Even Faster Algorithms for CSAT Over supernilpotent Algebras

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

Jacek Krzaczkowski
Maria Curie-Sklodowska University, Faculty of Mathematics, Physics and Computer Science, Department of Computer Science, Lublin, Poland
krzacz@poczta.umcs.lublin.pl

Abstract

Recently, a few papers considering the polynomial equation satisfiability problem and the circuit satisfiability problem over finite supernilpotent algebras from so called congruence modular varieties were published. All the algorithms considered in these papers are quite similar and rely on checking a not too big set of potential solutions. Two of these algorithms achieving the lowest time complexity up to now, were presented in [1] (algorithm working for finite supernilpotent algebras) and in [5] (algorithm working in the group case). In this paper we show a deterministic algorithm of the same type solving the considered problems for finite supernilpotent algebras which has lower computational complexity than the algorithm presented in [1] and in most cases even lower than the group case algorithm from [5]. We also present a linear time Monte Carlo algorithm solving the same problem. This, together with the algorithm for nilpotent but not supernilpotent algebras presented in [17], is the very first attempt to solving the circuit satisfiability problem using probabilistic algorithms.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Problems, reductions and completeness; Mathematics of computing → Combinatorial algorithms; Mathematics of computing → Probabilistic algorithms; Computing methodologies → Equation and inequality solving algorithms

Keywords and phrases circuit satisfiability, solving equations, supernilpotent algebras, satisfiability in groups

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.55

Funding The project is partially supported by Polish NCN Grant # 2014/14/A/ST6/00138.

1 Introduction

Solving equations is one of the most important mathematical problems with applications in many areas. We are interested in the computational complexity of the equation satisfiability problem for a fixed finite algebra. The most-studied version of this problem (called PolSat($\mathbf{A}$)) asks if a given equation of polynomials over a fixed algebra $\mathbf{A}$ has a solution or not. There is a number of papers which characterized algebras for which this problem is tractable in polynomial time and for which it is hard in terms of some well-established complexity assumptions (i.e. $\mathsf{P} \neq \mathsf{NP}$). Most of these papers consider only well-known structures like groups [8], [15], [12], [13], [5], [6], rings [12], [20] or lattices [25]. However there is a number of papers considering more general cases e.g. [10], [9], [1]. A new look on the problem was proposed in [19]. This paper was the first systematic study on solving equations in quite general setting. The authors of [19] decided to allow a more compact representation of polynomials on the input of the problem, so to represent them as multi-valued circuits. It leads to the following definition of the problem:

CSAT($\mathbf{A}$) given a circuit over $\mathbf{A}$ with two output gates g_1, g_2 is there a valuation of input gates $\overline{x}$ that gives the same output on g_1, g_2, i.e. $g_1(\overline{x}) = g_2(\overline{x})$.

Under this definition, the computational complexity of CSAT($\mathbf{A}$) depends only on the polynomial clone of $\mathbf{A}$ and, in consequence, can be characterized in terms of algebraic properties of $\mathbf{A}$.

Several articles considering this new approach to solving equations have appeared e.g. [23], [16], [1], [21], [24], [17]. In this paper, we also study CSAT problems. However, for clarity we mention that all the algorithms and upper bounds presented here apply also to POLSAT since polynomials can be represented by circuits expanding the size of the representation only by a constant factor.

Algebras generating congruence modular varieties form a wide class of algebras containing among others many popular algebraic structures like groups, rings and lattices. We will call this class of algebras CM for short. Analyzing a partial characterization of the computational complexity of CSAT for algebras from CM presented in [19] and also the results of [16], [24] and [17], we can see the truly rich world in which one can find problems of different complexities: NP-complete problems, problems contained in P and those, that are natural candidates for NP-intermediate problems. Surprisingly, there are only three known essentially different polynomial time algorithms solving CSAT over algebras from congruence modular varieties. Two of them are black-box algorithms i.e. algorithms which treat circuits as a black-box and try to find the solution by checking some not too big set of potential solutions (so called hitting set). One of them originally was proposed for nilpotent groups [8] and the second one works for distributive lattices [25]. The third of the algorithms mentioned above solves CSAT by inspecting some kind of normal form of a given circuit but it seems that the usefulness of such kind of algorithm is limited to so called 2-step supernilpotent algebras [16], [17].

