLSAT has been provided by Idziak, Kawałek and Krzaczkowski in [IKKW22b].

Now the results of [IKKW22b] allows us to complete the long term extensive investigations [BL05, HS11, FH20, Wei20, IKKW22a] on equivalence problem POLEQV of polynomials (i.e. terms with some variables already evaluated) over finite groups.

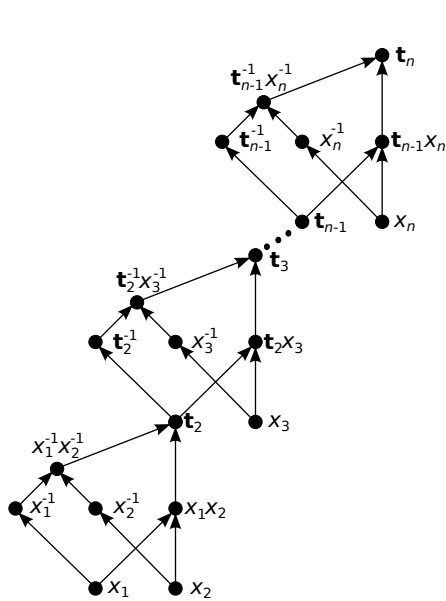
The lower bound in our characterization relies on the randomized version of the Exponential Time Hypothesis (rETH), while the upper bounds explore the so-called Constant Degree Hypothesis (CDH). This hypothesis, introduced in [BST90], can be rephrased to state that for a fixed integer d , a prime p and an integer m which is not just a power of p , any 3-level circuits of the form $\text{AND}_d \circ \text{MOD}_m \circ \text{MOD}_p$ require exponential size to compute AND_n of arbitrary large arity n (in which AND_d gates are in the input layer, MOD_m gates are in the middle layer, and there is one MOD_p gate in the output layer). The best known lower bound, due to Chattopadhyay et al. [CGPT06], for the size of $\text{AND}_d \circ \text{MOD}_m \circ \text{MOD}_p$ computing AND_n is only superlinear. Much earlier CDH has been considered in many different contexts. Already in [BST90] the case $d = 1$, i.e. no AND_d layer, has been confirmed. Also restrictions put either on the number of AND_d used locally, or on the local structure of $\text{AND}_d \circ \text{MOD}_m$ fragments, allowed Grolmusz and Tardos [GT00, Gro01] to confirm CDH. Very recently Kawałek and Weiß [KW23] confirmed CDH for symmetric circuits.

In this paper, under these two complexity hypothesis (i.e. rETH and CDH), the characterization of finite groups with tractable PROGSAT from [IKKW22b] is applied to get unexpectedly different characterization for tractable POLEQV.

Theorem 1.1. *Let \mathbf{G} be a finite group. Assuming rETH and CDH the problem $\text{POLEQV}(\mathbf{G})$ is in RP if and only if \mathbf{G} is solvable and has a nilpotent normal subgroup \mathbf{H} with the quotient \mathbf{G}/\mathbf{H} being also nilpotent.*

A surprising part of these investigations is that such characterization can not only be done, but that it can be stated, in terms of algebraic structure of the groups. In fact this reveals another connection between (universal) algebra and circuit complexity theory.

The goal of this paper is to leave the group realm and generalize Theorem 1.1 to a much broader setting of algebras. As we will see shortly this generalization is two-fold. First we generalize the concept of NUDFas to cover automata working over arbitrary finite algebraic structure \mathbf{A} . This requires to define a program over \mathbf{A} and can be done by simply replacing the monoid \mathbf{M} by the algebra \mathbf{A} and assume that this time \mathbf{t} is a term of the algebra \mathbf{A} .



Compressing the size of \mathbf{t}_n .
©Idziak, Krzaczkowski [IK22]

Second, in general algebraic context it is not clear which operations are to be chosen to be the basic ones. And this choice may be extremely important from computational point of view. Recall after [IK22], that adding the binary commutator operation $[x, y] = x^{-1}y^{-1}xy$ to the language of a group may exponentially shorten the size of the input. Indeed the term $\mathbf{t}_n(x_1, \dots, x_n) = [\dots [[x_1, x_2], x_3] \dots, x_n]$ has linear size if the commutator operation is allowed, while after writing this term in the pure group language (of multiplication and the inverse) we see that the size $|\mathbf{t}_n|$ of \mathbf{t}_n is $2|\mathbf{t}_{n-1}| + 2$, as $\mathbf{t}_n(x_1, \dots, x_n) = \mathbf{t}_{n-1}(x_1, \dots, x_{n-1})^{-1} \cdot x_n^{-1} \cdot \mathbf{t}_{n-1}(x_1, \dots, x_{n-1}) \cdot x_n$, so that $|\mathbf{t}_n|$ is exponential on n . Actually for the alternating group \mathbf{A}_4 it is shown in [HS12] that $\text{POLSAT}(\mathbf{A}_4)$ is in P, while after endowing \mathbf{A}_4 with the commutator operation (and therefore shortening the size of inputs) the problem becomes NP-complete. A solution to this phenomena has been proposed by Idziak and Krzaczkowski in [IK22] by presenting a term by the algebraic circuit that computes it. For example $\mathbf{t}_n(x_1, \dots, x_n)$ can be computed by the circuit

of size $6n - 5$ as shown by Figure 1.

Representing polynomials of an algebra with algebraic circuits leads to a modified version of PROGSAT which we explore in this paper.

PROGCSAT(\mathbf{A}): Decide if a given program $(\mathbf{t}, n, \iota, S)$ over \mathbf{A} accepts at least one word, where \mathbf{t} is given by a circuit over \mathbf{A} .

Measuring the size of the input, i.e. the expression of the form $\mathbf{p} = \mathbf{q}$ by the lengths of the polynomials \mathbf{p} and \mathbf{q} or by the sizes of their algebraic circuits used to compute \mathbf{p} and \mathbf{q} give rise to either POLSAT or CSAT in the satisfiability setting or to POLEQV or CEQV in the equivalence setting. Note here that [IK22] argues that the complexity of CSAT(\mathbf{A}) and CEQV(\mathbf{A}) is independent of which term operations of the algebra \mathbf{A} are chosen to be the basic ones. This independence gives a hope for a characterization of algebras \mathbf{A} with tractable CSAT(\mathbf{A}) or CEQV(\mathbf{A}) in terms of algebraic structure of \mathbf{A} . Actually already a series of papers [IKK20, Wei20, IKK22b, IKKW22b, IK22, Kom23] enforces a bunch of such necessary algebraic conditions for an algebra to have CSAT or CEQV tractable. Not surprisingly solvability and nilpotency are among these conditions.

