NONUNIFORM DETERMINISTIC FINITE AUTOMATA OVER FINITE ALGEBRAIC STRUCTURES

PAWEŁ M. IDZIAK

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

PIOTR KAWAŁEK

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 indentify 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 Expotential Time Hypothesis and Constant Degree Hypothesis.

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 $\delta : \Sigma \times Q \ni (\sigma, q) \longmapsto \sigma \cdot q \in Q$. 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 $a : \Sigma \longrightarrow Q^Q$ given by $a(\sigma)(q) = \sigma \cdot q$. Now, the word $\sigma^1 \ldots \sigma^n$ gets accepted if $a(\sigma^1) \cdot \ldots \cdot a(\sigma^n) \in S$, 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 $a(\sigma^1) \cdot \ldots \cdot a(\sigma^n)$ is nothing else but $\mathbf{t}_n(a(\sigma^1), \ldots, a(\sigma^n))$, where $\mathbf{t}_n(x_1, \ldots, x_n) = x_1 \cdot \ldots \cdot x_n$. In NUDFA over the monoid $\mathbf{M}$ to accept the word from Σ^n we are going to relax the term $x_1 \cdot \ldots \cdot x_n$ to an arbitrary semigroup term $\mathbf{t}(x_1, \ldots, x_k)$ and replace one action $a : \Sigma \longrightarrow Q^Q$ by a bunch of functions $a^x : \Sigma \longrightarrow M$, one for each variable of $\mathbf{t}$. Now a $\mathbf{t}$-program (with inputs from Σ^n represented by n-variable word $b_1 \ldots b_n$) 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 $\mathbf{t}_n = (x_1, \ldots, x_{k_n})$ 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 $b^1 \ldots b^n \in \Sigma^n$ gets accepted by such a NUDFA if $\mathbf{t}_n(a^{x_1}(b^{x_1}), \ldots, a^{x_{k_n}}(b^{x_{k_n}})) \in S_n$. 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 $\{0, 1\}^n \longrightarrow \{0, 1\}$. 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 $\iota(x) = (\mathsf{b}^x, a^x(0), a^x(1))$.

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.
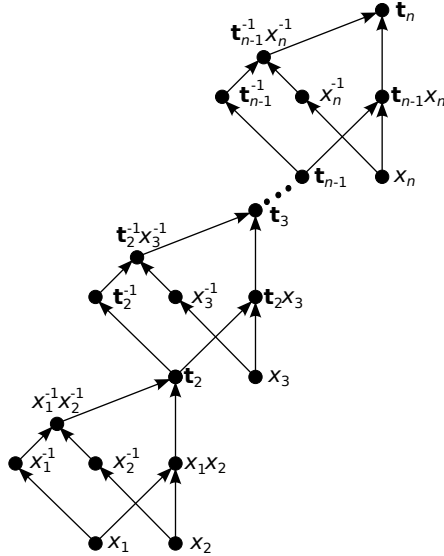
The lower bound in our characterization relies on the randomized version of the Exponential Time Hypothesis (rETH), while the upper bounds explore the so-called Constant Degree Hypothesis (CDH). This hypothesis, introduced in [BST90], can be rephrased to state that for a fixed integer d, a prime p and an integer m which is not just a power of p, any 3-level circuits of the form $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$ 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 $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$ 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 $\mathsf{AND}_d \circ \mathsf{MOD}_m$ 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 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 $\mathbf{A}$.

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 $\mathbf{t}_n(x_1, \ldots, x_n) = [\ldots[[x_1, x_2], x_3]\ldots, x_n]$ 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 $\mathbf{t}_n(x_1, \ldots, x_n) = \mathbf{t}_{n-1}(x_1, \ldots, x_{n-1})^{-1} \cdot x_n^{-1} \cdot \mathbf{t}_{n-1}(x_1, \ldots, x_{n-1}) \cdot x_n$, so that $|\mathbf{t}_n|$ is exponential on n. Actually for the alternating group $\mathbf{A}_4$ it is shown in [HS12] that 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 $\mathbf{t}_n(x_1, \ldots, x_n)$ can be computed by the circuit

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

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 $[\alpha, \alpha] = 0_{\mathbf{A}}$, $[\ldots[[\alpha, \alpha], \alpha], \ldots \alpha] = 0_{\mathbf{A}}$ or $[[[\alpha, \alpha], [\alpha, \alpha]], \ldots [[\alpha, \alpha], [\alpha, \alpha]]] = 0_{\mathbf{A}}$ respectively (for some number of nested commutators). Here, $0_{\mathbf{A}}$ is the identity relation/congrueence 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 $\mathsf{MOD}_2 \circ \mathsf{MOD}_3$ 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* PROGCSAT($\mathbf{A}$) *is in RP if and only if $\mathbf{A}$ is nilpotent and has a supernilpotent congruence σ with supernilpotent quotient $\mathbf{A}/\sigma$ and such that all cosets of σ have sizes that are powers of the same prime number p.*

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

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 **G** by bounded-depth circuits that use only modular gates. Besides the obvious requirement that the circuit representing a group-polynomial **p** has to compute the very same function as **p** does, we also want to control the size of the circuit to be polynomial in terms of the size (length) of **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-crcuits. However, since [Kom22] does not use the notion of a program/NUDFA, the author formulates his results for circuits over **A** representing only some specific functions. For those functions, it is natural how to interpret Boolean values 0/1 in the non-Boolean algebra **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* **A** *be a finite nilpotent algebra from a congruence modular variety with supernilpotent congruence* σ *of* **A** *such that cosets of* α *are of prime power size* p^k *and* **A**$/\sigma$ *is supernilpotent. Then the function computable by a boolean program of size* ℓ *over the algebra* **A** *can be also computed by an* $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-*circuits of size* $O(\ell^c)$ *with* d, m, c *being natural numbers depending only on* **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. Pol **A** is polynomial clone of **A**, that is the set of all polynomial operations of **A**. An algebra **A** is polynomially equivalent to an algebra **B** if it is isomorphic to an algebra which has the same set of polynomial operations as **B**. An induced algebra **A**$|_S$ is a set S with all polynomial operations of **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 typ{**A**}, let us denote a subset of {**1**, .., **5**} which describes the local behaviours we can find in an algebra **A**. Note that in a case of algebra **A** from a congruence modular variety only three types can appear in typ{**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 nad 1) such that

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

If we can find type **3** or **4** in typ{**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 typ{**A**} = {**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 Malcsev, i.e. it possesses a polynomial operation **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 **A** is an equivalence relation which is preserved by the operations of **A**. Such relations are naturally associatted with surjective homomorphisms from **A** to **A**/σ mapping x to $[x]_\sigma$ (equivalence class of x in σ), so congruencess 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 Con **A** for the set of all congruences of **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 writen as a meet (join) of any other two elements of the lattice. These special elements, generating Con **A**, will play a significant role in our analysis.

For $\alpha, \beta \in$ Con **A**, by $I[\alpha, \beta]$ we mean a set of congruencess γ such that $\alpha \leqslant \gamma \leqslant \beta$. In case when $I[\alpha, \beta] = \{\alpha, \beta\}$, i.e. there are no congruencess 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 $\alpha \prec \beta$ for short. Whenever α is meet-irreducible (join-irreducible) congruence, then there is a unique congruence α^+ (α^-) such that $\alpha \prec \alpha^+$ ($\alpha^- \prec \alpha$). For a nilpotent algebra **A** from CM wheneverver its congruencess α, β satisfy $\alpha \prec \beta$, the cosets (congruence classes) of β/α in **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 char(α, β) and call it a characteristic of a congruence cover $\alpha \prec \beta$. Moreover for arbitrary $\alpha < \beta$, we write 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 **A** forms a Prime Uniform Product Interval (PUPI), if there are congruencess $\alpha < \alpha_1, \ldots, \alpha_k \leqslant \beta$ 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 $|\mathsf{char}\{\alpha, \alpha_i\}| = 1$.

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 **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 **A** has supernilpotent rank k whenever k is the smallest number for which we can find sequence of congruences $0_\mathbf{A} = \alpha_0 < \alpha_1 < \ldots < \alpha_k = 1_\mathbf{A}$ such that interval $I[\alpha_i, \alpha_{i+1}]$ is a PUPI for each $0 \leqslant i < k$. Note that for a finite nilpotent algebra **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 sr(**A**), proved to be extremely usuful 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 sr(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 PolSat($\mathbf{G}$) for finite group $\mathbf{G}$ reduces in polynomial time to 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* 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. We start with recalling that [IKKW22a, Theorem 1] tells us that under ETH, PolEqv($\mathbf{G}$) $\in$ P forces $\mathbf{G}$ to have a normal nilpotent subgroup $\mathbf{H}$ with nilpotent quotient $\mathbf{G}/\mathbf{H}$. The very same proof actually gives the same structure of $\mathbf{G}$ under rETH and PolEqv($\mathbf{G}$) $\in$ RP.

For the converse we start by observing that the nilpotent normal subgroup $\mathbf{H}$ of $\mathbf{G}$ is a product of its Sylow subgroups, say $\mathbf{H}_1, \ldots, \mathbf{H}_s$ with $\mathbf{H}_j$ being a p_j-group. Each such subgroup $\mathbf{H}_j$ is isomorphic to the quotient $\mathbf{H}/\mathbf{H}_j'$ for the normal subgroup $\mathbf{H}_j'$ of $\mathbf{H}$ consisting of all elements in H with the order not divisible by p_j. In particular $H_1' \cap \ldots \cap H_s' = \{1\}$. Moreover for an inner automorphism h of $\mathbf{G}$ we have not only $h(H) = H$ but also $h(H_j') = H_j'$, as h has to preserve the order of elements. This means that $\mathbf{H}_j'$ is normal also in $\mathbf{G}$.

Now we know that the quotient $\mathbf{G}/\mathbf{H}_j'$ has a nilpotent normal subgroup $\mathbf{H}/\mathbf{H}_j'$ with a nilpotent quotient $(\mathbf{G}/\mathbf{H}_j')/(\mathbf{H}/\mathbf{H}_j') = \mathbf{G}/\mathbf{H}$. Thus CDH and Theorem 3.1 give us that ProgSat$\left(\mathbf{G}/\mathbf{H}_j'\right)$ and consequently PolSat$\left(\mathbf{G}/\mathbf{H}_j'\right)$ are in RP. Consequently PolEqv$\left(\mathbf{G}/\mathbf{H}_j'\right) \in$ RP as deciding whether $\mathbf{t} = \mathbf{s}$ holds in $\mathbf{G}/\mathbf{H}_j'$ reduces to check if none of the $\left|G/H_j'\right| - 1$ equations of the form $\mathbf{t}\mathbf{s}^{-1} = a$, with $a \in G/H_j' - \{1\}$ has a solution.

Finally, $H_1' \cap \ldots \cap H_s' = \{1\}$ tells us that an equation holds in $\mathbf{G}$ iff it holds in all the quotients $\mathbf{G}/\mathbf{H}_j'$, so that PolEqv($\mathbf{G}$) $\in$ RP. $\qquad\square$

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* CSat($\mathbf{A}$) *and* CEqv($\mathbf{A}$) *are Turing reducible to* ProgCSat($\mathbf{A}$).

Proof. To prove the fact we start with fixing $a_0 \in A$ and enumerating $A - \{a_0\} = \{a_1, \ldots, a_k\}$ to form $A_j = \{a_0, a_j\}$. Using Malcev term $\mathbf{d}$ we define the k-ary polynomial $\mathbf{f}$ by putting

$$\mathbf{f}(x_1, \ldots, x_k) = \mathbf{d}(\ldots \mathbf{d}(\mathbf{d}(x_1, a_0, x_2), a_0, x_3) \ldots, a_0, x_k).$$

Obviously $\mathbf{f}(a_0, \ldots, a_0) = a_0$. To see that the other a_j's are in $\mathbf{f}(A_1, \ldots, A_k)$ simply evaluate x_j by $a_j \in A_j$ and the rest of the x_i's by a_0.

Observe that if $\mathbf{A}$ is a nilpotent algebra with a Malcev term $\mathbf{d}(x, y, z)$ and $e, a, b \in A$ then, by [FM87, Lemma 7.3] we have $a = b$ iff $\mathbf{d}(a, b, e) = e$. This immediately shows that in nilpotent Malcev algebras only equations of the form $\mathbf{t}(\overline{x}) = e$ (i.e. equations in which one side is a constant polynomial) need to be considered in satisfiability or equivalence problems.

Thus we start with an instance of CSat [or CEqv] $\mathbf{t}(\overline{x}) = e$ and define kn-ary polynomial

$$\mathbf{t}'(x_1^1, \ldots, x_1^k, \ldots, x_n^1, \ldots, x_n^k) = \mathbf{t}(\mathbf{f}(x_1^1, \ldots, x_1^k), \ldots, \mathbf{f}(x_n^1, \ldots, x_n^k)).$$

Due to $A = \mathbf{f}(A_1, \ldots, A_k)$ we easily get that $\mathbf{t} = e$ has a solution in $\mathbf{A}$ [holds identically in $\mathbf{A}$] iff $\mathbf{t}' = \mathbf{e}$ has a solution with x_i^j restricted to be taken from A_j [or holds for all 2^{kn} evaluations of the x_i^j's in A_j, respectively].