In this paper we consider supernilpotent algebras from CM which are natural generalizations of nilpotent groups (among all groups only those nilpotent ones induce tractable problems, assuming $\mathsf{P} \neq \mathsf{NP}$). Every such a supernilpotent algebra $\mathbf{A}$ decomposes into a direct product of supernilpotent algebras of prime power order. That is why we can reduce the problem of solving equations over $\mathbf{A}$ to a fixed number (at most $\log|A|$) of similar problems over supernilpotent algebras, but this time of prime power order. This Turing reduction can be performed in linear time and thus we will only be looking for an algorithm for solving equations over supernilpotent algebras of prime power order.

We slightly modify the algorithm that was applied in the group setting. In this algorithm we check potential solutions in which at most d variables are assigned to some non-zero value. It was introduced by Goldmann and Russell in [8] for nilpotent groups and its correctness was reproved by Horváth in [12]. In both cases, it was shown that the considered algorithm works in polynomial time but the degree of the polynomial came from an application of Ramsey Theory and was really huge. Later, it was independently shown in [23] and [19] that essentially the same algorithm works for supernilpotent algebras from a congruence modular variety in polynomial time with the same huge degree of the polynomial. These results were improved by Aichinger in [1]. In his paper the degree of the polynomial describing the complexity of CSAT($\mathbf{A}$) was bounded by $d = |A|^{\log_2|A|+\log_2 m+1}$, where m is the maximal arity of basic operations of $\mathbf{A}$. Using similar tools as Aichinger and some new ideas we show the following.

▶ **Theorem 1.1.** *Let $\mathbf{A}$ be a supernilpotent algebra of prime power order q^h from a congruence modular variety. Then there exists a black-box algorithm solving CSAT($\mathbf{A}$) in time $O(n^d k)$, where $d = |A|^{\log_q m+1}$, n is the number of input gates in the given circuit, k is the size of the circuit, and m is the maximal arity of basic operations of $\mathbf{A}$.*

The proof of this theorem can be found in Section 5. Note, that after applying Theorem 1.1 to nilpotent groups of prime power order q^h we obtain that $d = |G|^{\log_q 2+1}$. We note here that in [5] A. Földvári, using some group specific tools, gave a different algorithm for the PolSat($\mathbf{A}$) problem of time complexity $O(n^d)$, where $d = \frac{1}{2}|G|^2 \cdot \log|G|$ (here n denotes the input size). So in most cases our algorithm improves this result too (especially when the prime q is big) and it applies to a much more general class of algebras.

It turns out that switching from a deterministic computational model to a probabilistic one we obtain a great improvement. It is shown in the second main result of this paper, which states the following.

▶ **Theorem 1.2.** *Let* $\mathbf{A}$ *be a supernilpotent algebra of prime power order from a congruence modular variety. Then, there exists a linear time Monte Carlo algorithm solving* Csat($\mathbf{A}$).

The surprising corollary we get when we apply Theorem 1.2 to finite groups and use results from [8] and [14].

▶ **Corollary 1.3.** *Let* $\mathbf{G}$ *be a finite group. Then,* Csat($\mathbf{G}$)

- *can be solved by a linear time Monte Carlo algorithm if* $\mathbf{G}$ *is nilpotent,*
- *is NP-complete otherwise.*

To obtain the algorithms mentioned above we study the structure of nilpotent algebras of prime power order. Thanks to deep universal algebraic tools developed in [7] and [26] our study does not contain hard to read technical proofs. The readers not interested in algebraic details can skip Section 3. The readers interested in more systematic and detailed study in this spirit but in more general settings can see [18].

The main conclusion of Section 3 is that solving equations over nilpotent algebras of prime power order q^h can be reduced to solving one special equation of bounded degree between polynomials in the finite field $\mathbf{F}_q$. Thus, in the next sections we do not need the universal algebraic tools and we work with finite fields only.

Our randomized algorithm solving equations over $\mathbf{F}_q$ of low degree is very simple. It turned out that all we need to do is randomly draw solutions with a uniform distribution. In such a way, we obtain a c-correct true-biased algorithm for some constant c depending on the algebra. It works thanks to the nice behavior of polynomial over $\mathbf{F}_q$ of not too high degree. This behavior is described in the following Lemma.