To define these two notions of solvability and nilpotency outside the group realm we need a notion of a commutator. However we need to work with a commutator $[\alpha, \beta]$ of two congruences α, β (that in the group setting correspond to normal subgroups) instead of a commutator of elements of an algebra. For more details on the definition and the properties of commutator, we refer to the book [FM87] and Section 2. Here we only note that this concepts of commutator of congruences works smoothly only in some restricted setting of the so called congruence modular varieties, i.e. equationally definable classes of algebras with modular congruence lattices. Fortunately this setting includes groups, rings, quasigroups, loops, Boolean algebras, Heyting algebras, lattices and almost all algebras related to logic. In groups, rings or Boolean/Heyting algebras the congruences are determined by normal subgroups, ideals or filters respectively. Obviously this new concept of commutator of normal subgroups coincides with the old one. The commutator of two ideals I, J of a commutative ring is simply their algebraic product $I \cdot J$, while the commutator of filters in a Boolean/Heyting algebra is their intersection. Now we can say that a congruence α is abelian, nilpotent or solvable if $[\alpha, \alpha] = 0_{\mathbf{A}}$, $[\dots [[\alpha, \alpha], \alpha], \dots \alpha] = 0_{\mathbf{A}}$ or $[[[\alpha, \alpha], [\alpha, \alpha]], \dots [[\alpha, \alpha], [\alpha, \alpha]]] = 0_{\mathbf{A}}$ respectively (for some number of nested commutators). Here, $0_{\mathbf{A}}$ is the identity relation/congruence of \mathbf{A} . The algebra \mathbf{A} itself is said to be abelian, nilpotent or solvable if the total congruence $1_{\mathbf{A}}$ collapsing everything is abelian, nilpotent or solvable, respectively.

Note that, for nilpotent groups, boolean programs (of NUDFAs) compute AND functions only of bounded arity, i.e. for each nilpotent group \mathbf{G} there is a constant k such that AND_k is computable by no program over \mathbf{G} [BST90]. This nonexpressibility phenomena does not transfer to nilpotent algebras in general congruence modular context. The most natural example here is the algebra $(\mathbb{Z}_6; +, \%2)$, i.e. the group $(\mathbb{Z}_6; +)$ endowed with the unary operation $\%2$ computing the parity. In this algebra all the circuits of the form $\text{MOD}_2 \circ \text{MOD}_3$ can be modelled so that AND_n can be expressed for all n (however by exponential size of the circuits). This action of the prime 2 acting over prime 3 cannot occur in nilpotent groups. Indeed, due to the Sylow theorem, each finite nilpotent group is a product of p -groups. This decomposition prevents interaction between different primes, as they occur on different stalks/coordinates. And the lack of such interactions is crucial in bounding the arity of expressible AND_n 's. Also in our considerations the finite nilpotent algebras that decompose into a product of algebras of prime power order occurs naturally. They are known as supernilpotent algebras [Bul00, AM10]. We will return to this concept of supernilpotent algebras and its relativization to supernilpotent congruences in Section 2.

Now we are ready to state the other two main results of the paper.

Theorem 1.2. *Let \mathbf{A} be a finite algebra from a congruence modular variety. Assuming $rETH$ and CDH the problem $\text{PROGCSAT}(\mathbf{A})$ is in RP if and only if \mathbf{A} is nilpotent and has a supernilpotent congruence σ with supernilpotent quotient \mathbf{A}/σ and such that all cosets of σ have sizes that are powers of the same prime number p .*

Theorem 1.3. *Let \mathbf{A} be a finite algebra from a congruence modular variety. Assuming $rETH$ and CDH the problem $CEQV(\mathbf{A})$ is in RP if and only if \mathbf{A} is nilpotent and has a supernilpotent congruence σ with supernilpotent quotient \mathbf{A}/σ .*

Results from [IKKW22b] that we use in the proof of Theorem 1.1 heavily rely on a method of representing terms/polynomials of finite solvable group \mathbf{G} by bounded-depth circuits that use only modular gates. Besides the obvious requirement that the circuit representing a group-polynomial \mathbf{p} has to compute the very same function as \mathbf{p} does, we also want to control the size of the circuit to be polynomial in terms of the size (length) of \mathbf{p} .

A very similar approach can be found in [Kom22], where M. Kompatscher considers generalizations of finite nilpotent groups, i.e. nilpotent algebras from the congruence modular varieties. He provides a method to rewrite circuits over such algebras to constant-depth circuits which, again, use only modulo-counting gates. These modular circuits, which appear in both of the mentioned cases, are known as CC-circuits. However, since [Kom22] does not use the notion of a program/NUDFA, the author formulates his results for circuits over \mathbf{A} representing only some specific functions. For those functions, it is natural how to interpret Boolean values 0/1 in the non-Boolean algebra \mathbf{A} . In our paper, the notion of a program/NUDFa provides us with a formal framework which helps to relate the expressive power of algebraic structures to some standard circuit complexity classes. Here, we present a very precise characterization of functions computable by programs over algebras corresponding to polynomial-time cases of PROGCSAT.

Theorem 1.4. *Let \mathbf{A} be a finite nilpotent algebra from a congruence modular variety with supernilpotent congruence σ of \mathbf{A} such that cosets of α are of prime power size p^k and \mathbf{A}/σ is supernilpotent. Then the function computable by a boolean program of size ℓ over the algebra \mathbf{A} can be also computed by an $AND_d \circ MOD_m \circ MOD_p$ -circuits of size $O(\ell^c)$ with d, m, c being natural numbers depending only on \mathbf{A} , and m being relatively prime to p .*

This theorem not only is an interesting result on its own, but also is crucial in proving Theorem 1.2 and 1.3.

2. ALGEBRAIC PRELIMINARIES

An algebra is a set called universe together with a finite set of operations acting on it called basic operations of the algebra. We usually use boldface letter to denote the algebra and the very same latter, but with a regular font, to denote its universe. $\text{Pol } \mathbf{A}$ is polynomial clone of \mathbf{A} , that is the set of all polynomial operations of \mathbf{A} . An algebra \mathbf{A} is polynomially equivalent to an algebra \mathbf{B} if it is isomorphic to an algebra which has the same set of polynomial operations as \mathbf{B} . An induced algebra $\mathbf{A}|_S$ is a set S with all polynomial operations of \mathbf{A} closed on S (or in other word polynomial operations which for arguments from S have value in S). Idempotent function is a function f such that $f(f(x)) = f(x)$.

In proves of intractability of PROGCSAT and CEQV for algebras from congruence modular varieties the crucial role is played by Tame Congruence Theory (see [HM88] for details). This is a deep algebraic tool describing local behavior of finite algebras. TCT shows that locally finite algebras behave in one of following five ways:

1. a finite set with a group action on it,
2. a finite vector space over a finite field,
3. a two-element Boolean algebra,
4. a two-element lattice,
5. a two-element semilattice.