We define the reduction which for a given instance of $\mathrm{CSAT}(\mathbf{A})$ [$\mathrm{CEQV}(\mathbf{A})$] in the form $\mathbf{t}(\overline{x}) = e$ returns nk-ary boolean $\mathbf{t}'$-program (over the variables $\mathsf{b}_1^1, \ldots, \mathsf{b}_1^k, \ldots, \mathsf{b}_n^1, \ldots, \mathsf{b}_n^k$) with the instructions $\iota(x_i^j) = (\mathsf{b}_i^j, a_0, a_j)$. One can easily see that these reductions work, after setting the accepting values to be $S = \{e\}$ in case of $\mathrm{CSAT}(\mathbf{A})$, and $S = A - \{e\}$ for $\mathrm{CEQV}(\mathbf{A})$.   □

In the proof of Theorem 1.2 we will use Fact 4.1 to show that, under our assumptions, nilpotent Malcev algebra with tractable $\mathrm{PROGCSAT}$ 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 $\mathrm{PROGCSAT}$ is tractable.

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

Then $\mathbf{A}$ has a supernilpotent congruence α with cosets of prime power order such that quotient algebra $\mathbf{A}/\alpha$ 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 $\nu \geqslant 1$ be an integer. Then for each 3-CNF formula $\Phi(\overline{x})$ with n variables there is a polynomial $w_p^\Phi(\overline{x})$ over $GF(p)$ of degree at most $O(p^\nu)$ such that for all $\overline{b} \in \{0,1\}^n$ we have*

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

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 $p_i^{\nu_i - 1} \leqslant \sqrt{m} < p_i^{\nu_i}$, we will get, by Chinese Remainder Theorem, that Φ is satisfied by $\overline{b} \in \{0,1\}$ iff $w_{p_1}^\Phi(\overline{b}) = w_{p_2}^\Phi(\overline{b}) = 0$. Moreover, the lengths of $w_{p_1}^\Phi(\overline{b})$ and $w_{p_2}^\Phi(\overline{b})$ are subexponential in the size of Φ. The core of the proof of Lemma 4.2 is showing (with haevilly use of Tame Congruence Theory and Commutator Theory) that if nilpotent Malcev algebra $\mathbf{A}$ with supernilpotentn rank 2 has supernilpotent congruence α which cosets are not of prime power size and such that $\mathbf{A}/\alpha$ is supernilpotent then we can simulate by programs over $\mathbf{A}$ systems of equations in the form:

$$w_{p_1}^\Phi(\overline{b}) = 0,$$
$$w_{p_2}^\Phi(\overline{b}) = 0.$$

Now we are ready to prove Theorem 3.1

Proof of Theorem 3.1: We start with the following observation:

(4.1) $\mathrm{PROGCSAT}((\{0,1\}; \wedge, \vee))$ is NP-complete.

To see that we start with an n-ary CNF-formula $\Phi(b_1, \ldots, b_n)$, treat it as a function $\{0,1\}^n \longrightarrow \{0,1\}$, and convert to n-ary program over the lattice $(\{0,1\}; \wedge, \vee)$. First, by introducing the variables $x_1, \ldots, x_n$ and $x_1', \ldots, x_n'$ we produce a new $2n$-ary formula $\Phi'(x_1, \ldots, x_n, x_1', \ldots, x_n')$ by simply replacing each positive literal b_i by x_i and negative literal $\neg b_i$ by x_i'. This leads to a function $\Phi' : \{0,1\}^{2n} \longrightarrow \{0,1\}$ and allows us to transform the formula Φ to the program $(\Phi', n, \iota, \{1\})$ by putting $\iota(x_i) = (b_i, 0, 1)$ and $\iota(x_i') = (b_i, 1, 0)$.

Using (4.1) we can exclude TCT types **3** and **4** from the typeset $\mathrm{typ}\{\mathbf{A}\}$ so that:

(4.2) Either $\mathrm{PROGCSAT}(\mathbf{A})$ is NP-complete or $\mathbf{A}$ is solvable.

Actually we can force $\mathbf{A}$ to be nilpotent. Indeed, Lemma 2.2 of [IKK22b] supplies us with an element $e \in A$ and a partition of A into two nonempty disjoint subsets $A = A_0 \cup A_1$ which allows to associate (in linear time $O(m)$) with a 3-CNF-formula Φ (with m clauses and n variables) a $3m$-ary circuit(polynomial) $\mathbf{sat}_\Phi$ of $\mathbf{A}$ such that for $b_1^1, b_2^1, b_3^1, \ldots, b_1^m, b_2^m, b_3^m \in \{0,1\}$ and $x_1^1, x_2^1, x_3^1, \ldots, x_1^m, x_2^m, x_3^m \in A$ with $x_i^j \in A_{b_i^j}$ we have

$$\Phi(b_1^1, b_2^1, b_3^1, \ldots, b_1^m, b_2^m, b_3^m) = 1 \quad \text{iff} \quad \mathbf{sat}_\Phi(x_1^1, x_2^1, x_3^1, \ldots, x_1^m, x_2^m, x_3^m) = e.$$

Thus fixing $a_0 \in A_0$ and $a_1 \in A_1$ we end up with a program $(\mathbf{sat}_\Phi, 3m, \iota, \{e\})$ over $\mathbf{A}$, where $\iota(x_i^j) = (b_i^j, a_0, a_1)$. This reduction from 3-CNF-SAT to $\textsc{ProgCSat}(\mathbf{A})$ shows that:

(4.3) Either $\textsc{ProgCSat}(\mathbf{A})$ is NP-complete or $\mathbf{A}$ is nilpotent.

To enforce that tractability of $\textsc{ProgCSat}(\mathbf{A})$ enforces $\mathbf{A}$ to have supernilpotent rank 2 we refer to [IKK20], where $\mathsf{sr}\,(\mathbf{A}) \geqslant 3$ gives a chain $p_1 \neq p_2 \neq p_3$ of primes occurring as characteristics in consecutive PUPI in $\mathsf{Con}\,\mathbf{A}$. Moreover [IKK20] shows how to use one alternation of characteristics to represent n-ary AND by a polynomial/circuit of size $2^{O(n)}$, and further how to use two alternations of characteristics to compose such created $\sqrt{n}$-ary AND functions to represent n-ary AND by a polynomial/circuit of size $2^{O(\sqrt{n})}$. From this one expects that under (r)ETH $\textsc{CSat}(\mathbf{A})$ is not in (R)P if $\mathsf{sr}\,(\mathbf{A}) \geqslant 3$. In fact [IKK20] provides examples of such algebras, while [Kom23] contains a nice proof of this expectation. This, together with Fact 4.1, gives us that:

(4.4) If $\textsc{ProgCSat}(\mathbf{A}) \in (\mathsf{R})\mathsf{P}$ then $\mathsf{sr}\,(\mathbf{A}) \leqslant 2$, or (r)ETH fails.

Now, the immediate consequence of (4.4) and Lemma 4.2 is that:

(4.5) If $\textsc{ProgCSat}(\mathbf{A}) \in (\mathsf{R})\mathsf{P}$ then there exists supernilpotent congruence α of $\mathbf{A}$ with costes of prime power size and such that $\mathbf{A}/\alpha$ is supernilpotent, or (r)ETH fails.

Finally, let $\mathbf{A}$ be a nilpotent algebra of supernilpotent rank 2 having supernilpotent congruence α of $\mathbf{A}$ with costes of prime power size and such that $\mathbf{A}/\alpha$ is supernilpotent. In such the case programs over $\mathbf{A}$ computing n-ary AND functions have size at least $2^{\Omega(n)}$, or (r)ETH and CDH fails. This is an immediate consequence of Theorem 1.4, for if not, the AND_n function computed by a program of size $\ell(n)$ can be also computed by a $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit of size $O(\ell(n)^c)$ for the constants c, d, m, p depending only of the algebra $\mathbf{A}$. But then CDH tells us that $O(\ell(n)^c)$, and therefore $\ell(n)$ itself, has to dominate $2^{\Omega(n)}$. Now, to prove that $\textsc{ProgCSat}(\mathbf{A})$ is in RP recall that the second part of [IKKW22b, Proposition 9] tells that a set of binary words accepted by a program $(\mathbf{p}, n, \iota, S)$ of size ℓ is either empty or has the size at least $2^n/\ell^c$, for some constant c. Thus checking at least ℓ^c random boolean words of length n finds a word accepted by the program (if there is one at all) with probability at least $1/2$. This obviously puts $\textsc{ProgCSat}(\mathbf{A})$ into RP. $\square$

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 o Theorem 1.3.

Proof of Theorem 1.3. First suppose that $\textsc{CEqv}(\mathbf{A})$ is tractable to eliminate types **3** and **4** from the $\mathsf{typ}\{\mathbf{A}\}$. It should be obvious how to do it for type **3**. For the lattice type **4** we first note that the existence of a solution (in the two element lattice) to the systems of two equations of the form $\mathbf{m}(\overline{x}) = 1$ & $\mathbf{j}(\overline{x}) = 0$ is NP-complete, where $\mathbf{m}(\overline{x})$ is in CNF and $\mathbf{j}(\overline{x})$ is in DNF, both with only positive literals. But obviously this system has no solutions iff $\mathbf{m}(\overline{x}) \vee \mathbf{j}(\overline{x}) = \mathbf{j}(\overline{x})$ holds identically in $(\{0,1\}; \wedge, \vee)$. To put this into the minimal set U of type **4** simply replace each variable x by $\mathbf{e}_U(x)$ (where $\mathbf{e}_U$ is an idempotent polynomial of $\mathbf{A}$ with the range U) and the operations $\wedge, \vee$ by the corresponding polynomials of $\mathbf{A}$ turning $\mathbf{A}|_U$ into the two element lattice. This shows that $\mathsf{typ}\{\mathbf{A}\} \subseteq \{\mathbf{2}\}$ (i.e. $\mathbf{A}$ is solvable) or $\textsc{CEqv}(\mathbf{A})$ is co-NP-complete.

Now the discussion made in [IKK22b, Section 4] between Problems 2 and 3 shows that (under (r)ETH) in fact $\mathbf{A}$ has to be nilpotent and that $\mathsf{sr}\,(\mathbf{A}) \leqslant 2$. This shows the 'only if' direction of our theorem.

To prove the converse note that an identity holds in an algebra $\mathbf{A}$ iff it holds in all quotients of $\mathbf{A}$ by meet-irreducible congruences (as the intersection of all meet-irreducible congruences is the identity relation $\mathbf{0_A}$). Let α be a meet-irreducible congruence of $\mathbf{A}$. To complete the proof it suffices to show that (under CDH) $\mathrm{CEQV}(\mathbf{A}/\alpha) \in \mathsf{RP}$.

Observe that if $\mathbf{A}$ is nilpotent and $\mathsf{sr}\,(\mathbf{A}) \leqslant 2$ then each quotient of $\mathbf{A}$ has these two properties as well. Moreover, identity relation in $\mathbf{A}/\alpha$ (i.e. $0_{\mathbf{A}/\alpha}$) has the unique cover. Since congruence classes for covering pairs have equal sizes [FM87, Colloraly 7.5], it is clear from definition of supernilpotency that every supernilpotent congruences of $\mathbf{A}/\alpha$ has cosets of prime power sizes (as for every $\beta_1, \beta_2 \in \mathsf{Con}\,\mathbf{A}/\alpha$, if $\beta_1, \beta_2 > 0_{\mathbf{A}/\alpha}$, then $\beta_1 \wedge \beta_2 > 0_{\mathbf{A}/\alpha}$). Therefore by Theorem 1.2, $\mathrm{PROGCSAT}(\mathbf{A}/\alpha) \in \mathsf{RP}$. Finally Fact 4.1 gives us $\mathrm{CEQV}(\mathbf{A}) \in \mathsf{RP}$, as required. $\qquad\square$

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 $\alpha, \beta \in \mathsf{Con}\,\mathbf{A}$ 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)-mimal 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 $\alpha, \beta, \gamma \in \mathsf{Con}\,\mathbf{A}$ $\alpha \leqslant \beta$ 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 $\mathsf{Con}\,\mathbf{A}$, 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 **A** we write σ for a biggest supernilpotent congruence and κ for the smallest congruence such that $\mathbf{A}/\kappa$ is supernilpotent. Moreover, we write σ_p for the biggest supernilpotent congruence with cosets of size being a power of p. The existance 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 $p_1^\alpha \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_s^{\alpha_s}$ we write $\Delta(m) = \{p_1, \ldots, p_s\}$ for the set of prime divisors of m and $\delta(m) = p_1 \cdot \ldots \cdot p_s$ 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 **A** and $\mu(f)$ is the arity of the function f. For a set $X = \{d_1, \ldots, d_s\}$ and a function $f : X^k \longrightarrow Y$ we associate to f its binary expansion $f^\circ : \{0,1\}^{|X| \cdot k} \longrightarrow Y$, i.e., arbitrary function satisfying

$$f(x_1, \ldots, x_k) = f^\circ(x_1\text{==}d_1, \ldots, x_1\text{==}d_s, \ldots, x_k\text{==}d_1, \ldots, x_k\text{==}d_s).$$