▶ **Lemma 1.4.** *Let* $\mathbf{f}$ *be an* n*-ary polynomial of degree* d *over the finite field* $\mathbf{F}_q$*. Then, for every* $y \in F_q$ *such that* $|\mathbf{f}^{-1}(y)| > 0$ *we have* $|\mathbf{f}^{-1}(y)| \geqslant q^{n-d-q\log_2 q}$.

The lemma is in fact a generalization of the one proven in [11] for the field $\mathbf{F}_2$. However, the method presented there cannot be applied to prove the similar fact in every finite field. Note that if the degree of the polynomial was smaller than the size of the field, then we would just need to apply the famous Schwartz–Zippel lemma to get that the density of solutions among all possible assignments to variables is huge. In our case the degree of the polynomial is bounded by a constant depending on $\mathbf{A}$ and almost always it exceeds the field size we are working with. There are also a number of other results, that can be applied here, introduced for polynomial identity checking of s-sparse polynomials, but they do not lead to linear time algorithms.

The article is organized as follows. The second section contains some definitions and background materials. In Section 3 we present the structure of supernilpotent algebras and show that Csat for such algebras can be reduced to solving equations between polynomials of a bounded degree over a finite field. The proof of Lemma 1.4 is contained in Section 4.

In Sections 5 and 6 we show deterministic and randomized algorithms solving CSAT for supernilpotent algebras and prove Theorem 1.1 and Theorem 1.2. Finally, Section 7 contains remarks regarding the results contained in this paper and conclusions.

2 Background material

In this paper we use the standard notation of universal algebra (see e.g. [4]). An algebra is a structure consisting of a set called universe and a set of finitary operations on it. Groups and fields are obviously examples of algebras. All algebras considered in this paper are finite i.e. with finite universe and finite set of operations. We usually denote algebras using bold capital letters and their universes by the same but non-bold letters. The language or type of algebra is the set $\mathcal{F}$ of function symbols together with non-negative integers assigned to each member of $\mathcal{F}$. We say that an algebra $\mathbf{A} = (A, F)$ is of type $\mathcal{F}$ if the set F of its operations is indexed by elements of $\mathcal{F}$ and for every n-ary function symbol the corresponding operation $f^{\mathbf{A}} \in F$ is also n-ary. We use overlined small letters e.g. $\overline{x}$, $\overline{a}$ to denote tuples of variables or elements of an algebra and the same letters without overline but with subscript to denote elements of tuples e.g. x_i, a_i. Let $\mathbf{A} = (A, F)$ be an algebra.

By terms over an algebra $\mathbf{A}$ we mean all proper expressions in language of $\mathbf{A}$ built up from variables and function symbols. Polynomials are expressions in which we additionally admit constants from the algebra's universe. With terms and polynomials we can associate term and polynomial operations in the obvious way. We say that a polynomial $\mathbf{p}(x_1, \ldots, x_{k-1}, z)$ of an algebra $\mathbf{A}$ is a commutator polynomial of rank k iff $\mathbf{p}(a_1, \ldots, a_{k-1}, b) = b$ whenever $b \in \{a_1, \ldots, a_{k-1}\} \subseteq A$ and $\mathbf{p}(a_1, \ldots, a_{k-1}, b) \neq b$ for some $a_1, \ldots, a_{k-1}, b \in A$.

In this paper we consider supernilpotent algebras from congruence modular varieties. The notion of supernilpotency is strongly connected with the modular commutator theory (see [7] for details) and so called higher commutators introduced by A. Bulatov [3] and further developed by E. Aichinger and N. Mudrinski [2]. Since, the definition of supernilpotent algebras is quite technical and requires introducing a bunch of auxiliary notions we decided to omit it. Instead of the definition we will use the following characterization of supernilpotent algebras from a congruence modular varieties which can be easily inferred from the deep work of R. Freese and R. McKenzie [7] and K. Kearnes [22], and has been observed in [2].