By $\text{typ}\{\mathbf{A}\}$, let us denote a subset of $\{1, \dots, 5\}$ which describes the local behaviours we can find in an algebra \mathbf{A} . Note that in a case of algebra \mathbf{A} from a congruence modular variety only three types can appear in $\text{typ}\{\mathbf{A}\}$, that is types 2, 3 and 4. In case of types 3 and 4 there has to be two-element set U (let's call element of U as 0 and 1) such that

- there is a polynomial \mathbf{e} of \mathbf{A} fulfilling $\mathbf{e}(A) = U$,
- there are polynomials of \mathbf{A} which behaves on U like \wedge and \vee ,
- in case of type 3 there is also unary polynomial of \mathbf{A} which is a negation on U .

If we can find type **3** or **4** in $\text{typ}\{\mathbf{A}\}$, the complexity of both PROGCSAT and CEQV is relatively easy to determine, as we shall see in forthcoming chapters. For this reason the most of the volume of the paper is devoted to algebras with $\text{typ}\{\mathbf{A}\} = \{\mathbf{2}\}$. In the congruence modular variety those are precisely the solvable algebras [FM87, HM88]. In fact, some of the earlier papers already dealt with algebras that are solvable but not nilpotent, so we will be mostly concerned with the notion of nilpotency and its properties.

Every solvable (so in particular - nilpotent) algebra in the congruence modular variety is Malcev, i.e. it possesses a polynomial operation \mathbf{d} satisfying the following identity: $\mathbf{d}(y, x, x) = \mathbf{d}(x, x, y) = y$. A standard example of Malcev algebras are groups with a Malcev term of the form $x \cdot y^{-1} \cdot z$. Unlike for groups, nilpotent algebras do not necessarily decompose into a direct product of algebras of prime power order. However, the technique contained in this paper splits an algebra into slices on which such nice decomposition can be observed.

To define this slicing properly, we need to consider congruences. Recall that congruence σ of an algebra \mathbf{A} is an equivalence relation which is preserved by the operations of \mathbf{A} . Such relations are naturally associated with surjective homomorphisms from \mathbf{A} to \mathbf{A}/σ mapping x to $[x]_\sigma$ (equivalence class of x in σ), so congruences are essentially generalization of normal subgroups of a group. Similarly as normal subgroups, congruences of an algebra form a lattice. From now on, we write $\text{Con } \mathbf{A}$ for the set of all congruences of \mathbf{A} . Every element of a finite lattice can be written as a meet (join) of meet-irreducible (join-irreducible) elements, i.e. elements which cannot be written as a meet (join) of any other two elements of the lattice. These special elements, generating $\text{Con } \mathbf{A}$, will play a significant role in our analysis.

For $\alpha, \beta \in \text{Con } \mathbf{A}$, by $I[\alpha, \beta]$ we mean a set of congruences γ such that $\alpha \leq \gamma \leq \beta$. In case when $I[\alpha, \beta] = \{\alpha, \beta\}$, i.e. there are no congruences between α and β , we call β a cover of α , we call α a subcover of β , and we call α, β a covering pair. To highlight such a situation we write $\alpha < \beta$ for short. Whenever α is meet-irreducible (join-irreducible) congruence, then there is a unique congruence α^+ (α^-) such that $\alpha < \alpha^+$ ($\alpha^- < \alpha$). For a nilpotent algebra \mathbf{A} from CM whenever its congruences α, β satisfy $\alpha < \beta$, the cosets (congruence classes) of β/α in \mathbf{A}/α have equal sizes, being a power of some prime (see Lemma 7.1 and Lemma 7.2). We later denote this prime by $\text{char}(\alpha, \beta)$ and call it a characteristic of a congruence cover $\alpha < \beta$. Moreover for arbitrary $\alpha < \beta$, we write $\text{char}\{\alpha, \beta\}$ for the set of all possible prime characteristics of covering pairs, which are fully contained in $I[\alpha, \beta]$. We say that a pair of congruences $\alpha < \beta$ of a nilpotent algebra \mathbf{A} forms a Prime Uniform Product Interval (PUPI), if there are congruences $\alpha < \alpha_1, \dots, \alpha_k \leq \beta$ such that

- $\bigvee \alpha_i = \beta$
- $\alpha_i \wedge (\bigvee_{j \neq i} \alpha_j) = \alpha$, for $i, j \in \{1..k\}$,
- For every i we have $|\text{char}\{\alpha, \alpha_i\}| = 1$.

We call a congruence β supernilpotent whenever it is nilpotent and the interval $I[0_{\mathbf{A}}, \beta]$ is a PUPI, and we call an algebra \mathbf{A} supernilpotent, whenever $1_{\mathbf{A}}$ is supernilpotent. Each supernilpotent algebra is isomorphic to a direct product of nilpotent algebras of prime power size. In this sense supernilpotence generalizes nilpotence for groups. Note that this definition is equivalent to a standard definition of supernilpotent algebras in congruence modular varieties [MS21].

We say that a nilpotent algebra \mathbf{A} has supernilpotent rank k whenever k is the smallest number for which we can find sequence of congruences $0_{\mathbf{A}} = \alpha_0 < \alpha_1 < \dots < \alpha_k = 1_{\mathbf{A}}$ such that interval $I[\alpha_i, \alpha_{i+1}]$ is a PUPI for each $0 \leq i < k$. Note that for a finite nilpotent algebra \mathbf{A} we can always find such a finite sequence, since each covering pair forms a PUPI. This notion of supernilpotent rank of an algebra, later denoted by $\text{sr}(\mathbf{A})$, proved to be extremely useful in some very recent results on the computation complexity of circuit satisfiability problem [IKK20, Kom23]. In fact, results for PROGCSAT / CEQV we present in this paper, are essentially about nilpotent algebras with $\text{sr}(\mathbf{A}) = 2$.

3. POLYNOMIAL EQUIVALENCE

Our characterization of polynomial time cases of POLEQV is achieved through a reduction to some special instances of POLSAT. It was noticed already in [GR02] that $\text{POLSAT}(\mathbf{G})$ for finite group \mathbf{G} reduces in polynomial time to $\text{PROGSAT}(\mathbf{G})$. Very recently, the full characterization, of groups for which PROGSAT can be solved in randomized polynomial time was shown under the assumptions of rETH and CDH in [IKKW22b].

Theorem 3.1. *Let \mathbf{G} be a finite group. Assuming rETH and CDH the problem $\text{PROGSAT}(\mathbf{G})$ is in RP if and only if \mathbf{G}/\mathbf{G}_p is nilpotent for some normal p -subgroup \mathbf{G}_p of \mathbf{G} (with p being prime).*

Theorem 3.1 together with a result from [IKKW22a] provides us with enough information to prove the Theorem 1.1.