Programs. Recall that a program $(\mathbf{p}, n, \iota, S)$ computes a Boolean function $\{0,1\}^n \to \{0,1\}$. 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 **A** we can naturally associate an inner function $(\mathbf{p})[\iota] : \{0,1\}^n \to A$ satisfying $(\mathbf{p})[\iota](\bar{b}) \in S$ iff $(\mathbf{p})[\iota, S](\bar{b}) = 1$. This function is computed by the expression $\mathbf{p}(a^{x_1}(b^{x_1}), \ldots, a^{x_k}(b^{x_k}))$, where all a^{x_i}, b^{x_i} are provided by the instructions $\iota(x) = (b^x, a^x)$. For a congruence $\sigma \in \mathsf{Con}\,\mathbf{A}$ 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 $\mathbf{A}/\sigma$, by simply reinterpreting $\mathbf{p}$ to be a circuit over $\mathbf{A}/\sigma$ (algebras **A** and $\mathbf{A}/\sigma$ 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 unbouded 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 $b_1, \ldots, b_n$, and computes their affine combination $\alpha_1 b_1 + \ldots + \alpha_n b_n + d$ in $\mathbb{Z}_p^\nu$. Here we interpret each b_i as v-dimensional vector $(b_i, \ldots, b_i)$. 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 $(b_i, \ldots, b_i)$. In this way $\Sigma_{(p,\nu)}$ gate computes a function of type $\{0,1\}^n \to \mathbb{Z}_p^\nu$. 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 $\alpha_1 b_1 + \ldots + \alpha_n b_n + d$ evaluates to c, and returns 0 otherwise. Note that for a field $\mathbb{F}$ with underlying group $\mathbb{Z}_p^v$, every scalar $g \in F$ defines a linear map $x \longmapsto g \cdot x$. 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 $(1, \ldots, 1)$ 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 easly 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 $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$ 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* $\mathsf{Con}\,\mathbf{A}$. *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. We show that collapsing in $\mathbf{A}|_{e/\beta}$ two elements $a \neq b$ we collapse any pair $(c, d) \in (e/\beta) \times (e/\beta)$. Obviously we have $\Theta_{\mathbf{A}}(a, b) = \beta$ so that c and d can be connected by the so-called Malcev chain, i.e. the projections of the pair (a, b) by unary polynomials of $\mathbf{A}$. More precisely it means that there are unary polynomials $\mathbf{p}_0, \ldots, \mathbf{p}_s$ of $\mathbf{A}$ with $c \in \{\mathbf{p}_0(a), \mathbf{p}_0(b)\}$, $\{\mathbf{p}_j(a), \mathbf{p}_j(b)\} \cap \{\mathbf{p}_{j+1}(a), \mathbf{p}_{j+1}(b)\} \neq \emptyset$ and $d \in \{\mathbf{p}_s(a), \mathbf{p}_s(b)\}$. This gives that each of the $\mathbf{p}_j$'s maps e/β into e/β, i.e. $\mathbf{p}_j$ is a basic operation of the induced algebra $\mathbf{A}|_{e/\beta}$. But then the very same chain is a Malcev chain inside $\mathbf{A}|_{e/\beta}$ showing that $(c, d) \in \Theta_{\mathbf{A}|_{e/\beta}}(a, b)$, as claimed. $\square$

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* $\mathsf{Con}\,\mathbf{A}$. *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. By [FM87, Corollary 5.8] we know that the coset e/β with the operation $x + y = \mathbf{d}(x, e, y)$ is a group. Moreover, abelianity of the congruence β gives that the induced algebra $\mathbf{A}|_{e/\beta}$ is abelian, and therefore, by [FM87, Corollary 5.9] is affine, i.e. polynomially equivalent to a finite module M. But Lemma 7.1 ensures us that this finite module is simple. This in particular gives that all non-zero (i.e. different from e) elements of M have the same order, as otherwise for $a, b \in M = e/\beta$ of orders m, n respectively, if $m < n$ then the endomorphism $M \ni x \longmapsto mx \in M$ is non-zero (as $mb \neq e$) and is not surjective (as $ma = e$), contradicting simplicity of M. This shows that the underlying group of the module M is a finite elementary abelian group, i.e. a finite power of the group $(\mathbb{Z}_p; +, 0)$ for some prime p. $\square$

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. Start with any (δ, θ)-minimal set V and pick $(c, d) \in \theta|_V - \delta$. By [FM87, Corollary 7.5] in nilpotent algebras all cosets of the same congruence have the same size. Thus there is $a \in A$ with $(e, a) \in \Theta(c, d) - \delta$. Since $\mathbf{A}$ is Malcev it has the unary polynomial $\mathbf{p}(x)$ such that $\mathbf{p}\binom{c}{d} = \binom{e}{a}$. In particular the polynamial $\mathbf{p}$ does not collapse θ to δ so that, by [HM88, Lemma 2.8], $\mathbf{p}(V)$ is a minimal set and it contains our element e. $\square$

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* PROGCSAT($\mathbf{A}/\theta$) *to* PROGCSAT($\mathbf{A}$).

Proof. Let $(\mathbf{p}, n, \iota, S)$ be a program over $\mathbf{A}/\theta$. Then for $\iota(x) = (b^x, [a_0^x]_\theta, [a_1^x]_\theta)$ we put $\iota'(x) = (b^x, a_0^x, a_1^x)$ and $S' = \bigcup S$ (where a_0^x, a_1^x are arbitrary representants of the respective cosets in $\mathbf{A}/\theta$). Then since θ is a congruence relation, for any $\bar{b} \in \{0,1\}^n$ we have $(\mathbf{p})[\iota](\bar{b}) \in S \iff (\mathbf{p})[\iota'](\bar{b}) \in S'$. So the program $(\mathbf{p}, n, \iota', S')$ over $\mathbf{A}$ computes exactly the same Boolean function as $(\mathbf{p}, n, \iota, S)$ over $\mathbf{A}/\theta$, while both programs clearly have the same size. Obviosuly $(\mathbf{p}, n, \iota', S')$ can be computed from $(\mathbf{p}, n, \iota, S)$ in polynomial (linear) time. $\square$

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* $\alpha, \beta \in \mathsf{Con}\,\mathbf{A}$ *such that* $I[\alpha, \beta]$ *is a PUPI . Then, for every* $\gamma \in \mathsf{Con}\,\mathbf{A}$ *join irreducible in* $I[\alpha, \beta]$ *it follows that* $|\mathsf{char}\{\alpha, \gamma\}| = 1$.

Proof. From the fact that $I[\alpha, \beta]$ is a PUPI we immediately have that β/α and hence γ/α are supernilpotent in $\mathbf{A}/\alpha$. Therefore, $I[\alpha, \gamma]$ is a PUPI . Hence, $\gamma = \bigvee_{i \in I} \alpha_i$ where $\{\alpha_i\}_{i \in I}$ is a set of maximal congruences from $I[\alpha, \gamma]$ with $|\mathsf{char}\{\alpha, \alpha_i\}| = 1$. But γ is join-irreducible in $I[\alpha, \beta]$, and hence in $I[\alpha, \gamma]$, and $\gamma = \bigvee_{i \in I} \alpha_i$ implies that $\gamma = \alpha_i$ for some i. Thus $|\mathsf{char}\{\alpha, \gamma\}| = 1$. $\square$

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 $\mathbb{Z}_m^s \longrightarrow \mathbb{Z}_p$.

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

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

where

- $b : \mathbb{Z}_m \longrightarrow \mathbb{Z}_p$ *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. We start with assuming that m is a prime, say $\Delta(m) = \{q\}$, to recursively define the functions $b_k^q : \mathbb{Z}_q^k \longrightarrow \mathbb{Z}_p$ by putting

$$ b_1^q(x_1) = b(x_1) $$

$$ b_k^q(x_1, \ldots, x_k) = q^{-1} \cdot \left(\sum_{i=0}^{q-1} b_{k-1}^q(x_1, \ldots, x_{k-2}, x_{k-1} \oplus i \cdot x_k) - \sum_{i=1}^{q-1} b_{k-1}^q(x_1, \ldots, x_{k-2}, x_k \oplus i \cdot 1_q)\right), $$

where q^{-1} is the inverse of q in the field $GF(p)$. Then we observe that $b_k^q(x_1, \ldots, x_k) = 1$ if $x_1 = \ldots = x_k = 0$ and $b_k^q(x_1, \ldots, x_k) = 0$ otherwise.

To see this we induct on k and first assume that $x_k \neq 0$. Then in each of the two big sums only one summand can be non-zero, i.e. the one in which the last argument $x_{k-1} \oplus i \cdot x_k$ (or $x_k \oplus i \cdot 1_q$, resp.) is zero. But then each big sum is $b_{k-1}^q(x_1, \ldots, x_{k-2}, 0)$, so that $b_k^q(x_1, \ldots, x_k) = 0$. On the other hand for $x_k = 0$ all summands in the first sum are equal to $b_{k-1}^q(x_1, \ldots, x_{k-1})$.

Simultaneously all summand of the second sum are zero, as $\mathsf{b}_{k-1}^q(x_1, \ldots, x_{k-2}, x_k \oplus i \cdot 1_q) = \mathsf{b}_{k-1}^q(x_1, \ldots, x_{k-2}, i \cdot 1_q)$ and $i \cdot 1_q$ ranges here over all non-zero elements of $\mathbb{Z}_q$. Thus

$$\mathsf{b}_k^q(x_1, \ldots, x_k) = q^{-1} \cdot \left(\sum_{i=0}^{q-1} \mathsf{b}_{k-1}^q(x_1, \ldots, x_{k-1}) - 0 \right) = \mathsf{b}_{k-1}^q(x_1, \ldots, x_{k-1}).$$

Now, for $m = q_1 \cdot \ldots \cdot q_r$ first we observe that the group $\mathbb{Z}_{q_j}$ is isomorphic to the subgroup $\frac{m}{q_j} \cdot \mathbb{Z}_m$ of $\mathbb{Z}_m$, e.g. under sending the unit 1_{q_j} to $\frac{m}{q_j} \cdot 1_m$. Now for $x_1, \ldots, x_k \in \mathbb{Z}_m$ we put $\mathsf{b}_k^j(x_1, \ldots, x_k) = \mathsf{b}_k^{q_j}(\frac{m}{q_j} \cdot x_1, \ldots, \frac{m}{q_j} \cdot x_k)$ to get $\mathsf{b}_k^j(x_1, \ldots, x_k) = 1$ if $x_1 \stackrel{q_j}{\equiv} \ldots \stackrel{q_j}{\equiv} x_k \stackrel{q_j}{\equiv} 0$ and $\mathsf{b}_k^j(x_1, \ldots, x_k) = 0$ otherwise.

Finally we put $\mathsf{b}_k(\overline{x}) = \prod_{j=1}^r \mathsf{b}_k^j(\overline{x})$ to get the functions $\mathsf{b}_k : \mathbb{Z}_m^k \longrightarrow \{0,1\} \subseteq \mathbb{Z}_p$ satisfying $\mathsf{b}_k(x_1, \ldots, x_k) = 1$ iff $x_i \stackrel{q_j}{\equiv} 0$ for all i, j.

Note that the recursive definition of the b_k^j's puts them in a nice form

$$\sum_{(\overline{\beta}, u) \in \mathbb{Z}_m^k \times \mathbb{Z}_m} \mu_{\overline{\beta}, u} \cdot \mathsf{b}\left(\frac{m}{q_j} \cdot \left(\bigoplus_{i=1}^s \beta_i x_i \oplus u \right) \right),$$

i.e. they are $\mathbb{Z}_p$-linear combinations of the expressions of the form $\mathsf{b}(A \cdot \overline{x})$ with $A \cdot \overline{x}$ being $\mathbb{Z}_m$-linear combinations of the $\frac{m}{q_j} \cdot x_i$'s and $\frac{m}{q_j} \cdot 1_m$. Now, distributing these $\mathbb{Z}_p$-linear combinations in the product of the b_k^j's we end up with a $\mathbb{Z}_p$-linear combinations of the expressions of the form $\mathsf{b}(\frac{m}{q_1} \cdot A_1 \overline{x}) \cdot \ldots \cdot \mathsf{b}(\frac{m}{q_r} \cdot A_r \overline{x})$. But this last product is in fact equal to $\mathsf{b}(\frac{m}{q_1} A_1 \overline{x} \oplus \ldots \oplus \frac{m}{q_r} A_r \overline{x})$, as $\mathbb{Z}_m$ is the direct sum of the $\mathbb{Z}'_{q_j}s$, i.e. the subgroups of $\mathbb{Z}_m$ generated by the $\frac{m}{q_j}$'s.

Obviously this final $\mathbb{Z}_p$-linear combination of the $\mathsf{b}(\frac{m}{q_1} A_1 \overline{x} \oplus \ldots \oplus \frac{m}{q_r} A_r \overline{x})$'s can be computed in $m^{O(k)}$ steps, as nice representations of the b_k^j's are computable in $q^{O(k)}$ steps.