▶ **Theorem 2.1.** *For a finite algebra* $\mathbf{A}$ *from a congruence modular variety the following conditions are equivalent:*

1. $\mathbf{A}$ *is supernilpotent,*
2. $\mathbf{A}$ *is nilpotent, decomposes into a direct product of algebras of prime power order and the term clone of* $\mathbf{A}$ *is generated by finitely many operations,*
3. $\mathbf{A}$ *is nilpotent and there exists* k *such that all commutator polynomials have rank at most* k*.*

In the next sections we will see that Theorem 2.1 shows two key properties of supernilpotent algebras: the possibility of decomposition into direct product of algebras of prime power order and bounded essential arity of commutator polynomials. The second property can be formulated in a less formal way that for every supernilpotent algebra there exists k such that no polynomial function behaves similarly to $k+1$-ary conjunction.

We define circuits in a common way as a directed acyclic graphs. Note that every gate is in fact an operation on some domain. Hence, the set of gates with the same domain can be treated as an algebra. In the similar way we can look at a finite algebra as a collection of gates. Thus, every circuit over an algebra $\mathbf{A}$ can be represented as a term of $\mathbf{A}$ (or

polynomial of $\mathbf{A}$ if values on some input gates are fixed). Note that in many cases circuits enable much shorter description of functions than polynomials. The reason is that the value computed in some subcircuit can be used as an input for many gates. In the case of terms/polynomials to use many times the value of some subterm/subpolynomial we have to make many copies of it. It implies that for many algebras (e.g. solvable but non-nilpotent groups) PolSat is easier than Csat. On the other hand, in our case there are no differences in the complexity between PolSat and Csat. The reason is that we consider algorithms which do not analyze the form of the input but use the hitting set which depends only on the number of input gates/variables. Note that computing the value of either circuit or polynomial for any evaluation of input gates/variables can be done in linear time. Thus, in this paper we do not focus on the form of the input but on its algebraic properties.

3 The structure of supernilpotent algebras

In this section we will see that every supernilpotent algebra of prime power order q^h from CM is in fact a wreath product of algebras polynomially equivalent to simple modules of order q^α. In fact, we will see even more. We will prove that every polynomial operation of such an algebra can be described by a bunch of polynomials over $\mathbf{F}_q$ of bounded degree. More detailed investigations of structure of supernilpotent and not only supernilpotent algebras can be found in [18].

First, we present the mentioned decomposition of supernilpotent algebras into a wreath product of algebras that are polynomially equivalent to simple abelian groups. We will use Freese and McKenzie's ideas from [7] developed in more general settings in VanderWerf's PhD thesis [26]. In particular, consider algebras $\mathbf{Q} = (Q, F^{\mathbf{Q}})$ and $\mathbf{B} = (B, F^{\mathbf{B}})$ of the same type $\mathcal{F}$, such that $\mathbf{Q}$ is abelian with the associated group $(Q, +, -)$. Morover let $\mathbf{T}$ be a set of operations such that for every n-ary operation $f \in \mathcal{F}$ there is $t_f : B^n \longmapsto Q$. According to [7] we can define the algebra $\mathbf{A} = \mathbf{Q} \otimes^T \mathbf{B}$ of type $\mathcal{F}$ with universe $Q \times B$ and operations defined as follows

$$f^{\mathbf{A}}((q_1, b_1), \ldots, (q_n, b_n)) = \left(f^{\mathbf{Q}}(q_1, \ldots, q_n) + t_f(b_1, \ldots, b_n), f^{\mathbf{B}}(b_1, \ldots, b_n)\right),$$

where f is an n-ary operation from $\mathcal{F}$. Note that since $\mathbf{Q}$ is an abelian algebra from a congruence modular variety and hence affine $f^{\mathbf{Q}}$ can be expressed in the form $f^{\mathbf{Q}}(q_1, \ldots, q_n) = \sum_{i=1}^{n} \lambda_i q_i + c$, where λ_i's are endomorphisms of $(Q, +)$.