Proof. **TOPROVE 0** □

4. PROGRAM SATISFIABILITY

The goal of this section is to prove Theorem 1.2. First we observe in Fact 4.1 that if a nilpotent algebra has a Malcev term then CSAT and CEQV for this algebra reduces to PROGCSAT. Thus, intractability of CSAT or CEQV implies intractability of PROGCSAT and conversely if PROGCSAT is in RP, so are CSAT and CEQV.

Fact 4.1. *For a finite nilpotent Malcev algebra \mathbf{A} the problems $\text{CSAT}(\mathbf{A})$ and $\text{CEQV}(\mathbf{A})$ are Turing reducible to $\text{PROGCSAT}(\mathbf{A})$.*

Proof. **TOPROVE 1** □

In the proof of Theorem 1.2 we will use Fact 4.1 to show that, under our assumptions, nilpotent Malcev algebra with tractable PROGCSAT has supernilpotent rank equal at most 2. In section 8 we use advanced tools of Tame Congruence Theory and Commutator Theory to prove the following lemma which shows that not for every algebra with supernilpotent rank equal 2 PROGCSAT is tractable.

Lemma 4.2. *Let \mathbf{A} be a finite nilpotent algebra from a congruence modular variety with $\text{sr}(\mathbf{A}) = 2$ and tractable $\text{PROGCSAT}(\mathbf{A})$.*

Then \mathbf{A} has a supernilpotent congruence α with cosets of prime power order such that quotient algebra \mathbf{A}/α is also supernilpotent, or rETH fails.

The important ingredient of the proof of above lemma is an idea of Barrington et al [BBR94] heavily explored in [IKK22a] and [IKKW22b] resulting in the following lemma.

Lemma 4.3. *Let p be a prime number and $\nu \geq 1$ be an integer. Then for each 3-CNF formula $\Phi(\bar{x})$ with n variables there is a polynomial $w_p^\Phi(\bar{x})$ over $GF(p)$ of degree at most $O(p^\nu)$ such that for all $\bar{b} \in \{0, 1\}^n$ we have*

$$w_p^\Phi(\bar{b}) = \begin{cases} 0, & \text{if the number of unsatisfied (by } \bar{b} \text{) clauses in } \Phi \\ & \text{is divisible by } p^\nu \\ 1, & \text{otherwise.} \end{cases}$$

Moreover, computing w_p^Φ from Φ can be done in $2^{O(p^\nu(\log n + \log p))}$ steps.

The power of Lemma 4.3 can be observed when we use it simultaneously for two different primes, say p_1 and p_2 . Then if for a given 3-CNF formula Φ with m clauses we will choose positive integers ν_1, ν_2 such that $p_i^{\nu_i-1} \leq \sqrt{m} < p_i^{\nu_i}$, we will get, by Chinese Remainder Theorem, that Φ is satisfied by $\bar{b} \in \{0, 1\}^n$ iff $w_{p_1}^\Phi(\bar{b}) = w_{p_2}^\Phi(\bar{b}) = 0$. Moreover, the lengths of $w_{p_1}^\Phi(\bar{b})$ and $w_{p_2}^\Phi(\bar{b})$ are subexponential in the size of Φ . The core of the proof of Lemma 4.2 is showing (with heavily use of Tame Congruence Theory and Commutator Theory) that if nilpotent Malcev algebra \mathbf{A} with supernilpotent rank 2 has supernilpotent congruence α which cosets are not of

prime power size and such that \mathbf{A}/α is supernilpotent then we can simulate by programs over \mathbf{A} systems of equations in the form:

$$\begin{aligned} w_{p_1}^\Phi(\bar{b}) &= 0, \\ w_{p_2}^\Phi(\bar{b}) &= 0. \end{aligned}$$

Now we are ready to prove Theorem 3.1

Proof. **TOPROVE 2** □

5. CIRCUIT EQUIVALENCE

In this section we will prove Theorem 1.3. To do it we will show that solving CEQV for an algebra \mathbf{A} can be reduced to solving the very same problem for quotients of \mathbf{A} by a meet-irreducible congruences. Obviously, every quotient algebra by a meet-irreducible congruence has the smallest congruence bigger than identity relation. This observation plays a crucial role in the proof of Theorem 1.3.

Proof. **TOPROVE 3** □

6. NOTATION

In this section we introduce detailed notation, needed in further sections of the paper.

Algebra. We use the standard universal algebraic notation the reader can find e.g. in [BS11]. Our results heavily rely on Tame Congruence Theory and Modular Commutator Theory which have a detailed description in [HM88] and [FM87] respectively. One can find the not too long summary of needed notions and facts in [IK22, Section 2]. Here we just recall, for readers convenience, the most important notation.

For an algebra \mathbf{A} and $a, b \in A$ congruence $\Theta(a, b)$ is the smallest congruence containing the pair (a, b) . Note that every join-irreducible congruence is generated by a single pair of elements of the algebra. In particular, covers of the identity relation, called atoms, are generated by a single pair.

Minimal sets are the central notion of the Tame Congruence Theory. Formally, for an algebra \mathbf{A} and $\alpha, \beta \in \text{Con } \mathbf{A}$ such that $\alpha < \beta$ we define the (α, β) -minimal set $U \subseteq A$ as minimal, with respect to the inclusion, among sets of the form $\mathbf{f}(A)$ for all unary polynomials \mathbf{f} of \mathbf{A} such that $\mathbf{f}(\beta) \not\subseteq \alpha$. In this paper we consider (α, β) -minimal sets for $\alpha \prec \beta$ only (i.e. for so-called prime quotients). For every minimal set U of \mathbf{A} there exists unary idempotent polynomial \mathbf{e}_U of \mathbf{A} such that $\mathbf{e}_U(A) = U$. A trace of the (α, β) -minimal set U is a set $u/\alpha \cap U$ for some $u \in U$ such that $u/\alpha \cap U \neq u/\beta \cap U$. For (α, β) -minimal set U (or its trace N) of algebra \mathbf{A} we usually consider an induced algebra $\mathbf{A}/\alpha|_{U/\alpha}$ ($\mathbf{A}/\alpha|_{N/\alpha}$). The five ways of local behaviour of finite algebras mentioned in Section 2 are exactly the types of algebras (or rather their polynomial clones) induced on minimal sets traces. Note that all traces of a minimal set are polynomially equivalent. In this paper we usually work with minimal sets of type **2**, that is in which algebras induced on traces are polynomially equivalent to one-dimensional vector spaces. Traces of a minimal set of type **2** are of prime power order, this prime is called a characteristic of the minimal set. Note that all minimal sets taken with respect to the same prime quotients induce polynomially equivalent algebras. It allows us to define the type of prime quotient as a type of minimal sets taken with respect to that quotient. In case of prime quotients of type **2** a prime characteristic of the quotient defined in Section 2 is equal to the characteristic of minimal sets taken with respect to it.