To provide a nice representation of an arbitrary function $f : \mathbb{Z}_m^s \longrightarrow \mathbb{Z}_p$ simply take the $\mathbb{Z}_p$-linear combinations of the b_s with appropriate coefficients, i.e. observe that

$$f(x_1, \ldots, x_s) = \sum_{(a_1, \ldots, a_s) \in \mathbb{Z}_m^s} f(a_1, \ldots, a_s) \cdot \mathsf{b}_s(x_1 - a_1, \ldots, x_s - a_s).$$

Again this can be easily done in $m^{O(s)} = 2^{O(s)}$ steps. $\square$

proof of Lemma 4.3. Suppose the formula Φ has ℓ clauses and n variables. The mentioned result of [BBR94], reformulated in [IKK22a, Fact 3.4], endows us with an ℓ-ary polynomial $w(c_1, \ldots, c_\ell)$ of the field $GF(p)$ of degree $p^\nu - 1$ such that for $x \in \{0,1\}^\ell$ we have

$$w(\overline{x}) = \begin{cases} 0, & \text{if the number of zero entries of } \overline{x} \text{ is divisible by } p^\nu, \\ 1, & \text{otherwise.} \end{cases}$$

Note, that we are interested only in the behavior of $w(\overline{x})$ on values from the set $\{0,1\}$. It means that we can assume that the highest power of the variable that occurs in the polynomial is one as for $x \in \{0,1\}$ we have that $x = x^2$. Hence, it is enough to consider polynomials in sparse form consisting of monomials of the form $\prod_{x \in V} x$, where V is a subset of variables of size at most $p^\nu - 1$. Now, observe that considering subsets of variables of size at most $p^\nu - 1$ ordered by its sizes we can step by step determine coefficients for all monomials. Indeed, coefficient α_V cooresponding to monomial $\prod_{x \in V} x$ can be computed from all $\alpha_{V'}$ for $V' \subsetneq V$ and the evaluation of w which puts 1's precisely to the variables in V (and 0's to other variables). Since we consider at most $O(n^{p^\nu}) = O(2^{p^\nu \log n})$ monomials in which is also the upper bound for the size of w, we can compute w in time bounded by $2^{O(p^\nu \log n)}$.

All we need to do to get w_p^Φ is to substitute the variable c_i in w by a polynomial $1 - \hat{l}_1^i \cdot \hat{l}_2^i \cdot \hat{l}_3^i$ of degree 3 that codes a clause $C_i = l_1^i \vee l_2^i \vee l_3^i$ of Φ, i.e. $\hat{l}_j^i$ is set to $1 - x$ if $l_j^i = x$ and to x if $l_j^i = \neg x$. $\square$

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* $\beta \in \mathsf{Con}\,\mathbf{A}$ *be a subcover of* α^- *with* $\mathsf{char}(\beta, \alpha^-) \neq \mathsf{char}(\alpha^-, \alpha)$. *Then for any choice of* $(c, d) \in \alpha - \alpha^-$ *and* $(e, a) \in \alpha^- - \beta$ *every function* $f : \{c, d\}^s \longrightarrow \{e, a\}$ *can be* β-*interpolated by an* s-*ary polynomial* $\mathbf{p}$ *of* $\mathbf{A}$, *i.e.* $f(\overline{x}) \overset{\beta}{\equiv} \mathbf{p}(\overline{x})$, *whenever* $\overline{x} \in \{c, d\}^s$. *Moreover such a polynomial* $\mathbf{p}$ *can be obtained from the function* f *in* $2^{O(s)}$ *steps.*

Proof. To start with we fix a Malcev term $\mathbf{d}(x, y, z)$ for $\mathbf{A}$ and the characteristics $p = \mathsf{char}(\beta, \alpha^-) \neq \mathsf{char}(\alpha^-, \alpha) = q$.

By [HM88, Theorem 2.8.(4)] we can project the pair (c, d) by a unary polynomial, say $\mathbf{g}$, of $\mathbf{A}$ to a (α^-, α)-minimal set U so that $(c', d') = (\mathbf{g}(c), \mathbf{g}(d)) \in \alpha - \alpha^-$. Note that the trace $N = c'/\alpha \cap U$ modulo α^- (i.e. N/α^-) is an elementary q-group with respect to $x \oplus y = \mathbf{d}(x, c', y)$ and c'/α^- being its zero element. This in particular tells us that the subgroup of N/α^- generated by d'/α^- is isomorphic to $\mathbb{Z}_q$. We pick $c_0, c_1, \ldots, c_{q-1} \in A$ so that $c_0 = c'$, $c_1 = d'$ and $c_0/\alpha^-|_U, c_1/\alpha^-|_U, \ldots, c_{q-1}/\alpha^-|_U$ lists all elements of this subgroup and put $N' = \bigcup_{j \in \mathbb{Z}_q} c_j/\alpha^-|_U$.

With the help of Lemma 7.3 we pick a (β, α^-)-minimal set V containing e. Again we know that the trace $M = e/\beta \cap V$ is, modulo β, an elementary p-group with respect to $x + y = \mathbf{d}(x, e, y)$. Pick $b \in M - e/\beta$ and let $\mathbf{h}$ be a unary polynomial of $\mathbf{A}$ witnessing the fact that $(e, b) \in \alpha^- \subseteq \alpha = \Theta(c', d')$, i.e. $\mathbf{h}\binom{c'}{d'} = \binom{e}{b}$. We may additionally assume that $\mathbf{h}(A) \subseteq V$, or replace $\mathbf{h}$ by $\mathbf{e}_v \mathbf{h}$, where $\mathbf{e}_v$ is a unary idempotent polynomial with the range V. Since $p \neq q$ we know that, due to [HM88, Lemma 4.30], the minimal set $U = \mathbf{e}_U(A)$ can contain no (β, α^-)-minimal set, and therefore $\alpha^-|_U \subseteq \beta$. By a similar reason the minimal set V, and therefore its subset $\mathbf{h}(A)$, cannot contain (α^-, α)-minimal set. Thus $\mathbf{h}(\alpha) \subseteq \alpha^-$. Summing up we know that

(8.1) The algebra $\mathbf{A}$ has a unary polynomial $\mathbf{h}$ satisfying
- $\mathbf{h}(A) \subseteq V$,
- $\mathbf{h}(c_0) \in e/\beta$,
- $(\mathbf{h}(c_0), \mathbf{h}(c_1)) \in \alpha^- - \beta$,
- $\mathbf{h}(\alpha) \subseteq \alpha^-$,
- $\mathbf{h}(\alpha^-|_U) \subseteq \beta$.

Now we note that the operations $\oplus$ and $+$ do not need to be associative or commutative in A, but they do enjoy these properties, modulo α^- or β respectively, on the traces N or M.

In the following we will need a polynomial $\mathbf{h}$ that enjoys the properties listed in (8.1) as well as

(8.2) $\sum_{j \in \mathbb{Z}_q} \mathbf{h}(c_j) \notin e/\beta$.

If our original $\mathbf{h}$ does not satisfy (8.2), we simply replace it by $\mathbf{h}'(x) = \mathbf{h}(x \oplus c_1) - \mathbf{h}(c_1)$, so that (8.1) still holds and $\sum_{j \in \mathbb{Z}_q} \mathbf{h}'(c_j) = \sum_{j \in \mathbb{Z}_q} (\mathbf{h}(c_j \oplus c_1) - \mathbf{h}(c_1)) \overset{\beta}{\equiv} \left(\sum_{j \in \mathbb{Z}_q} \mathbf{h}(c_j) \right) - q \cdot \mathbf{h}(c_1) \overset{\beta}{\equiv} -q \cdot \mathbf{h}(c_1) \notin e/\beta$. To see the middle equality note that $\{(c_0 \oplus c_1)/\alpha^-, \ldots, (c_{q-1} \oplus c_1)/\alpha^-\} = \{c_0/\alpha^-, \ldots, c_{q-1}/\alpha^-\}$ and consequently $\sum_{j \in \mathbb{Z}_q} \mathbf{h}(c_j \oplus c_1) \overset{\beta}{\equiv} \sum_{j \in \mathbb{Z}_q} \mathbf{h}(c_j)$, by the last item in (8.1).

Since $a' = \sum_{j \in \mathbb{Z}_q} \mathbf{h}(c_j) \notin e/\beta$ the subgroup of $(M/\beta, +)$ generated by a'/β is isomorphic to $\mathbb{Z}_p$. Moreover with the help of the polynomial $\mathbf{h}$ we define $\mathbf{b}(x) = a' - \sum_{j \in \mathbb{Z}_q} \mathbf{h}(j \cdot x)$ to see that $\mathbf{b}(c_0/\alpha^-) \subseteq a'/\beta$, while $\mathbf{b}(c_j/\alpha^-) \subseteq e/\beta$ for $j \neq 0$.

Now, endowed with the polynomials $\mathbf{b}, \oplus, +$ of $\mathbf{A}$ and inspired by Lemma 8.1 (with $m = q$) we define an s-ary polynomial $\mathbf{p}'$ of $\mathbf{A}$ by putting

$$\mathbf{p}'(x_1, \ldots, x_s) = \sum_{(\overline{\beta}, u) \in \mathbb{Z}_m^s \times \mathbb{Z}_m} \mu_{\overline{\beta}, u} \cdot \mathbf{b} \left(\bigoplus_{i=1}^s \beta_i x_i \oplus u \right),$$

with the coefficients $\mu_{\overline{\beta},u} \in \mathbb{Z}_p$ chosen in a way that secures

$$(1) \qquad \mathbf{p}'(\mathbf{g}(x_1),\ldots,\mathbf{g}(x_s)) \overset{\beta}{\equiv} \begin{cases} a', & \text{if } f(\overline{x}) = a, \\ e, & \text{if } f(\overline{x}) = e, \end{cases}$$

whenever $\overline{x} \in \{c,d\}^s$. Obviously we cannot control the behavior of $\mathbf{p}'$ over the entire algebra $\mathbf{A}$, but it suffices to have control over the trace N, or even over its subset N'. Finally, congruence permutability applied to $(a,e) \in \alpha^- = \beta \vee \Theta(a',e)$ gives a'' such that $a \overset{\beta}{\equiv} a'' \overset{\Theta(a',e)}{\equiv} e$. In particular $\mathbf{A}$ has a unary polynomial $\mathbf{g}'$ with $\mathbf{g}'\binom{a'}{e} = \binom{a''}{e}$. Then the polynomial $\mathbf{p}(x_1,\ldots,x_s) = \mathbf{g}'(\mathbf{p}'(\mathbf{g}(x_1),\ldots,\mathbf{g}(x_s)))$ does the job, as $\mathbf{p}(\overline{x}) \overset{\beta}{\equiv} f(\overline{x})$ for $\overline{x} \in \{c,d\}^s$. $\qquad\qquad \square$

Proof of Lemma 4.2. Assume that $\mathbf{A}$ is the smallest counterexample to our Lemma. This means that two different primes q_0, q_1 occur in the interval $I[0,\kappa]$. Since this interval is a PUPI , this is witnessed by two subcovers, say γ_0, γ_1, of κ with $\mathsf{char}(\gamma_i,\kappa) = q_i$. By minimality of our counterexample we know that $\gamma_0 \cap \gamma_1 = 0$, as otherwise – in view of Fact 7.4 – the quotient algebra $\mathbf{A}/(\gamma_0 \cap \gamma_1)$ would be a smaller counterexample. This, by modularity of $\mathsf{Con}\,\mathbf{A}$ gives that γ_0, γ_1 are atoms in $\mathsf{Con}\,\mathbf{A}$. Again by minimality we argue that in fact γ_0, γ_1 are the only atoms in $\mathsf{Con}\,\mathbf{A}$. Indeed, if there is another atom θ in $\mathsf{Con}\,\mathbf{A}$ then in the smaller quotient $\mathbf{A}/\theta$ of $\mathbf{A}$ we recover our bad configuration, namely the smallest co-supernilpotent congruence $\kappa_{\mathbf{A}/\theta} = (\kappa \vee \theta)/\theta$ with to its subcovers $(\gamma_i \vee \theta)/\theta$ with $\mathsf{char}((\gamma_i \vee \theta)/\theta, (\kappa \vee \theta)/\theta) = \mathsf{char}(\gamma_i, \kappa) = q_i$.

Now, for $i = 0, 1$ we pick a maximal congruence φ_i that dominates γ_i but not κ. Obviously φ_i is meet-irreducible and we call its unique cover by φ_i^+. Since $\varphi_i \not\geqslant \kappa \leqslant \varphi_i^+$, we know that $\mathsf{sr}\left(\mathbf{A}/\varphi_i^+\right) = 1$ while $\mathsf{sr}\left(\mathbf{A}/\varphi_i\right) = 2$. This means that the subdirectly irreducible algebra $\mathbf{A}/\varphi_i$ is not supernilpotent so that in $\mathsf{char}\{\mathbf{A}/\varphi_i\}$ there is a prime different than q_i, i.e. the characteristic of the monolith φ_i^+/φ_i of $\mathbf{A}/\varphi_i$. Again, $\mathsf{sr}\left(\mathbf{A}/\varphi_i^+\right) = 1$ gives that the interval $I\left[\varphi_i^+, 1\right]$ is a PUPI and this different characteristics has to be witnessed by a successor ψ_i of φ_i^+ with say $\mathsf{char}(\varphi_i^+, \psi_i) = p_i \neq q_i = \mathsf{char}(\varphi_i, \varphi_i^+)$.

Now, for a minimal congruence α_i below ψ_i but not below φ_i^+, we know that α_i is join-irreducible (with the unique subcover α_i^-), and that $\varphi_i^+ \prec \psi_i$ projects down to $\alpha_i^- \prec \alpha_i$. In fact one can easily see that the entire interval $I[\varphi_i, \psi_i]$ project down to $I[\varphi_i \cap \alpha_i, \alpha_i]$ and we have:

(8.3) $\varphi_i \cap \alpha_i \prec \alpha_i^- \prec \alpha_i$,
(8.4) $\mathsf{char}(\varphi_i \cap \alpha_i, \alpha_i^-) = q_i \neq p_i = \mathsf{char}(\alpha_i^-, \alpha_i)$.