Let $\mathbf{A}$ be a supernilpotent algebra of prime power order q^h and θ be one of its atoms i.e. congruences covering $0_{\mathbf{A}}$. Then, using results from [7] it can be shown that $\mathbf{A}$ can be decomposed into a wreath product of $\mathbf{A}/\theta$ and some algebra $\mathbf{Q}$ polynomially equivalent to a simple module. More precisely $\mathbf{A}$ is isomorphic to the algebra $\mathbf{Q} \otimes^T \mathbf{A}/\theta$ for some T and $\mathbf{Q}$. Note that if $|Q| = q^\alpha$ then $\mathbf{A}/\theta$ has order $q^{h-\alpha}$. Repeating this procedure recursively for $\mathbf{A}/\theta$ we obtain that $\mathbf{A}$ is isomorphic to some algebra which is the wreath product of algebras polynomially equivalent to simple modules of order $p^{\alpha_1}, \ldots, p^{\alpha_s}$. From this point we assume that $\mathbf{A}$ itself is such an algebra. Denote e_i the projection on the i-th coordinate of A (for $i = 1 \ldots s$). Now, unwinding the recursive procedure we get that every basic operation f of

A fulfills the following properties:

$$e_s(f(x_1,\dots,x_n)) = \sum_{i=1}^{n} \lambda_i^s e_s(x_i) + t_f^s,$$

$$\dots$$

$$e_j(f(x_1,\dots,x_n)) = \sum_{i=1}^{n} \lambda_i^j e_j(x_i) + t_f^j(e_{j+1}(x_1),\dots,e_s(x_1),\dots,e_{j+1}(x_n),\dots,e_s(x_n)),$$

for some λ_i^j's being endomorphisms of j-th module (of order p^{α_j}) and some t_f^j's. Note that constant summands in the above expressions are hidden in t_f^j's and t_f^s is just a constant.

Now, we will translate every polynomial $\mathbf{g}$ over $\mathbf{A}$ to a system of polynomials over the field $\mathbf{F}_q$, that will simulate the behaviour of $\mathbf{g}$, From the above observations about wreath products we see that every element $a \in A$ can be written as a tuple $a = (e_1 a.\dots, e_s a)$. Furthermore, each $e_i a$ can be identified with a tuple $b_1,\dots,b_{\alpha_s}$, where each $b_j \in Z_q$. Indeed, each simple module of size q^α has a group reduct of prime exponent. This group must be then isomorphic to the group $\mathbf{Z}_q^\alpha$. So, each element $a \in A$ can be identified in such a way with a tuple $(\pi_1(a),\dots,\pi_h(a))$ (with $\pi_i(a) \in Z_q$) and without loss of generality we will just write $a = (a_1,\dots,a_h)$ (as we can replace the algebra $\mathbf{A}$ with an isomorphic algebra accordingly) or $a = (a_1, a_2.\dots, a_{\alpha_i})$, when $a \in e_i A$.

So now it is clear, that for $i = 1\dots h$ each $\pi_i\mathbf{g}(x_1,\dots,x_n)$ is in fact a function from $(Z_q)^{nh} \longrightarrow Z_q$ and hence it can be represented by multivariate polynomial over variables $\pi_1 x_1,\dots\pi_h x_1,\dots,\pi_1 x_n,\dots\pi_h x_n$. So for each $i = 1\dots h$ we have some polynomial $\mathbf{p}_i$ satisfying $\pi_i\mathbf{g}(x_1,\dots,x_n) = \mathbf{p}_i(\pi_1 x_1,\dots\pi_h x_1,\dots,\pi_1 x_n,\dots\pi_h x_n)$. From basic algebra we know that $\mathbf{p}_i$ has a unique representation up to equations $x^q = x$ (for all variables). We will always mean by polynomial representing $\pi_i\mathbf{g}$ this of the smallest total degree up to those equations. We will also write $\deg \pi_i\mathbf{g}$ for the total degree of the polynomial representing $\pi_i\mathbf{g}$. Now, we want to prove, that such polynomials have small degrees.

▶ **Lemma 3.1.** *Let* $\mathbf{A}$ *be a supernilpotent algebra of prime power order* q^h *from a congruence modular variety and* $\mathbf{g}$ *be an* n*-ary polynomial of* $\mathbf{A}$*. Let* d_i *be the maximal total degree of* $\pi_j\mathbf{g}$ *for* $\alpha_1 + \dots + \alpha_{i-1} < j \leqslant \alpha_1 + \dots + \alpha_{i-1} + \alpha_i$*. Then*

$$\sum_{i=1}^{s} \alpha_i \cdot d_i \leqslant (mq)^{\alpha_1+\dots+\alpha_{s-1}} \cdot \alpha_s$$

where m *is the maximal arity of basic operatios in the signature of* $\mathbf{A}$*.*