Algebras belonging to congruence modular variety are a wide class of algebras for which Tame Congruence Theory and Commutator Theory work particularly well. In this context term „modular” means that all algebras from the variety have modular congruence lattice i.e. for $\alpha, \beta, \gamma \in \text{Con } \mathbf{A}$ $\alpha \leq \beta$ implies $\alpha \vee (\gamma \wedge \beta) = (\alpha \vee \gamma) \wedge \beta$. Equivalently, lattice is modular if it has no elements α, β, γ for which $\alpha < \beta$, $\alpha \vee \gamma = \beta \vee \gamma$ and $\alpha \wedge \gamma = \beta \wedge \gamma$. Such a sublattice $\{\alpha, \beta, \gamma, \alpha \vee \gamma, \alpha \wedge \gamma\}$, if found in $\text{Con } \mathbf{A}$, is called a *pentagon*. If $I[\alpha, \beta]$ and $I[\gamma, \delta]$ are intervals such that $\beta \wedge \gamma = \alpha$ and $\beta \vee \gamma = \delta$, then $I[\alpha, \beta]$ is said to transpose up to $I[\gamma, \delta]$, written

$I[\alpha, \beta] \nearrow I[\gamma, \delta]$ and $I[\gamma, \delta]$ is said to transpose down to $I[\alpha, \beta]$, written $I[\gamma, \delta] \searrow I[\alpha, \beta]$ and the two intervals are called transposes of one another. Two intervals are said to be projective if one can be obtained from the other by a finite sequence of transposes. A fundamental fact in lattice theory is that a lattice is modular if and only if its projective intervals are isomorphic. In the case of congruence lattice of an algebra this isomorphism is even stronger since projective prime quotients have exactly the same minimal sets and in a consequence the same types and, in case of type **2**, the same characteristics.

Malcev algebras are examples of algebras from congruence modular varieties. Such algebras form so called congruence permutable varieties. Congruences of algebras from this class commute i.e. for arbitrary congruences α and β we have that $\alpha \circ \beta = \beta \circ \alpha$.

Supernilpotency plays a crucial role in our investigations. For a fixed algebra \mathbf{A} we write σ for a biggest supernilpotent congruence and κ for the smallest congruence such that \mathbf{A}/κ is supernilpotent. Moreover, we write σ_p for the biggest supernilpotent congruence with cosets of size being a power of p . The existence of such biggest/smallest congruences follows from term definition of supernilpotence and its properties [AM10, Corollary 6.6].

Arithmetic operations and functions. Sometimes we use addition in two different abelian groups inside one formula. To avoid ambiguity, different symbols for different types of addition are used: $+$, \sum for one type of addition (for instance in \mathbb{Z}_p) and \oplus , \bigoplus for the other (in \mathbb{Z}_m). Moreover, for a natural number m with a prime decomposition $p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \dots \cdot p_s^{\alpha_s}$ we write $\Delta(m) = \{p_1, \dots, p_s\}$ for the set of prime divisors of m and $\delta(m) = p_1 \cdot \dots \cdot p_s$ for the largest square-free divisor of m . We also use $\Delta(A)$ for $\Delta(|A|)$ and $\delta(A)$ for $\delta(|A|)$. Likewise, $\mu(A)$ denotes the maximal arity among basic operations of \mathbf{A} and $\mu(f)$ is the arity of the function f . For a set $X = \{d_1, \dots, d_s\}$ and a function $f : X^k \rightarrow Y$ we associate to f its binary expansion $f^\circ : \{0, 1\}^{|X| \cdot k} \rightarrow Y$, i.e., arbitrary function satisfying

$$f(x_1, \dots, x_k) = f^\circ(x_1 == d_1, \dots, x_1 == d_s, \dots, x_k == d_1, \dots, x_k == d_s).$$

Programs. Recall that a program $(\mathbf{p}, n, \iota, S)$ computes a Boolean function $\{0, 1\}^n \rightarrow \{0, 1\}$. From now on we write $(\mathbf{p})[\iota, S](\bar{b})$ for evaluation of this function on a tuple $\bar{b} \in \{0, 1\}^n$. Moreover, if $S = \{c\}$ is one-element set, we simply write $(\mathbf{p})[\iota, c](\bar{b})$ instead of $(\mathbf{p})[\iota, \{c\}](\bar{b})$. With each program $(\mathbf{p}, n, \iota, S)$ over \mathbf{A} we can naturally associate an inner function $(\mathbf{p})[\iota] : \{0, 1\}^n \rightarrow A$ satisfying $(\mathbf{p})[\iota](\bar{b}) \in S$ iff $(\mathbf{p})[\iota, S](\bar{b}) = 1$. This function is computed by the expression $\mathbf{p}(a^{x_1}(b^{x_1}), \dots, a^{x_k}(b^{x_k}))$, where all a^{x_i}, b^{x_i} are provided by the instructions $\iota(x) = (b^x, a^x)$. For a congruence $\sigma \in \text{Con } \mathbf{A}$ and a program $(\mathbf{p}, n, \iota, S)$ we can define the quotient program $(\mathbf{p}/\sigma, n, \iota/\sigma, S/\sigma)$ of an algebra \mathbf{A}/σ , by simply reinterpreting \mathbf{p} to be a circuit over \mathbf{A}/σ (algebras \mathbf{A} and \mathbf{A}/σ have the same signature), and taking $\iota(x) = (b^x, a^x/\sigma)$, and $S/\sigma = \{[s]_\sigma : s \in S\}$.

Circuits. In next chapters we use different types of gates to build bounded-depth circuits computing Boolean functions. For instance we write AND_d to denote a gate which takes at most d inputs and computes their conjunction and OR_d for a gate computing at most d -ary disjunction. A MOD_m -type boolean gate is any unbounded fan-in gate, which sums the inputs modulo m and returns 1 iff the sum belongs to some accepting set $S \subseteq \mathbb{Z}_m$. We allow different accepting sets for different gates. Other kinds of gates appearing in the paper are $\Sigma_{(p, \nu)}$ gates which also rely on modulo counting, but in a more complex way. Each such a gate takes n inputs b_1, \dots, b_n , and computes their affine combination $\alpha_1 b_1 + \dots + \alpha_n b_n + d$ in \mathbb{Z}_p^v . Here we interpret each b_i as v -dimensional vector (b_i, \dots, b_i) . We allow each α_i to be arbitrary endomorphism of the abelian group \mathbb{Z}_p^v , which can be also viewed as arbitrary $v \times v$ matrix with coefficients from \mathbb{Z}_p . Hence $\alpha_i b_i$ can be viewed as applying linear map α_i to (b_i, \dots, b_i) . In this way $\Sigma_{(p, \nu)}$ gate computes a function of type $\{0, 1\}^n \rightarrow \mathbb{Z}_p^v$. Additionally, for $c \in \mathbb{Z}_p$ let $\Sigma_{(p, \nu)}^c$ denote a boolean variant of $\Sigma_{(p, \nu)}$ gate which returns value 1 when the affine combination $\alpha_1 b_1 + \dots + \alpha_n b_n + d$ evaluates to c , and returns 0 otherwise. Note that for a field \mathbb{F} with underlying group \mathbb{Z}_p^v , every scalar $g \in \mathbb{F}$ defines a linear map $x \mapsto g \cdot x$. Thus such g (or rather its associated endomorphism) can be used as a coefficient α_i inside $\Sigma_{(p, \nu)}$ gate. To simplify some of the later