In fact

(8.5) $\left(\varphi_i \cap \alpha_i = 0 \text{ and } \alpha_i^- = \gamma_{1-i}\right)$ or $\left(\varphi_i \cap \alpha_i \geqslant \gamma_i \text{ and } \alpha_i^- \geqslant \kappa\right)$.

Suppose first that $\varphi_i \cap \alpha_i = 0$. Then $\alpha_i^- \succ \varphi_i \cap \alpha_i = 0$ is one of the atoms γ_0, γ_1. But $\alpha_i^- = \gamma_i$ would give $q_{1-i} = \mathsf{char}(0, \gamma_i) = \mathsf{char}(\varphi_i \cap \alpha_i, \alpha_i^-) = q_i$. In the other case $\varphi_i \cap \alpha_i$ is above one of the atoms γ_0, γ_1. But $\varphi_i \geqslant \varphi_i \cap \alpha_i \geqslant \gamma_{1-i}$ together with $\varphi_i \geqslant \gamma_i$ gives $\varphi_i \geqslant \gamma_i \vee \gamma_{1-i} = \kappa$ contrary to our choice of φ_i. To see that in this case $\kappa \leqslant \alpha_i^-$ suppose otherwise to get $\kappa > \kappa \cap \alpha_i^- \geqslant \gamma_i$ and consequently $\gamma_i = \kappa \cap \alpha_i^-$. Therefore $I[\gamma_i, \alpha_i] \nearrow I[\kappa, \kappa \vee \alpha_i]$ so that $p_i, q_i \in \mathsf{char}\{\varphi_i \cap \alpha_i, \alpha_i\} \subseteq \mathsf{char}\{\gamma_i, \alpha_i\} = \mathsf{char}\{\kappa, \kappa \vee \alpha_i\}$. This however is not possible as α_i is join-irreducible in the lattice $\mathsf{Con}\,\mathbf{A}$ so that $\kappa \vee \alpha_i$ is join-irreducible in the PUPI $I[\kappa, 1]$ and thus the interval $I[\kappa, \alpha_i]$ has only one characteristic (see Fact 7.5).

(8.6) $I[0, \gamma_{1-i}] \nearrow I\left[\varphi_i \cap \alpha_i, \alpha_i^-\right] \nearrow I\left[\varphi_i, \varphi_i^+\right]$

As $I[0, \gamma_{1-i}] \nearrow I[\gamma_i, \kappa] \nearrow I\left[\varphi_i, \varphi_i^+\right]$ then (8.6) is shown for $\varphi_i \cap \alpha_i = 0$. Otherwise (8.5) gives $\varphi_i \cap \alpha_i \geqslant \gamma_i$ and $I[\gamma_i, \kappa] \nearrow I\left[\varphi_i \cap \alpha_i, \alpha_i^-\right]$, as required.

Now note that (8.4) gives $p_i, q_i \in \mathsf{char}\{\varphi_i \cap \alpha_i, \alpha_i\} \subseteq \mathsf{char}\{\gamma_i, \alpha_i\}$. However we can show that

(8.7) $q_i \notin \mathsf{char}\{\gamma_i, \varphi_i \cap \alpha_i\}$

Indeed, if $q_i \in \mathsf{char}\{\gamma_i, \varphi_i \cap \alpha_i\}$ then in particular $\gamma_i < \varphi_i \cap \alpha_i$ and by (8.5) we have $\kappa \leqslant \alpha_i^- < \alpha_i$. But then modularity gives $(\varphi_i \cap \alpha_i) \vee \kappa = (\varphi_i \vee \kappa) \cap \alpha_i = \varphi_i^+ \cap \alpha_i = \alpha_i^-$, while from $I[\gamma_i, \kappa] \nearrow I\left[\varphi_i, \varphi_i^+\right]$ we get $(\varphi_i \cap \alpha_i) \cap \kappa = \varphi_i \cap \kappa = \gamma_i$. These two equalities result in $I[\gamma_i, \varphi_i \cap \alpha_i] \nearrow I\left[\kappa, \alpha_i^-\right]$, which together with $p_i = \mathsf{char}(\varphi_i^+, \psi_i) = \mathsf{char}(\alpha_i^-, \alpha_i) \in \mathsf{char}\{\kappa, \alpha_i\}$

and join-reducibility of α_i in the PUPI $I[\kappa, 1]$ gives $\mathsf{char}\{\gamma_i, \varphi_i \cap \alpha_i\} = \mathsf{char}\{\kappa, \alpha_i^-\} = \{p_i\} \not\ni q_i$ as claimed.

We fix an element $e \in A$ and use Lemma 7.3 to get $(\varphi_i \cap \alpha_i^-, \alpha_i^-)$-minimal set V_i that contains e. Note that the minimal sets V_i have different characteristics, so that they cannot be equal. Therefore by [HM88, Lemma 4.30] we have

(8.8) $V_0 \cap V_1 = \{e\}$.

Now let $N_i = V_i \cap e/\alpha_i^-$ is a $(\varphi_i \cap \alpha_i^-, \alpha_i^-)$-trace in the minimal set V_i. Obviously, due to (8.6) the set V_i is also $(0, \gamma_{1-i})$-minimal, but in general it may have smaller traces. This however is not possible as, due to (8.7), we have

(8.9) $\alpha_i^-|_{V_i} \subseteq \gamma_{1-i}$ and $\varphi_i \cap \alpha_i^-|_{V_i} = 0$.

Indeed, we fix a chain of congruences $\gamma_{1-i} = \theta_0 \prec \theta_1 \prec \ldots \prec \theta_s = \alpha_i^-$ to show that $\theta_j|_{V_i} \subseteq \theta_{j-1}$. First observe that $I[\theta_j \cap \varphi_i \cap \alpha_i, \theta_j] \searrow I[\theta_{j-1} \cap \varphi_i \cap \alpha_i, \theta_{j-1}]$ so that V_i is $(\theta_j \cap \varphi_i \cap \alpha_i, \theta_j)$-minimal. If $\theta_j|_{V_i} \not\subseteq \theta_{j-1}$ then the set V_i, as the range of some unary idempotent polynomial of $\mathbf{A}$, would contain (θ_{j-1}, θ_j)-minimal set of characteristic different from q_i, the characteristic of V_i. Again this is not possible in view of [HM88, Lemma 4.30], and we are done with the first part of (8.9). To see the second one simply note that $\varphi_i \cap \alpha_i^-|_{V_i} \subseteq \varphi_i \cap \gamma_{1-i} = 0$.

Now we fix $(c_i, d_i) \in \alpha_i - \alpha_i^-$ and $a_i \in V_i - \{e\}$. Note that, due to (8.9) we have even $a_i \in V_i - e/\beta_i$, where $\beta_i = \varphi_i \cap \alpha_i^-$.

With the help of Lemma 8.2 we are going to show the following claim:

(8.10) For a polynomial $w(\overline{x})$ over the field $GF(q_i)$ of degree s one can construct (in $n^{O(s)}$ steps) a polynomial $\mathbf{p}_w(\overline{x})$ of $\mathbf{A}$ such that for $\overline{x} \in \{c_i, d_i\}^n$ we have

$$\mathbf{p}_w(x_1, \ldots, x_n) = \begin{cases} e, & \text{if } w(\pi_i(x_1), \ldots, \pi_i(x_n)) = 0, \\ a_i, & \text{if } w(\pi_i(x_1), \ldots, \pi_i(x_n)) = 1, \end{cases}$$

where $\pi_i : A \longrightarrow GF(q_i)$ satisfies $\pi_i\binom{c_i}{d_i} = \binom{0}{1}$.

We first prove (8.10) for monomials $u(x_1, \ldots, x_s)$ over $GF(q_i)$ of degree s. Lemma 8.2 allows us to $(\varphi_i \cap \alpha_i^-)$-interpolate the monomial u by a polynomial, say $\mathbf{p}_u(x_1, \ldots, x_s)$ of $\mathbf{A}$ so that for $\overline{x} \in \{c_i, d_i\}^s$ we have

$$\mathbf{p}_u(x_1, \ldots, x_s) \stackrel{\varphi_i \cap \alpha_i^-}{\equiv} \begin{cases} e, & \text{if } u(\pi_i(x_1), \ldots, \pi_i(x_s)) = 0, \\ a_i, & \text{if } u(\pi_i(x_1), \ldots, \pi_i(x_s)) = 1. \end{cases}$$

Since $\{e, a_i\} \subseteq V_i$ we may additionally assume that the range of $\mathbf{p}_u$ is contained in V_i, or simply replace $\mathbf{p}_u$ by $\mathbf{e}_{V_i} \mathbf{p}_u$, where $\mathbf{e}_{V_i}$ is an idempotent polynomial of $\mathbf{A}$ with the range V_i. But then the second part of (8.9) allows us to replace $\stackrel{\varphi_i \cap \alpha_i^-}{\equiv}$ by the equality in the above display.

Now with the help of the binary polynomial $x + y = \mathbf{d}(x, e, y)$ we sum up all the polynomials $\mathbf{p}_u$, with u ranging over the monomials w to get $\mathbf{p}_w \in \mathrm{Pol}\,\mathbf{A}$. Note that each $\mathbf{p}_u$ can be obtained in $2^{O(s)}$ steps, so that $\mathbf{p}_w$ requires $n^{O(s)}$ steps. Note also that for $(x_1, \ldots, x_s) \in \{c_i, d_i\}^s$ the values $\mathbf{p}_u(x_1, \ldots, x_s)$ are in $\{e, a_i\}$, so that their sum $\mathbf{p}_w(\overline{x})$ lies in the cyclic subgroup $\langle a_i \rangle$ of $(V_i \cap e/\alpha_i^-; +; e)$ generated by a_i. Summing the polynomials $\mathbf{p}_u$ in the group $\langle a_i \rangle$ isomorphic to $\mathbb{Z}_{q_i}$ corresponds to summarizing monomials of w in the underlying additive group $\mathbb{Z}_{q_i}$ of $GF(q_i)$. Thus the values $\mathbf{p}_w(\overline{x})$ correspond to the values $w(\pi_i(x_1), \ldots, \pi_i(x_n))$ in the way described in (8.10).

To conclude our proof we associate with each 3-CNF formula Φ (with ℓ clauses and n variables $x_1, \ldots, x_n$) a program $(\mathbf{p}^\Phi, n, \iota, e)$ over $\mathbf{A}$ such that $\overline{b} \in \{0, 1\}^n$ satisfies Φ iff $(\mathbf{p}^\Phi)[\iota](\overline{b}) = e$.

The polynomial $\mathbf{p}^\Phi(x_1^0, \ldots, x_n^0, x_1^1, \ldots, x_n^1)$ is going to be $2n$-ary. The instructions of our program are given by $\iota(x_j^i) = (\mathbf{b}_j, c_i, d_i)$. To construct the polynomial $\mathbf{p}^\Phi$ we first pick two integers ν_0, ν_1 satisfying $q_i^{\nu_i - 1} \leqslant \sqrt{\ell} < q_i^{\nu_i}$. Then, with $i = 0, 1$, we use Lemma 4.3 to get the polynomials $w_i^\Phi(\overline{x})$ over $GF(q_i)$ of degree $O(q_i^{\nu_i})$ so that for $\overline{b} \in \{0, 1\}^n$ we have

$$w_i^\Phi(\overline{b}) = \begin{cases} 0, & \text{if the number of unsatisfied (by } \overline{b}\text{) clauses in } \Phi \text{ is divisible by } p_i^{\nu_i} \\ 1, & \text{otherwise.} \end{cases}$$

Now we use (8.10) to translate the w_i^Φ's into the polynomials $\mathbf{p}_i^\Phi$ of $\mathbf{A}$ that satisfy for $\overline{x} \in \{c_i, d_i\}^n$

$$\mathbf{p}_i^\Phi(\overline{x}) = \begin{cases} 0, & \text{if the number of unsatisfied (by } \pi_i(\overline{x}) \text{) clauses in } \Phi \text{ is divisible by } p_i^{\nu_i} \\ a_i, & \text{otherwise.} \end{cases}$$

Finally we let $\mathbf{p}^\Phi$ to be the $2n$-ary polynomial of $\mathbf{A}$ defined by $\mathbf{p}^\Phi(\overline{x}^0, \overline{x}^1) = \mathbf{d}(\mathbf{p}_0^\Phi(\overline{x}^0), \mathbf{p}_1^\Phi(\overline{x}^1), e)$, with $\overline{x}^0, \overline{x}^1$ being disjoint sets of variables. Obviously for $\mathbf{p}_0^\Phi(\overline{x}^0) = e$ and $\mathbf{p}_1^\Phi(\overline{x}^1) = e$ we have $\mathbf{p}^\Phi(\overline{x}^0, \overline{x}^1) = e$. But also $\mathbf{p}^\Phi(\overline{x}^0, \overline{x}^1) = e$ implies $\mathbf{p}_0^\Phi(\overline{x}^0) = \mathbf{p}_1^\Phi(\overline{x}^1)$, by the fact that Malcev polynomial in nilpotent algebra is a permutation with respect to the first variable (see [FM87, Lemma 7.3]) and $\mathbf{d}(\mathbf{p}_1^\Phi(\overline{x}^0), \mathbf{p}_1^\Phi(\overline{x}^1), e) = e$. But then $V_0 \ni \mathbf{p}_0^\Phi(\overline{x}^0) = \mathbf{p}_1^\Phi(\overline{x}^1) \ni V_1$ together with (8.8) gives $\mathbf{p}_0^\Phi(\overline{x}^0) = e = \mathbf{p}_1^\Phi(\overline{x}^1)$. Thus $e = (f)[\iota](\overline{b}) = \mathbf{p}^\Phi(\pi_0^{-1}(\overline{b}), \pi_1^{-1}(\overline{b}))$ is equivalent to $\mathbf{p}_0^\Phi(\pi_0^{-1}(\overline{b})) = e = \mathbf{p}_1^\Phi(\pi_1^{-1}(\overline{b}))$ or, in other words tells us that the number of unsatisfied (by $\overline{b}$) clauses in Φ is divisible by both $p_0^{\nu_0}$ and $p_1^{\nu_1}$. But this together with $\sqrt{\ell} < q_i^{\nu_i}$ tells us that $\overline{b}$ satisfies the formula Φ.