Proof. We will inductively decrease $j = s\dots 1$ and consider coordinates of $e_j A$ (there are α_j of them) to obtain the degree of $\pi_j\mathbf{g}$ for $\alpha_1 + \dots + \alpha_{i-1} < j \leqslant \alpha_1 + \dots + \alpha_{i-1} + \alpha_i$. Observe, that from the form of any basic operation of $\mathbf{A}$ that we derived from wreath product representation we can get (by simple induction) that for any n-ary polynomial $\mathbf{g}$ its j-th coordinate $e_j\mathbf{g}$ can be written as a sum of elements of one of the forms:

- $\lambda e_j x_i$, where x_i is a variable and λ is some endomorphism of a module corresponding to $e_j\mathbf{A}$,
- $t_f^j(e_{j+1}\mathbf{g}^{(1)},\dots,e_s\mathbf{g}^{(1)}\dots,\mathbf{e}_{j+1}\mathbf{g}^{(l)}\dots,e_s\mathbf{g}^{(l)})$, where t_f^j comes from the l-ary basic operation f of the algebra $\mathbf{A}$ and $\mathbf{g}^{(i)}$ are other polynomials of $\mathbf{A}$,
- constant.

For $j = s$ we do not have the second type of the above summands. To start with take $j = s$. Then, $e_s A$ is then the underlying set of a module of size q^{α_s}, so it has α_s coordinates.

We want to bound the degree of the polynomial representing $e_s f$ projected to each such coordinate. Notice that $\lambda e_s x_i$ is essentially a unary function, that depends only on projections of x_i to α_s coordinates. Moreover, on each coordinate it must be a linear function, because λ is an endomorphism of an abelian group of exponent q. It means that on each coordinate it can be represented by an polynomial of degree at most 1. So we get that $d_s \leqslant 1$ (because the degree of sum of polynomials is at most the maximal degree of the summands and adding constants does not affect our upper bound).

In case $j < s$ the degrees of polynomials for $\lambda e_j x_i$ are again bounded by 1 and we are left with summands of the form $t_f^j(e_{j+1}\mathbf{g}^{(1)}, \ldots, e_s\mathbf{g}^{(1)}, \ldots, e_{j+1}\mathbf{g}^{(l)}, \ldots, e_s\mathbf{g}^{(l)})$, where l is the arity of the basic operation f. For $u > j$ each $e_u\mathbf{g}^{(v)}$ can be represented by α_u many polynomials of degree at most d_u. Every projection of t_f^j itself can be represented as a polynomial, each of whose variable appears with degree at most $q-1$, so $t_f^j(e_{j+1}\mathbf{g}^{(1)}, \ldots, e_s\mathbf{g}^{(1)}, \ldots, e_{j+1}\mathbf{g}^{(l)}, \ldots, e_s\mathbf{g}^{(l)})$ projected to any of its α_j coordinates can be represented by a polynomial of degree at most $l \cdot (q-1) \cdot \sum_{i=j+1}^{s} \alpha_i d_i$. As m is the maximal arity of basic operations of $\mathbf{A}$ we get:

$$d_j \leqslant m \cdot (q-1) \cdot \sum_{i=j+1}^{s} \alpha_i d_i.$$

Since this holds for any j < s we have that:

$$\sum_{i=1}^{s} \alpha_i d_i = \alpha_1 d_1 + \sum_{i=2}^{s} \alpha_i d_i \leqslant \alpha_1 \cdot m \cdot (q-1) \cdot \sum_{i=2}^{s} \alpha_i d_i + \sum_{i=2}^{s} \alpha_i d_i = ((q-1)m\alpha_1 + 1)(\sum_{i=2}^{s} \alpha_i d_i).$$