calculations, we will always assume that vector $(1, \dots, 1)$ is the unit of such a field \mathbb{F} . Note that for a field $\mathbb{F} = (F, \cdot, +, 0, 1)$, for each $e \in F$ we can define a new field $\mathbb{F}_e = (F, \circ, +, 0, e)$, by defining $x \circ y$ to be $x \cdot e^{-1} \cdot y$.

It is indeed a valid assumption, since for every non-zero element $g \in \mathbb{Z}_p^v$ which is not a unit of the field \mathbb{F} over \mathbb{Z}_p^v , we can redefine multiplication to be $x \cdot g^{-1} \cdot y$ in terms of old multiplication, and fix the new inverse of an element x to be $x^{-1} \cdot g^2$. One can easily check that such a rewriting of \mathbb{F} defines a new field in which g plays role of the unit (and we did not alter the underlying group).

Having all these gates, we build a bounded-depth circuits by listing types of gates which are allowed on each layer, starting with the input layer on the left, finishing with the output layer on the right. For instance $\text{AND}_d \circ \text{MOD}_m \circ \text{MOD}_p$ denotes a 3-level circuit with inputs wired to some set of AND_d gates, then this AND_d gates have wires to MOD_m gates and on the output level there is one MOD_p gate. We allow multiple wires between two gates.

7. USEFUL ALGEBRAIC FACTS

Before dealing with the remaining parts of the proofs, we present a number of lemmas describing certain useful aspects of the local behaviour of algebras.

Lemma 7.1. *Let \mathbf{A} be a finite algebra and β be an atom in $\text{Con } \mathbf{A}$. Then for each element $e \in A$ the induced algebra $\mathbf{A}|_{e/\beta}$ on the coset e/β is either trivial or simple.*

Proof. **TOPROVE 4** □

The next lemma is a specialized version of some much more general facts known in Universal Algebra (see for instance [FM87, Corollary 5.8]).

Lemma 7.2. *Let \mathbf{A} be a finite nilpotent algebra from a congruence modular variety with a Malcev term $\mathbf{d}(x, y, z)$ and β be an abelian atom in $\text{Con } \mathbf{A}$. Then for each element $e \in A$ the induced algebra $\mathbf{A}|_{e/\beta}$ on the coset e/β is polynomially equivalent to a simple module in which the underlying group structure is determined by the binary operation $x + y = \mathbf{d}(x, e, y)$ and the corresponding group $(e/\beta; +, e)$ is isomorphic to some power of the group $(\mathbb{Z}_p; +, 0)$.*

Proof. **TOPROVE 5** □

Now we present yet another simple observation.

Lemma 7.3. *Let \mathbf{A} be a finite nilpotent Malcev algebra and $\delta \prec \theta$ two of its congruences. Then every element $e \in A$ belongs to some (δ, θ) -minimal set.*

Proof. **TOPROVE 6** □

Now we demonstrate that easiness of PROGCSAT can be transferred to quotients.

Fact 7.4. *For an algebra \mathbf{A} and its congruence θ there is a polynomial time reduction from $\text{PROGCSAT}(\mathbf{A}/\theta)$ to $\text{PROGCSAT}(\mathbf{A})$.*

Proof. **TOPROVE 7** □

The following simple fact provides us with a way to reason about characteristics of intervals below a join irreducible congruence in the congruence lattice of an algebra.

Fact 7.5. *Let \mathbf{A} be a solvable Malcev algebra and $\alpha, \beta \in \text{Con } \mathbf{A}$ such that $I[\alpha, \beta]$ is a PUPI. Then, for every $\gamma \in \text{Con } \mathbf{A}$ join irreducible in $I[\alpha, \beta]$ it follows that $|\text{char}\{\alpha, \gamma\}| = 1$.*

Proof. **TOPROVE 8** □

8. PROGRAMCSAT – HARDNESS

The next Lemma is modeled after Lemma 3.1 from [IKK18]. It provides us with a normal form for all the s -ary functions of type $\mathbb{Z}_m^s \rightarrow \mathbb{Z}_p$.

Lemma 8.1. *Let m be a square-free positive integer and $p \nmid m$ be a prime. Then every function $f : \mathbb{Z}_m^s \rightarrow \mathbb{Z}_p$ can be expressed by*

$$f(x_1, \dots, x_s) = \sum_{(\bar{\beta}, u) \in \mathbb{Z}_m^s \times \mathbb{Z}_m} \mu_{\bar{\beta}, u} \cdot b\left(\bigoplus_{i=1}^s \beta_i x_i \oplus u\right),$$

where

- $b : \mathbb{Z}_m \rightarrow \mathbb{Z}_p$ is given by $b(0) = 1$ and $b(x) = 0$ for $x \neq 0$,
- $\mu_{\bar{\beta}, u} \in \mathbb{Z}_p$,
- \sum is the addition from \mathbb{Z}_p ,
- \bigoplus is the addition from \mathbb{Z}_m .

Moreover, coefficients $\mu_{\bar{\beta}, u} \in \mathbb{Z}_p$ are computable in $2^{O(s)}$ steps.

Proof. **TOPROVE 9** □

Proof. **TOPROVE 10** □

Lemma 8.2. *Let \mathbf{A} be a finite nilpotent Malcev algebra, α its join irreducible congruence with α^- being its unique subcover. Moreover let $\beta \in \mathbf{Con} \mathbf{A}$ be a subcover of α^- with $\text{char}(\beta, \alpha^-) \neq \text{char}(\alpha^-, \alpha)$. Then for any choice of $(c, d) \in \alpha - \alpha^-$ and $(e, a) \in \alpha^- - \beta$ every function $f : \{c, d\}^s \rightarrow \{e, a\}$ can be β -interpolated by an s -ary polynomial \mathbf{p} of \mathbf{A} , i.e. $f(\bar{x}) \stackrel{\beta}{\equiv} \mathbf{p}(\bar{x})$, whenever $\bar{x} \in \{c, d\}^s$. Moreover such a polynomial \mathbf{p} can be obtained from the function f in $2^{O(s)}$ steps.*

Proof. **TOPROVE 11** □

Proof. **TOPROVE 12** □

9. PROGRAMCSAT – EASINESS

We start this chapter by presenting number of usefull facts about low-depth circuits which use only arithmetic operations and conjunction. Similar results can be found for instance in [Gro01, GT00].