To complete our proof we bound the time needed to construct the polynomial $\mathbf{p}^\Phi$. First note that by Lemma 4.3 the polynomials w_i^Φ of $GF(q_i)$ can be obtained in $2^{O(q_i^{\nu_i}(\log n + \log p))} = 2^{O(\sqrt{\ell}\log n)}$ steps. Then (8.10) helps us to translate the w_i^Φ's into the polynomials $\mathbf{p}_i^\Phi$ of $\mathbf{A}$ in $n^{O(\sqrt{(l)})} = 2^{O(\sqrt{l}\log n)}$ steps. Thus also the final polynomial $\mathbf{p}^\Phi$ is computable in $2^{O(\sqrt{l}\log n)}$ steps. But this bound, together with rETH, obviously blocks the existence of a polynomial time probabilistic algorithm solving $\textsc{ProgCSat}(\mathbf{A})$. $\qquad\square$

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* $\{0,1\}^n \longrightarrow \mathbb{Z}_p^k$ *can be computed by an* $\mathsf{AND}_n \circ \Sigma_{(p,k)}$-*circuit of size at most* 2^n.

Proof. Our function can be extended to $GF(p^k) \longrightarrow GF(p^k)$ and then represented by a polynomial of $GF(p^k)$ of degree $n \cdot (p^k - 1)$. However, due to the fact that for $x \in \{0,1\}$ and $j \geqslant 1$ we have $x^j = x$, each nonconstant monomial can be replaced by the one of degree between 1 and n. Such monomials, with unit coefficients, can obviously be represented by AND_n-gates. The other coefficients as well as the sums of monomials with arbitrary coefficients are covered by the $\Sigma_{(p,k)}$-gate, since multiplication by element of $GF(p^k)$ is a linear map when we treat $GF(p^k)$ as a k dimensional vector space over $GF(p)$. $\qquad\square$

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

Proof. The boolean function f computable by a $\mathsf{MOD}_m \circ \mathsf{AND}_d$-circuit can be decomposed into $f(\overline{x}) = \mathsf{AND}_d(\chi_{S_1} g_1(\overline{x}), \dots, \chi_{S_d} g_d(\overline{x}))$ for some $\mathbb{Z}_m$-linear functions $g_1, \dots, g_m$ and characteristic functions χ_{S_i} of some subsets $S_1, \dots, S_d \subseteq \mathbb{Z}_m$. With the help of Lemma 8.1 the d-ary function

$$\mathbb{Z}_m^d \ni (y_1, \dots, y_d) \longmapsto \mathsf{AND}_d(\chi_{S_1}(y_1), \dots, \chi_{S_d}(y_d)) \in \{0,1\} \subseteq \mathbb{Z}_p$$

can be presented in the form $\sum_{(\overline{\beta}, u) \in \mathbb{Z}_m^d \times \mathbb{Z}_m} \mu_{\overline{\beta}, u} \cdot \mathsf{b}\left(\bigoplus_{i=1}^d \beta_i y_i \oplus u\right)$. Substituting y_j by $g_j(\overline{x})$ we end up with a function of the same form in which the inner sum $\bigoplus$ increases in arity, but the outer sum has the same number of summands bounded by m^{d+1}. Obviously the expressions of the form $\mathsf{b}(\bigoplus \dots)$ can be computed by MOD_m-gates, while the outer sum is $\Sigma_{(p,1)}$-computable. $\qquad\square$

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

Proof. This time a $\mathsf{MOD}_m \circ \mathsf{MOD}_p$-computable function $f(\overline{x})$ can be decomposed into $\chi_T \left(\sum_{i=1}^{\lambda-1} \alpha_i \cdot \chi_{S_i} g_i(\overline{x}) \right)$ where $T \subseteq \mathbb{Z}_p$, $S_i \subseteq \mathbb{Z}_m$ and the g_i's are $\mathbb{Z}_m$-linear functions. Note that $\chi_T(z)$ can be represented by a unary polynomial, of degree $p-1$ over $GF(p)$. Thus $\chi_T \left(\sum_{i=1}^{\lambda-1} \alpha_i y_i \right)$ is also a polynomial over $GF(p)$ of $\lambda - 1$ variables with degree $p-1$. Thus this polynomial, when restricted to $\{0, 1\}$, is computable by $\mathsf{AND}_{p-1} \circ \Sigma_{(p,1)}$-circuit of size λ^{p-1}, i.e. the maximal possible number of monomials here. Since the $\chi_{S_i} g_i(\overline{x})$'s are MOD_m-computable we end up with $\mathsf{MOD}_m \circ \mathsf{AND}_{p-1} \circ \Sigma_{(p,1)}$-circuit. However Lemma 9.2 help us to replace $\mathsf{MOD}_m \circ \mathsf{AND}_{p-1}$-parts by $\mathsf{MOD}_m \circ \Sigma_{(p,1)}$-circuits of size $O(m^p)$ so that the size of the final circuit computing f can be bounded by $O(m^p \lambda^p)$. $\qquad \square$

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 $f_1, \ldots, f_k$ are computable by $\mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuits of size $O(\lambda)$ then also $g(f_1, \ldots, f_k)$ is computable by such a circuit, but of size $O(\lambda^{kp})$.*

Proof. As in the proof of Lemma 9.1 the function g can be represented as a restriction of a polynomial $g'(z_1, \ldots, z_k)$ of degree k over $GF(p)$ to the set $\{0,1\}^k$. On the other hand, by Lemma 9.3, each of the f_j's is computable by a $\mathsf{MOD}_m \circ \Sigma_{(p,1)}$-circuit of size $O(m^p \lambda^p)$. In particular $f_j(\overline{x})$ can be decomposed into $\sum_{i \in I_j} \alpha_i \cdot \chi_{S_i^j} g_i^j(\overline{x})$ where $S_i^j \subseteq \mathbb{Z}_m$ and the g_i^j's being $\mathbb{Z}_m$-linear. Now plugging the linear polynomials $f_j'(\ldots, y_i^j, \ldots) = \sum_{i \in I_j} \alpha_i \cdot y_i^j$ over $GF(p)$, with at most $O(m^p \lambda^p)$ variables y_i^j, into $g'(z_1, \ldots, z_k)$ we get a polynomial of degree k which, as before, on the arguments from $\{0,1\}$ is $\mathsf{AND}_k \circ \Sigma_{(p,1)}$-computable. Note that the polynomial $g'(f_1', \ldots, f_k')$ may have up to $O(km^p \lambda^p)$ variables, as the f_j's do not need to share variables. Thus the size of $g'(f_1', \ldots, f_k')$ and therefore of the corresponding $\mathsf{AND}_k \circ \Sigma_{(p,1)}$-circuit is bounded by the number of monomials of degree k over $O(km^p \lambda^p)$ variables, i.e. roughly by $O((km^p \lambda^p)^k)$. To compute $g(f_1, \ldots, f_k)$ we simply replace the y_i^j's in the above circuit computing $g'(f_1', \ldots, f_k')$ by MOD_m-circuits computing $\chi_{S_i^j} g_i^j(\overline{x})$ to get $\mathsf{MOD}_m \circ \mathsf{AND}_k \circ \Sigma_{(p,1)}$-circuit of size $O((km^p \lambda^p)^k)$. However, in view of Lemma 9.2, this can be replaced by $\mathsf{MOD}_m \circ \Sigma_{(p,1)}$-circuit, by the expense of increasing the size $O(m^{k+1})$ times. Finally we note that since g returns only 0 or 1 the $\Sigma_{(p,1)}$-gate can be replaced by MOD_p type gate. In our applications we treat m, k and p as constants, so the size of the resulting circuit is $O(\lambda^{kp})$. $\qquad \square$

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 $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p \circ \mathsf{AND}_{d'} \circ \Sigma_{(p,\nu)}^c$-circuit of size λ can be replaced a $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit of size $O(\lambda^{\nu d' p^3})$.*

Proof. We start with noting that for a constant $c \in \mathbb{Z}_p^\nu$ an equation of the form $t = c$ in the module $\mathbb{Z}_p^\nu$ can be replaced by a system of ν equations $t^{[j]} = c^{[j]}$ (with $t^{[j]}$ being the projection of t onto the j-th coordinate) and then by a single equation $g(t^{[1]}, \ldots, t^{[\nu]}) = 1$, where $g(y_1, \ldots, y_\nu) = \prod_{j=1}^\nu (1 - (y_j - c^{[j]})^{p-1})$ is a polynomial of degree $\nu \cdot (p-1)$ over $GF(p)$.

Now we observe that our starting function can be actually computed by a $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p \circ \Sigma_{(p,\nu)}^c$-circuit, by simply using Lemma 9.4 to eliminate the $\mathsf{AND}_{d'}$-level. Thus this function can be represented (with a sum in the group $\mathbb{Z}_p^\nu$) as $\chi_{\{c\}}(f(\overline{x}))$ for $f(\overline{x}) = \sum_{i=1}^s \alpha_i \cdot f_i(\overline{x})$ with α_i being the $k \times k$ matrices over the field $GF(p)$ and $f_1, \ldots, f_s$ being $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-computable. Formally the values $f_i(\overline{x})$ are boolean, but we interpret them by the constant ν-tuples $(f_i(\overline{x}), \ldots, f_i(\overline{x})) \in \mathbb{Z}_p^\nu$.

Observe also that $f(\overline{x})^{[j]} = \sum_{i=1}^s \alpha_{ij} \cdot f_i(\overline{x})$ for $\alpha_{ij} \in \mathbb{Z}_p$ is a sum of the j-th row of the matrix α_{ij}, where this time the boolean values taken by f_i are interpreted by $0, 1 \in \mathbb{Z}_p$. The polynomial g' obtained from g by $g'(z_1, \ldots, z_s) = g(\sum_{i=1}^s \alpha_{i1} \cdot z_i, \ldots, \sum_{i=1}^s \alpha_{i\nu} \cdot z_i)$ has also degree bounded by νp so that, on the set $\{0, 1\}$, it is $\mathsf{AND}_{\nu p} \circ \Sigma_{(p,1)}$-computable. Thus checking if $f(\overline{x}) = c$ reduces to checking $g'(f_1(\overline{x}), \ldots, f_s(\overline{x})) = 1$ which in turn can be done with the help of $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p \circ \mathsf{AND}_{\nu p} \circ \Sigma_{(p,1)}$-circuit. Now Lemma 9.4 and Lemma 9.3 allows us to

eliminate the $\mathsf{AND}_{\nu p}$-level, then replace the two levels $\mathsf{MOD}_m \circ \mathsf{MOD}_p$ by $\mathsf{MOD}_m \circ \Sigma_{(p,1)}$, and finally compress two $\Sigma_{(p,1)}$-levels to a single one and replace it by MOD_p.

A careful analysis of this process gives the required bound for the final $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit. $\qquad\square$

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* $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)} \circ \mathsf{OR}_{|S|}$-*circuits of size* $O(n^d)$ *with* $d \leqslant |A|^{1+\log \mu(\mathbf{A})} = (2 \cdot \mu(\mathbf{A}))^{\log|A|}$.

Proof. First we reduce our setting to $|S| = 1$ as, after building appropriate $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)}$-circuits separately for each $c \in S$, we eventually join them by the final gate $\mathsf{OR}_{|S|}$.

Next we note that the algebra $\mathbf{A}$, being supernilpotent, decomposes into a product $\mathbf{A}_1 \times \ldots \times \mathbf{A}_s$ of supernilpotent algebras $\mathbf{A}_i$, each of which is of prime power order, say $|A_i| = p_i^{\nu_i}$, where $\Delta(A) = \{p_1, \ldots, p_s\}$.

In order to construct $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)}$-circuit computing the boolean function $(\mathbf{p})[\iota, c]$ we first produce an n-ary polynomial $w(\mathsf{b}_1, \ldots, \mathsf{b}_n)$ of degree $d \leqslant |A|^{1+\log \mu(\mathbf{A})}$ over the ring $\mathbb{Z}_{\delta(A)}$ such that for all $\bar{b} \in \{0,1\}$ we have

$$(\mathbf{p})[\iota, c](\bar{b}) = 1 \quad \text{iff} \quad w(\bar{b}) = \delta',$$

where $\delta' = \frac{\delta(A)}{p_1} + \ldots + \frac{\delta(A)}{p_s}$. Given the polynomial $w(\mathsf{b}_1, \ldots, \mathsf{b}_n)$ as above we translate it into $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)}$-circuit by imitating the monomials of w (of degree at most d) by AND_d-gates and then summing them up by an appropriate $\mathsf{MOD}_{\delta(A)}$-gate with accepting set $\{\delta'\}$. Obviously the bound for the degree of the polynomial w determines the upper bound for the size of just constructed circuit, as there are at most $O(n^d)$ monomials of degree d.