As $(q-1)m\alpha_1 + 1 \leqslant (qm)^{\alpha_1}$ we get

$$\sum_{i=1}^{s} \alpha_i d_i \leqslant (qm)^{\alpha_1} \cdot (\sum_{i=2}^{s} \alpha_i d_i)$$

and applying the same reasoning recursively to $\sum_{i=j}^{s} \alpha_i d_i$ for $j = 2, 3, \ldots, s$ we will end up

$$\sum_{i=1}^{s} \alpha_i d_i \leqslant (qm)^{\alpha_1}(qm)^{\alpha_2} \cdot \ldots \cdot (qm)^{\alpha_{s-1}} \alpha_s d_s = (qm)^{\alpha_1 + \ldots + \alpha_{s-1}} \cdot \alpha_s$$

which is what we wanted to prove. ◀

Lemma 3.1 shows in fact how to reduce solving one equation over a supernilpotent algebra $\mathbf{A}$ of prime power order q^h to a system of h equations over the field $\mathbf{F}_q$. Now, we would like to reduce solving one equation over $\mathbf{A}$ to solving one equation of the from $\mathbf{p}(\overline{x}) = 1$, where $\mathbf{p}$ is a bounded degree polynomial over the field $\mathbf{F}_q$. Moreover, the lemma shows that there is an easy to compute one to one mapping between the solutions of the new equation and the original one.

▶ **Lemma 3.2.** *Let* $\mathbf{A}$ *be a supernilpotent algebra of prime power order* q^h *from a congruence modular variety and* m *be the maximal arity of basic operations of* $\mathbf{A}$*. Then, for* n*-ary polynomials* $\mathbf{p}$ *and* $\mathbf{g}$ *over* $\mathbf{A}$*, there exists an* nh*-ary polynomial* $\mathbf{f}$ *over* $\mathbf{F}_q$ *of degree at most* $|A|^{\log_q m+1}$*, such that* $f(F_q^{hn}) \subseteq \{0,1\}$ *and for* $\overline{a} \in A^n$

$$\mathbf{p}(a_1, \ldots, a_n) = \mathbf{g}(a_1, \ldots, a_n)$$

iff

$$\mathbf{f}(\pi_1 a_1, \ldots, \pi_h a_1, \ldots, \pi_1 a_n, \ldots, \pi_h a_n) = 1.$$

Proof. Let

$$\mathbf{p}(x_1, \ldots, x_n) = \mathbf{g}(x_1, \ldots, x_n) \tag{1}$$

be an equation over $\mathbf{A}$. Note that every polynomial of $\mathbf{A}$ projected by every π_i can be represented by a polynomial over the field $\mathbf{F}_q$. So, naturally we can write our equations equivalently as system of h polynomial equations:

$$\begin{cases} \mathbf{p}_1(\pi_1 x_1, \ldots, \pi_h x_n) = 0 \\ \mathbf{p}_2(\pi_1 x_1, \ldots, \pi_h x_n) = 0 \\ \ldots \\ \mathbf{p}_h(\pi_1 x_1, \ldots, \pi_h x_n) = 0 \end{cases} \tag{2}$$

It is easy to see that the function defined as follows

$$\mathbf{f}(\overline{x}) = \prod_{i=1}^{h} (1 - \mathbf{p}_i(\overline{x})^{q-1}) \tag{3}$$

fulfills the conditions of the Lemma. It remains to determine the degree of $\mathbf{f}$. As α_j of those polynomials have the degree bounded by d_j for $j = 1 \ldots s$, we get that degree of $\mathbf{f}$ is bounded by $(q-1)(\sum_{i=1}^{s} \alpha_i d_i)$. So by Lemma 3.1 as $q^{\alpha_1 + \ldots + \alpha_s} = |A|$ this is bounded by

$$(q-1) \cdot (mq)^{\alpha_1 + \ldots + \alpha_{s-1}} \cdot \alpha_s \leqslant (mq)^{\alpha_1 + \ldots, + \alpha_s} = (q^{\log_q m + 1})^{\alpha_1 + \ldots, + \alpha_s} = |A|^{\log_q m + 1}$$ ◀

4 Behavior of polynomials over finite fields

This section contains the proof of Lemma 1.4. The main idea of the proof is to show that a given polynomial over a finite field can be transformed into some special polynomial of known degree. The way we do this transformation allows us to establish a lower bound of the given polynomial's degree depending among other, on the inverse image of the chosen element of the field. Hence, by elementary calculations we obtain that the statement of the lemma holds.