Lemma 9.1. *Every function of the form $\{0, 1\}^n \rightarrow \mathbb{Z}_p^k$ can be computed by an $\text{AND}_n \circ \Sigma_{(p, k)}$ -circuit of size at most 2^n .*

Proof. **TOPROVE 13** □

Lemma 9.2. *Let m be a square-free positive integer, $p \nmid m$ be a prime. Then every function computable by a $\text{MOD}_m \circ \text{AND}_d$ -circuit can be also computed by a $\text{MOD}_m \circ \Sigma_{(p, 1)}$ -circuit of size $O(m^{d+1})$.*

Proof. **TOPROVE 14** □

Lemma 9.3. *Every function computable by a $\text{MOD}_m \circ \text{MOD}_p$ -circuit of size λ can be also computed by a $\text{MOD}_m \circ \Sigma_{(p, 1)}$ -circuit of size $O(m^p \lambda^p)$.*

Proof. **TOPROVE 15** □

Lemma 9.4. *Let m be a square-free positive integer, $p \nmid m$ be a prime and g be a k -ary boolean function. If the functions f_1, \dots, f_k are computable by $\text{MOD}_m \circ \text{MOD}_p$ -circuits of size $O(\lambda)$ then also $g(f_1, \dots, f_k)$ is computable by such a circuit, but of size $O(\lambda^{kp})$.*

Proof. **TOPROVE 16** □

Lemma 9.5. *Let m be a square-free positive integer, $p \nmid m$ be a prime and $c \in \mathbb{Z}_p^\nu$. Then every $\text{AND}_d \circ \text{MOD}_m \circ \text{MOD}_p \circ \text{AND}_{d'} \circ \Sigma_{(p,\nu)}^c$ -circuit of size λ can be replaced a $\text{AND}_d \circ \text{MOD}_m \circ \text{MOD}_p$ -circuit of size $O(\lambda^{\nu d' p^3})$.*

Proof. **TOPROVE 17** □

Now we present a characterization of Boolean functions that can be represented in supernilpotent algebras.

Lemma 9.6. *Let \mathbf{A} be a finite supernilpotent Malcev algebra. Then the functions computable by an n -ary boolean program $(\mathbf{p}, n, \iota, S)$ over the algebra \mathbf{A} can be also computed by $\text{AND}_d \circ \text{MOD}_{\delta(\mathbf{A})} \circ \text{OR}_{|S|}$ -circuits of size $O(n^d)$ with $d \leq |A|^{1+\log \mu(\mathbf{A})} = (2 \cdot \mu(\mathbf{A}))^{\log |A|}$.*

Proof. **TOPROVE 18** □

Theorem 9.7 (Theorem 1.4 restated). *Let \mathbf{A} be a finite nilpotent Malcev algebra with $\text{char}\{0, \kappa\} = \{p\}$. Then the function computable by a boolean program $(\mathbf{p}, n, \iota, S)$ of size ℓ over the algebra \mathbf{A} can be also computed by an $\text{AND}_d \circ \text{MOD}_{\delta(\mathbf{A})/p} \circ \text{MOD}_p$ -circuits of size $O(\ell^c)$ with $d \leq |A|^{1+\log \mu(\mathbf{A})}$ and the degree c bounded by $O(|A| \cdot \mu(\mathbf{A}))^{O(\log |A|)}$.*

Proof. **TOPROVE 19** □

Note here that the degree of the polynomial bounding the size of the circuit, i.e. $O(|A| \cdot \mu(\mathbf{A}))^{O(\log |A|)}$ have two sources. The $\log |A|$ comes from the number of iterative use of claim (9.??), i.e. from the height of the congruence lattice of \mathbf{A} . The $(|A| \cdot \mu(\mathbf{A}))$ -part is a consequence of arity of a binary expansion $f^\circ : \{0, 1\}^{|A| \cdot \mu(f)} \rightarrow A$ coding basic operations f . The careful reader can easily note some room for improvement here to $O(\log |A| \cdot \mu(\mathbf{A}))^{O(\log |A|)}$ by coding the elements of A with $\log |A|$ bits and therefore shrinking the arity of f° to $\log |A| \cdot \mu(f)$.

10. FINAL REMARKS

Constant Degree Hypothesis plays crucial role in our proofs of RP algorithms existence. One can ask if this assumption is really needed. In fact, there are some unconditional results. For example very recent paper [KKK24] shows that $\text{CEQV}(\mathbf{A})$ is in P whenever \mathbf{A} from CM is 2-nilpotent, i.e. it has abelian congruence with abelian quotient. Also supernilpotent algebras admit (unconditional) polynomial-time algorithm for CSAT/CEQV [AM10, Kom17, IK22], and if we allow random bits the time complexity drops down to linear [KK20]. Unfortunately, it is not hard to construct for a given d, m, p an algebra \mathbf{A} with supernilpotent rank equal 2 such that functions computable by $\text{AND}_d \circ \text{MOD}_m \circ \text{MOD}_p$ -circuit are exactly functions computable by, not too long, programs over \mathbf{A} . This, together with our results, shows that (under ETH) showing unconditional algorithms solving PROGCSAT for algebras from congruence modular variety with supernilpotent rank equal 2 is equivalent to proving CDH. Hence, the natural question is if CDH holds.

Problem 1. *Prove or disprove the Constant Degree Hypothesis.*

The natural next step in our investigations is to go outside congruence modular realm. The first problem in such a case is that Tame Congruence Theory and Commutator Theory (our heavily used tools) for arbitrary algebras do not work as well as for algebras from congruence modular varieties. Moreover [IK22, Example 2.8] shows that there is an algebra \mathbf{A} (not contained in congruence modular variety) and its congruence σ such that $\text{CSAT}(\mathbf{A})$ is in P, while $\text{CSAT}(\mathbf{A}/\sigma)$ is NP-complete. This suggests that we cannot expect a nice characterization of polynomial-time cases for CSAT outside CM. On the other hand, we are not aware of any such examples for PROGCSAT and CEQV. In fact we saw that the hardness of $\text{PROGCSAT}(\mathbf{A}/\sigma)$ implies the hardness for $\text{PROGCSAT}(\mathbf{A})$. This gives hope for characterization of tractable cases of PROGCSAT for general finite algebras.