The polynomial w is going to be obtained separately for each factor in our decomposition $\mathbf{A} = \mathbf{A}_1 \times \ldots \times \mathbf{A}_s$. In what follows an element a of A is to be presented as a tuple $(a(1), \ldots, a(s))$. Moreover our decomposition of $\mathbf{A}$ leads to the decomposition of the program $(\mathbf{p}, n, \iota, c)$ over $\mathbf{A}$ into s programs $(\mathbf{p}_j, n, \iota_j, c_j)$, over $\mathbf{A}_j$ respectively, where

- $\mathbf{p}_j$ is the polynomial of $\mathbf{A}_j$ corresponding to $\mathbf{p}$,
- $\iota_j(x_i) = (\mathsf{b}^x, a_0^x(j), a_1^x(j))$ whenever $\iota(x) = (\mathsf{b}^x, a_0^x, a_1^x)$,
- $c_j = c(j)$.

Note here that for this decomposition we have

$$(\mathbf{p})[\iota, c](\bar{b}) = 1 \quad \text{iff} \quad \mathbf{p}(a_{b^{x_1}}^{x_1}, \ldots, a_{b^{x_k}}^{x_k}) = (\mathbf{p})[\iota](\bar{b}) = c$$

$$\text{iff} \quad \bigotimes_{j=1}^{s} \mathbf{p}(a_{b^{x_1}}^{x_1}(j), \ldots, a_{b^{x_k}}^{x_k}(j)) = (\mathbf{p}_j)[\iota_j](\bar{b}) = c(j)$$

$$\text{iff} \quad \bigotimes_{j=1}^{s} (\mathbf{p}_j)[\iota_j, c_j](\bar{b}) = 1$$

The plan is to construct, for each $j = 1, \ldots, s$ an appropriate polynomial $w_j(\mathsf{b}_1, \ldots, \mathsf{b}_n)$ (of degree bounded by $|A|^{1+\log \mu(\mathbf{A})}$) over the prime field $GF(p_j)$ satisfying

$$(\mathbf{p}_j)[\iota_j, c_j](\bar{b}) = 1 \quad \text{iff} \quad w_j(\bar{b}) = 1$$

and then sum them up to get the polynomial $w = \frac{\delta(A)}{p_1} \cdot w_1 + \ldots + \frac{\delta(A)}{p_s} \cdot w_s$ over the ring $\mathbb{Z}_{\delta(A)}$ that would clearly satisfy

$$w(\bar{b}) = \delta \quad \text{iff} \quad \bigotimes_{j=1}^{s} w_j(\bar{b}) = 1 \quad \text{iff} \quad (\mathbf{p})[\iota, c](\bar{b}) = 1,$$

as promised.

Before we construct the w_j's note that despite the fact that their variables $\mathsf{b}_1, \ldots, \mathsf{b}_n$ are denoted here to be boolean, they in fact may range over entire $GF(p_j)$ (and finally over entire $\mathbb{Z}_{\delta(A)}$) but we do not care what they return on nonboolean arguments from the set $\{0, 1\}$.

Now we are left with a supernilpotent algebra $\mathbf{A}$ of size p^ν with p being a prime. In such algebra all prime quotients of congruences have characteristic p (since $\mathbf{A}$ has classes of congruences of equal size). Thus, by Lemma 7.2 we know that for each $e \in A$ and each prime quotient $\alpha \prec \beta$ the binary operation $\mathbf{d}(x, e, y)$ defines (modulo α) the structure of an elementary p-group on the set $\{a/\alpha : a \in e/\beta\}$, i.e. a group isomorphic to some power of the group $\mathbb{Z}_p$. Thus, going up along the chain $0 = \beta_0 \prec \beta_1 \prec \ldots \beta_{t-1} \prec \beta_t = 1$ we can recursively conclude that for each j the coset e/β_j can be endowed with a group structure isomorphic to some power of $\mathbb{Z}_p$. Indeed, if $e/\beta_j = a_1/\beta_{j-1} \cup \ldots a_{p^{\nu_j}}/\beta_{j-1}$ and each coset a_i/β_{j-1} can be endowed with a group structure isomorphic to $\mathbb{Z}_p^{\nu_1 + \cdots + \nu_{j-1}}$ then the entire coset can be trated as having the (external) structure of the group product $\mathbb{Z}_p^{\nu_1 + \cdots + \nu_{j-1}} \times \mathbb{Z}_p^{\nu_j}$. This global group operation (which we call external) is not necessarily determined by a polynomial of $\mathbf{A}$ but it does not change the congruences of $\mathbf{A}$. The reader may want to consult [Aic19, KK20] where the structure of supernilpotent algebras of prime power order has been studied in more details.

In particular [KK20] ensures us that the decomposition of the set A into the product Z_p^ν translates a little bit further. Before stating this correspondence we present each element $a \in A$ as a tuple $(a^{(1)}, \ldots a^{(\nu)}) \in Z_p^\nu$. Next we treat Z_p not only as the underlying set of the group $(\mathbb{Z}_p; +)$ but also as the underlying set of the prime field $GF(p)$. Now, after this identification of A with Z_p^ν, [KK20, Lemma 3.2] allows to simulate k-ary polynomials of the algebra $\mathbf{A}$ by the $k\nu$-ary polynomials of the field $GF(p)$. In particular for the k-ary polynomial $\mathbf{p}(x_1, \ldots, x_k)$ from our program $(\mathbf{p}, n, \iota, \{c\})$ there is a polynomial $w'(x_1^{(1)}, \ldots, x_1^{(\nu)}, \ldots, x_k^{(1)}, \ldots, x_k^{(\nu)})$ over the field $GF(p)$ with degree bounded by $|A|^{1 + \log_p \mu(\mathbf{A})}$, such that for all $(a_1, \ldots, a_k) \in A^k$ we have

$$\mathbf{p}(a_1, \ldots, a_k) = c \quad \text{iff} \quad w'(a_1^{(1)}, \ldots, a_1^{(\nu)}, \ldots, a_k^{(1)}, \ldots, a_k^{(\nu)}) = 1.$$

Now to construct an $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)}$-circuit computing the function $(\mathbf{p})[\iota, c](b_1, \ldots, b_n) : \{0, 1\}^n \longrightarrow \{0, 1\}$ we first modify the $k\nu$-ary polynomial w' to n-ary polynomial $w(\mathsf{b}_1, \ldots, \mathsf{b}_n)$, again over the field $GF(p)$ and again of degree $d \leqslant |A|^{1 + \log_p \mu(\mathbf{A})}$. We simply replace the variables $x_i^{(1)}, \ldots, x_i^{(\nu)}$, by affine transformations of the corresponding boolean variable b^{x_i} (determined by the program instruction $\iota(x_i) = (\mathsf{b}^{x_i}, a_0^{x_i}, a_1^{x_i})$ for the variable x_i of $\mathbf{p}$) as follows:

$$x_i^{(1)} \quad \text{by} \qquad\qquad \mathsf{b}^{x_i} \cdot \left((a_1^{x_i})^{(1)} - (a_0^{x_i})^{(1)}\right) + (a_0^{x_i})^{(1)},$$

$$\vdots$$

$$x_i^{(\nu)} \quad \text{by} \qquad\qquad \mathsf{b}^{x_i} \cdot \left((a_1^{x_i})^{(\nu)} - (a_0^{x_i})^{(\nu)}\right) + (a_0^{x_i})^{(\nu)}.$$

It should be easy to observe that then for $\bar{b} = (b_1, \ldots, b_n) \in \{0, 1\}^n$ we have

$$(\mathbf{p})[\iota, c](\bar{b}) = 1 \quad (\text{i.e. } (\mathbf{p})[\iota](\bar{b}) = c) \quad \text{iff} \quad w(\bar{b}) = 1$$

so that we are done. $\qquad\square$

Theorem 9.7 (Theorem 1.4 restated). *Let $\mathbf{A}$ be a finite nilpotent Malcev algebra with $\mathsf{char}\{0, \kappa\} = \{p\}$. 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 $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)/p} \circ \mathsf{MOD}_p$-circuits of size $O(\ell^c)$ with $d \leqslant |A|^{1 + \log \mu(\mathbf{A})}$ and the degree c bounded by $O(|A| \cdot \mu(\mathbf{A}))^{O(\log |A|)}$.*

Proof. As in the proof of Lemma 9.6 we additionally assume that $|S| = 1$. After producing $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)/p} \circ \mathsf{MOD}_p$-circuits separately for each $c \in S$ we join them by $\mathsf{OR}_{|S|}$-gate to get $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)/p} \circ \mathsf{MOD}_p \circ \mathsf{OR}_{|S|}$-circuit. But now Lemma 9.4 allows us to completely eliminate $\mathsf{OR}_{|S|}$-gate by increasing the size of the resulting $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(A)/p} \circ \mathsf{MOD}_p$-circuit from λ to $\lambda^{p \cdot |A|}$. Thus, from now on we assume that $S = \{c\}$.

The first of the following two claims shows that $\mathbf{A}/\sigma_p$ is supernipotent, while the second one that $\delta\left(A/\sigma_p\right) = \delta\left(A\right)/p$.

(9.1) $\kappa \leqslant \sigma_p \leqslant \sigma$,
(9.2) $p \notin \mathsf{char}\{\sigma_p, 1\}$.

The second inequality in (9.1) is obvious. For the first one note that our assumption $\mathsf{char}\{0, \kappa\} = \{p\}$ tells that there is no other prime than p below κ. Since σ_p is the largest congruence with this property it must be over κ. To see (9.2) first note that the congruence lattice of the supernilpotent algebra $\mathbf{A}/\sigma_p$ is a PUPI , so that each characteristic occuring in $\mathbf{A}/\sigma_p$ has to occur in $\mathsf{Con}\,\mathbf{A}$ between σ_p and one of its successors. But σ_p has no successors of characteristic p.

Now Lemma 9.6 supplies us with a $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(\mathbf{A}/\sigma_p)}$-circuit computing the boolean function of the quotient program $(\mathbf{p}/\sigma_p, n, \iota/\sigma_p, c/\sigma_p)$.

The essential part of the proof is to go down along the chain of congruences $\sigma_p = \gamma_h \succ \gamma_{h-1} \succ \ldots \succ \gamma_1 \succ \gamma_0 = 0$ to consecutively produce $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(\mathbf{A}/\sigma_p)} \circ \mathsf{MOD}_p$-circuit computing the boolean function of the quotient program $(\mathbf{p}/\gamma_j, n, \iota/\gamma_j, c/\gamma_j)$. This in turn will be possible after showing the following claim:

(9.3) Suppose m is a square-free positive integer and $p \nmid m$ is a prime. Let β be an abelian congruence of characteristic p in a finite nilpotent Malcev algebra $\mathbf{D}$. Moreover let $(\mathbf{p}, n, \iota, c)$ be a boolean program of length ℓ over $\mathbf{D}$. If all programs of the form $(\mathbf{p}'/\beta, n, \iota/\beta, c'/\beta)$ over the quotient algebra $\mathbf{D}/\beta$, (with $\mathbf{p}'$ ranging over subterms of $\mathbf{p}$ and c' ranging over elements of D) are computable by $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuits of size at most λ then $(\mathbf{p}, n, \iota, c)$ is computable by a $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit of size $O((\lambda \cdot \ell)^{2p^4 |D|\mu(\mathbf{D})})$

With claim (9.3) we are ready to bound the size of the final circuit. We start with $\mathsf{AND}_d \circ \mathsf{MOD}_{\delta(\mathbf{A}/\sigma_p)}$-circuit computing the quotient program $(\mathbf{p}/\sigma_p, n, \iota/\sigma_p, c)$ and supplied by Lemma 9.6. Its size is bounded by $\lambda_0 \leqslant O(\ell^d)$ with $d \leqslant (2 \cdot \mu\left(\mathbf{A}\right))^{\log|A|}$. Now we go down along the chain of congruences $\sigma_p = \gamma_h \succ \gamma_{h-1} \succ \ldots \succ \gamma_1 \succ \gamma_0 = 0$ and use (9.3) to get the circuit computing $(\mathbf{p}, n, \iota, c)$. This requires $h \leqslant |A|$ iterations so that we end up with a circuit of size $O((\ldots((\lambda_0 \cdot \ell)^a \cdot \ell^a) \ldots \cdot \ell)^a) \leqslant (\lambda_0 \cdot \ell)^{a^{h+1}})$ where $a \leqslant 2p^4 |A| \mu\left(\mathbf{A}\right) \leqslant |A|^6 \mu\left(\mathbf{A}\right)$. One can combine all these inequalities to get the bound $O(\ell^c)$ with $c \leqslant 2|A| \mu\left(\mathbf{A}\right)^{13 \log|A|}$. To go from circuit for $(\mathbf{p}, n, \iota, c)$ to the one for $(\mathbf{p}, n, \iota, S)$, with arbitrary $S \subseteq A$, Lemma 9.4 increases the size to $O((\ell^c)^{|A|^2})$, but the exponent of ℓ is still bounded by $O(|A| \cdot \mu\left(\mathbf{A}\right))^{O(\log|A|)}$, as required.