Problem 2. *Characterize finite algebras with tractable PROGCSAT/CEQV.*

REFERENCES

- [Aic19] Erhard Aichinger. Bounding the free spectrum of nilpotent algebras of prime power order. *Israel Journal of Mathematics*, 230(2):919–947, 2019.
- [AM10] Erhard Aichinger and Nebojša Mudrinski. Some applications of higher commutators in Mal’cev algebras. *Algebra universalis*, 63(4):367–403, 2010.
- [Bar85] David A. Mix Barrington. Width-3 permutation branching programs. Technical Report TM-293, MIT Laboratory for Computer Science, 1985.
- [Bar86] David A. Mix Barrington. Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1 . In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC’86)*, pages 1–5, 1986.
- [BBR94] David A. Mix Barrington, Richard Beigel, and Steven Rudich. Representing Boolean Functions as Polynomials Modulo Composite Numbers. *Computational Complexity*, 4:367–382, 1994.
- [BL05] Stanley Burris and John Lawrence. Results on the equivalence problem for finite groups. *Algebra Universalis*, 52(4):495–500, 2005.
- [BMM⁺00] David A. Mix Barrington, Pierre McKenzie, Cristopher Moore, Pascal Tesson, and Denis Thérien. Equation Satisfiability and Program Satisfiability for Finite Monoids. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science (MFCS’2000)*, pages 172–181, 2000.
- [BS11] Stanley Burris and Hanamantagouda P. Sankappanavar. *A Course in Universal Algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer New York, NY, 2011.
- [BST90] David A. Mix Barrington, Howard Straubing, and Denis Thérien. Non-Uniform Automata Over Groups. *Information and Computation*, 89(2):109–132, 1990.
- [Bul00] Andrei Bulatov. On the number of finite Mal’tsev algebras. *Contributions to general algebra*, 13:41–54, 2000.
- [Bul17] Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS’17)*, pages 319–330, 2017.
- [CGPT06] Arkadev Chattopadhyay, Navin Goyal, Pavel Pudlak, and Denis Thérien. Lower bounds for circuits with MOD_m gates. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS’06)*, pages 709–718, 2006.
- [FH20] Attila Földvári and Gábor Horváth. The complexity of the equation solvability and equivalence problems over finite groups. *International Journal of Algebra and Computation*, 30(03):607–623, 2020.
- [FM87] Ralph Freese and Ralph McKenzie. *Commutator Theory for Congruence Modular Varieties*. London Mathematical Society Lecture Notes, No. 125. Cambridge University Press, 1987.
- [GR02] Mikael Goldmann and Alexander Russell. The complexity of solving equations over finite groups. *Information and Computation*, 178(1):253–262, 2002.
- [Gro01] Vince Grolmusz. A degree-decreasing lemma for $(MOD_p\text{-}MOD_m)$ circuits. *Discrete Mathematics & Theoretical Computer Science*, 4(2):247–254, 2001.
- [GT00] Vince Grolmusz and Gábor Tardos. Lower bounds for $(MOD_p\text{-}MOD_m)$ circuits. *SIAM Journal on Computing*, 29(4):1209–1222, 2000.
- [HM88] David Hobby and Ralph McKenzie. *Structure of Finite Algebras*. Contemporary Mathematics vol. 76. American Mathematical Society, 1988.
- [HS11] Gábor Horváth and Csaba Szabó. The extended equivalence and equation solvability problems for groups. *Discrete Mathematics & Theoretical Computer Science*, 13(4):23–32, 2011.
- [HS12] Gábor Horváth and Csaba Szabó. Equivalence and equation solvability problems for the alternating group A_4 . *Journal of Pure and Applied Algebra*, 216(10):2170–2176, 2012.
- [IK22] Paweł M. Idziak and Jacek Krzaczkowski. Satisfiability in MultiValued Circuits. *SIAM Journal on Computing*, 51(3):337–378, 2022.
- [IKK18] Paweł M. Idziak, Piotr Kawalek, and Jacek Krzaczkowski. Expressive Power, Satisfiability and Equivalence of Circuits over Nilpotent Algebras. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117, pages 17:1–17:15, 2018.
- [IKK20] Paweł M. Idziak, Piotr Kawalek, and Jacek Krzaczkowski. Intermediate Problems in Modular Circuits Satisfiability. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS’20)*, pages 578–590, 2020.
- [IKK22a] Paweł M. Idziak, Piotr Kawalek, and Jacek Krzaczkowski. Complexity of Modular Circuits. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS’22)*, pages 32:1–32:11, 2022.
- [IKK22b] Paweł M. Idziak, Piotr Kawalek, and Jacek Krzaczkowski. Satisfiability of circuits and equations over finite Malcev algebras. In *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science (STACS’22)*, pages 37:1–37:14, 2022.
- [IKKW22a] Paweł M. Idziak, Piotr Kawalek, Jacek Krzaczkowski, and Armin Weiß. Equation satisfiability in solvable groups. *Theory of Computing Systems*, 68:740–757, 2022.

- [IKKW22b] Paweł M. Idziak, Piotr Kawałek, Jacek Krzaczkowski, and Armin Weiß. Satisfiability Problems for Finite Groups. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming (ICALP'22)*, volume 229, pages 127:1–127:20, 2022.
- [KK20] Piotr Kawałek and Jacek Krzaczkowski. Even Faster Algorithms for CSAT Over supernilpotent Algebras. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS'20)*, volume 170, pages 55:1–55:13, 2020.
- [KKK24] Piotr Kawałek, Michael Kompatscher, and Jacek Krzaczkowski. Circuit equivalence in 2-nilpotent algebras. In *Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science (STACS'24)*, volume 289, pages 45:1–45:17, 2024.
- [Kom17] Michael Kompatscher. The equation solvability problem over supernilpotent algebras with Mal'cev term. *International Journal of Algebra and Computation*, 28:1005–1015, 2017.
- [Kom22] Michael Kompatscher. CC-circuits and the expressive power of nilpotent algebras. *Logical Methods in Computer Science*, 18(2):12:1–12:15, 2022.
- [Kom23] Michael Kompatscher. CSAT and CEQV for nilpotent Maltsev algebras of Fitting length > 2 . *arXiv eprint*, 2023. arXiv:2105.00689.
- [KW23] Piotr Kawałek and Armin Weiß. Violating Constant Degree Hypothesis requires breaking symmetry. *arXiv eprint*, 2023. arXiv:2311.17440.
- [MS21] Peter Mayr and Ágnes Szendrei. Algebras from congruences. *Algebra universalis*, 82(55), 2021.
- [Wei20] Armin Weiß. 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)*, pages 102:1–102:19, 2020.
- [Zhu20] Dmitriy Zhuk. A Proof of the CSP Dichotomy Conjecture. *Journal of the ACM*, 67(5), 2020.