Now we return to the proof of claim (9.3) and show how to construct the circuit for $(\mathbf{p}, n, \iota, c)$ from the circuits over $\mathbf{D}/\beta$ for the subterms of $\mathbf{p}$. To start this construction we note that (due to [FM87, Exercise 7.3]) the atomic congruence β lies below the center of $\mathbf{D}$ so that, following the proof of [FM87, Theorem 7.1], we can represent algebra $\mathbf{D}$ in the following way:

- the universe of $\mathbf{D}$ is a direct product $M \times D'$, where
    - $D' = D/\beta$, and
    - M is a β-coset of $\mathbf{D}$ (and therefore, by Lemma 7.2, it is polynomially equivalent to a simple module over a subring R of the ring $\mathrm{End}(\mathbb{Z}_p^\nu)$ of all endomorphisms of the abelian group $\mathbb{Z}_p^\nu$),
- each basic operation f of $\mathbf{D}$ is determined by a sequence of scalars $\alpha_1^f, \ldots, \alpha_{\mu(f)}^f \in R$ and a function $\widehat{f} : (D')^{\mu(\mathbf{F})} \longrightarrow M$ in the following way

$$(2) \qquad f^{\mathbf{D}}(d_1, \ldots, d_{\mu(f)}) = \left(\sum_{i=1}^{\mu(f)} \alpha_i^f \cdot d_i^M + \widehat{f}(d_1^{D'}, \ldots, d_{\mu(f)}^{D'}), f^{\mathbf{D}'}(d_1^{D'}, \ldots, d_{\mu(f)}^{D'})\right)$$

whenever $d_i = (d_i^M, d_i^{D'}) \in M \times D'$.

Actually for a more complex polynomial $\mathbf{p}(\overline{x}) = f(\mathbf{p}_1(\overline{x}), \ldots, \mathbf{p}_{\mu(f)}(\overline{x}))$, where f is a basic operation and $\mathbf{p}_1, \ldots, \mathbf{p}_{\mu(f)}$ are polynomials the above representation still holds, i.e.,

$$\mathbf{p}^{\mathbf{D}}(\overline{d}) = \left(\sum_{i=1}^{\mu(\mathbf{p})} \alpha_i^{\mathbf{p}} \cdot d_i^M + \widehat{\mathbf{p}}(d_1^{D'}, \ldots, d_{\mu(\mathbf{p})}^{D'}), \mathbf{p}^{\mathbf{D}'}(d_1^{D'}, \ldots, d_{\mu(\mathbf{p})}^{D'}) \right).$$

However to understand the function $\widehat{\mathbf{p}} : (D')^{\mu(\mathbf{p})} \longrightarrow M$, given for $\overline{d'} \in (D')^{\mu(\mathbf{p})}$ by $\widehat{\mathbf{p}}(\overline{d'}) = \sum_{i=1}^{\mu(f)} \alpha_i^f \cdot \widehat{\mathbf{p}}_i(\overline{d'}) + \widehat{f}(\mathbf{p}_1^{\mathbf{D}'}(\overline{d'}), \ldots, \mathbf{p}_{\mu(f)}^{\mathbf{D}'}(\overline{d'}))$ we need to recursively understand the $\widehat{\mathbf{p}}_i$'s. This in turn requires to go deeper into the structure of how the polynomial $\mathbf{p}$ [or corresponding circuit] is build from the basic operations [or nodes, respectively]. Thus for a basic operation f [or a node in a circuit labeled by f] we form the set N_f of all occurrences of this symbol [node]. Each such occurrence of f, say $f^{(j)}$ with $j \in N_f$ has its inputs determined by the polynomials/nodes $\mathbf{p}_1^{(j)}, \ldots, \mathbf{p}_{\mu(f)}^{(j)}$. It may happen that two occurrences, say $f^{(j)}$ and $f^{(j')}$ have the same inputs $(\mathbf{p}_1^{(j)}, \ldots, \mathbf{p}_{\mu(f)}^{(j)}) = (\mathbf{p}_1^{(j')}, \ldots, \mathbf{p}_{\mu(f)}^{(j')})$. We identify them by the equivalence relation $j \sim j'$. Note here that the number $\sum_{f \text{ occuring in } \mathbf{p}} |N_f|$ determines the size of the polynomial $\mathbf{p}$, while $\sum_{f \text{ occuring in } \mathbf{p}} |N_f|/\sim$ is (the lower bound for) the size of the corresponding algebraic circuit. Now, going recursively down with the formula (2) along the construction of the polynomial/circuit from the basic operations we end up with

$$\widehat{\mathbf{p}}(\overline{d'}) = \sum_f \sum_{j \in N_f} \beta_f^{(j)} \cdot \widehat{f}(\mathbf{p}_1^{(j)}(\overline{d'}), \ldots, \mathbf{p}_{\mu(f)}^{(j)}(\overline{d'})),$$

where the $\beta_f^{(j)}$'s are algebraic combinations of appropriate coefficients of the form α_i^g's for the basic operations g's. Since the values $\widehat{f}(\mathbf{p}_1^{(j)}(\overline{d'}), \ldots, \mathbf{p}_{\mu(f)}^{(j)}(\overline{d'}))$ are the same for the j's from one $\sim$-class we can sum corresponding $\beta_f^{(j)}$'s to shorten our expression appropriately (this may be important for the size of the circuit we are going to construct).

After this preparatory algebraic work we return to our construction of the boolean $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit computing the boolean function of the program $(\mathbf{p}, n, \iota, c)$. Note first that for $\overline{b} \in \{0,1\}^n$ we have $(\mathbf{p})[\iota, c](\overline{b}) = 1$ iff $(\mathbf{p})[\iota](\overline{b})$ coincides with c on both coordinates: on M and on D', or in other words iff $(\mathbf{p}/\beta)[\iota/\beta, c/\beta](\overline{b}) = 1$ and $(p^M)[\iota, c^M](\overline{b}) = 1$, for the quotient program $(\mathbf{p}/\beta, n, \iota/\beta, c/\beta)$ and the p^M-program (p^M, n, ι, c^M) with the function $p^M : D^k \longrightarrow M$ determined by $p^M(d_1, \ldots, d_k) = (\mathbf{p}(d_1, \ldots, d_k))^M$, respectively. Thus the circuit computing $(\mathbf{p})[\iota, c]$ will be obtained first by gluing (by the binary AND-gate) the $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuits for $(\mathbf{p}/\beta)[\iota/\beta, c/\beta]$ and $(p^M)[\iota, c^M]$ and then eliminating this binary AND-gate with the help of Lemma 9.4.

The circuit computing the function $(\mathbf{p}/\beta)[\iota/\beta, c/\beta]$ is supplied by our recursion hypothesis for the program $(\mathbf{p}/\beta, n, \iota/\beta, c/\beta)$. However the circuit for $(p^M)[\iota, c^M]$ requires more work. Our previous considerations combined with the instructions $\iota(x_i) = (\mathsf{b}^{x_i}, a_0^{x_i}, a_1^{x_i})$ allow us to compute the projection of $(p^M)[\iota](\overline{b})$ onto M to be

$$(p^M)[\iota](\overline{b}) = \sum_{i=1}^{\mu(\mathbf{p})} \alpha_i^{\mathbf{p}} \cdot (a_{\mathsf{b}^{x_i}}^{x_i})^M + \sum_f \sum_{j \in N_f} \beta_f^{(j)} \cdot \widehat{f}\left(\mathbf{p}_1^{(j)}(\ldots, (a_{\mathsf{b}^{x_i}}^{x_i})^{D'}, \ldots), \ldots, \mathbf{p}_{\mu(f)}^{(j)}(\ldots, (a_{\mathsf{b}^{x_i}}^{x_i})^{D'}, \ldots) \right)$$

or in other words

$$(p^M)[\iota](\overline{b}) = \sum_{i=1}^{\mu(\mathbf{p})} \alpha_i^{\mathbf{p}} \cdot ((x_i)[\iota](\overline{b}))^M + \sum_f \sum_{j \in N_f} \beta_f^{(j)} \cdot \widehat{f}\left(\left(\left(\mathbf{p}_1^{(j)}\right)[\iota](\overline{b})\right)^{D'}, \ldots, \left(\left(\mathbf{p}_{\mu(f)}^{(j)}\right)[\iota](\overline{b})\right)^{D'} \right),$$

where $(x_i)[\iota]$ is the inner function determined by applying instruction ι to the polynomial $\mathbf{q}_i(x_1, \ldots, x_k) = x_i$. By Lemma 9.1, the summands $\alpha_i^{\mathbf{P}} \cdot ((x_i)[\iota](\overline{b}))^M$, as (essentially) the functions of the form $\{0,1\} \longrightarrow M$, can be computed by $\Sigma_{(p,k)}$-circuits of size $O(1)$, so that also their sum can be computed by $\mu(\mathbf{p})$-ary $\Sigma_{(p,k)}$-circuits of size $O(1)$. To represent the other

summands in the above display by $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p \circ \mathsf{AND}_{|D'|} \circ \Sigma_{(p,k)}$-circuits we first replace

$$\widehat{f}\left(\left(\left(\mathbf{p}_1^{(j)}\right)[\iota]\,(\bar{b})\right)^{D'},\ldots,\left(\left(\mathbf{p}_{\mu(f)}^{(j)}\right)[\iota]\,(\bar{b})\right)^{D'}\right)$$

by

$$\widehat{f^\circ}\left(\left(\mathbf{p}_1^{(j)}\right)[\iota/\beta, d_1/\beta]\,(\bar{b}),\ldots,\left(\mathbf{p}_1^{(j)}\right)[\iota/\beta, d_s/\beta]\,(\bar{b}),\ldots,\right.$$
$$\left.\left(\mathbf{p}_{\mu(f)}^{(j)}\right)[\iota/\beta, d_1/\beta]\,(\bar{b}),\ldots,\left(\mathbf{p}_{\mu(f)}^{(j)}\right)[\iota/\beta, d_s/\beta]\,(\bar{b}),\ldots\right)$$

where $d_1,\ldots,d_s$ is the transversal of the quotient D/β. By induction hypothesis each of the functions $\left(\mathbf{p}_i^{(j)}\right)[\iota/\beta, d_l/\beta] : \{0,1\}^n \longrightarrow D'$ is computable by $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit of size $O(\lambda)$. On the other hand, again by Lemma 9.1, the function $\widehat{f^\circ} : \{0,1\}^{|D'|\cdot\mu(f)} \longrightarrow M$ is computable by $\mathsf{AND}_{|D'|\cdot\mu(f)} \circ \Sigma_{(p,k)}$-circuit of size $O(2^{\mu(\mathbf{D})\cdot s})$, or simply $O(1)$ as it does not depend on λ or ℓ. Summing all those $\ell = \sum_{f\text{ occuring in }\mathbf{p}} |N_f|$ summands (with appropriate scalars) we end up with $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p \circ \mathsf{AND}_{|D'|\cdot\mu(f)} \circ \Sigma_{(p,k)}$-circuit of size $O(\lambda\cdot\ell)$ computing $(p^M)[\iota]$. Now Lemma 9.5 allows us to replace the 5-level circuit, in which $\Sigma_{(p,k)}$ is replaced by $\Sigma_{(p,k)}^{c^M}$, computing $(p^M)[\iota, c^M]$ by a $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit of size $O((\lambda\cdot\ell)^{\nu p^3|D'|\mu(\mathbf{D})})$. Finally Lemma 9.4 allows us to glue, by binary AND, this circuit with the $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-circuit for $(\mathbf{p}/\beta)[\iota/\beta, c/\beta]$ (of size $O(\lambda)$) to end up with a circuit of size $O((\lambda\cdot\ell)^{2\nu p^4|D'|\mu(\mathbf{D})})$ computing $(\mathbf{p})[\iota, c]$ in $\mathbf{D}$.

Note that $|D| = |D'|\cdot p^\nu$ so that $\nu\cdot|D'| \leqslant |D|$ and consequently the size of the final circuit can be bounded by $O((\lambda\cdot\ell)^{2p^4|D|\mu(\mathbf{D})})$, as claimed in (9.3). $\square$

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 (9.3), i.e. from the hight of the congruence lattice of $\mathbf{A}$. The $(|A|\cdot\mu(\mathbf{A}))$-part is a consequence of arity of a binary expansion $f^\circ : \{0,1\}^{|A|\cdot\mu(f)} \longrightarrow A$ 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 $\mathrm{CEQV}(\mathbf{A})$ 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 $\mathrm{CSAT}/\mathrm{CEQV}$ [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 $\mathsf{AND}_d \circ \mathsf{MOD}_m \circ \mathsf{MOD}_p$-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 Hyphothesis.*

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 $\mathrm{CSAT}(\mathbf{A})$ is in P, while $\mathrm{CSAT}(\mathbf{A}/\sigma)$ 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 PROGCSAT($\mathbf{A}/\sigma$) implies the hardness for 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 Therien. 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$_p$-MOD$_m$) circuits. *Discrete Mathematics & Theoretical Computer Science*, 4(2):247–254, 2001.

[GT00]   Vince Grolmusz and Gábor Tardos. Lower bounds for (MOD$_p$-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 $\mathbf{A}_4$. *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]   Pawel 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 Kawałek, 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]   Pawel 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]   Pawel 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]   Pawel 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.