Let $\mathbf{f}$ be a n-ary polynomial over the field $\mathbf{F}_q$ for some prime q. We will prove that for every $y \in \mathbf{f}(F_q^n)$ we have that $|\mathbf{f}^{-1}(y)| > q^{n - \deg \mathbf{f} - q \log_2 q}$. Since for a constant polynomial this is obviously true, we assume that $\mathbf{f}$ is not constant. Fix $y \in \mathbf{f}(F_q^n)$. We will construct a sequence of at most n polynomials of decreasing arity, such that:

- $\mathbf{f}_0 = \mathbf{f}$,
- the arity of $\mathbf{f}_i$ is $n - i$,
- $\frac{|\mathbf{f}_i^{-1}(y)|}{c} \geqslant |\mathbf{f}_{i+1}^{-1}(y)| > 0$, where $c \in \{2, q\}$,
- the polynomial $\mathbf{f}_{i+1}$ is obtained by substituting some variable in $\mathbf{f}_i$ by a constant or a linear combination of other variables,
- if $\mathbf{f}_l$ is the last polynomial in the sequence, then either $|\mathbf{f}_l^{-1}(y)| = 1$ or $\mathbf{f}_l$ is a polynomial in one variable.

We start with the definition of the sequence $\{\mathbf{f}_i\}_{i=0}^{l}$. Let $\mathbf{f}_0 = \mathbf{f}$. If the arity of $\mathbf{f}_i$ is higher than 1 and $|\mathbf{f}_i^{-1}(y)| > 1$, then we define $\mathbf{f}_{i+1}$ in one of two ways depending on the size of $|\mathbf{f}_i^{-1}(y)| > 1$. If $1 < |\mathbf{f}_i^{-1}(y)| < q^q$, then there exists $\overline{a}, \overline{b} \in \mathbf{f}_i^{-1}(y)$, such that $\overline{a} \neq \overline{b}$. Since $\overline{a}$ and $\overline{b}$ are not equal we can choose j such that $a_j \neq b_j$. Without loss of generality assume that $j = n - i$. Now we obtain $\mathbf{f}_{i+1}$ from $\mathbf{f}_i$ by substituting the variable x_{n-i} by some constant $c \in Z_q$. We choose the value c to minimize $|\mathbf{f}_{i+1}^{-1}(y)|$, but to keep $|\mathbf{f}_{i+1}^{-1}(y)| > 0$.

Note that there are at least two possible values for c preserving $|\mathbf{f}_{i+1}^{-1}(y)| > 0$, namely a_{n-i} and b_{n-i}. So $1 \leqslant |\mathbf{f}_{i+1}^{-1}(y)| \leqslant \frac{|\mathbf{f}_i^{-1}(y)|}{2}$. Moreover, it is easy to see that $\deg \mathbf{f}_i \geqslant \deg \mathbf{f}_{i+1}$.

Case $|\mathbf{f}_i^{-1}(y)| \geqslant q^q$ is a bit more complicated since we want to reduce the size of $\mathbf{f}_i^{-1}(y)$ faster than in the previous case. As $|\mathbf{f}_i^{-1}(y)| \geqslant q^q$ we can find q elements of $\mathbf{f}_i^{-1}(y)$, say v^1, v^2,..., v^q, which treated as a vectors over the field $\mathbf{F}_q$ are linearly independent. Hence, there exists $(0,\ldots,0) \neq (\beta_1,\ldots,\beta_{n-i}) \in F_q^{n-i}$, such that for every $a \in F_q$ there exists k such that

$$\sum_{j=1}^{n-i} \beta_j \cdot v_j^k = a.$$

Since, v^j's are taken from $\mathbf{f}_i^{-1}(y)$ it follows that for every $a \in F_q$ the system of equations

$$\begin{cases} \mathbf{f}_i(\overline{x}) = y \\ \sum_{j=1}^{n-i} \beta_j \cdot x_j = a \end{cases}$$

has a solution. Denote the set of solutions of such a system of equations as S_a. Let u be such that $\beta_u \neq 0$. Assume without loss of generality, that $u = n-i$. We choose $b \in F_q$ which minimizes the size of S_b and produce $\mathbf{f}_{i+1}$ by substituting in $\mathbf{f}_i$ the variable x_{n-i} with

$$\beta_{n-i}^{-1}(b - \sum_{j=1}^{n-i-1} \beta_j \cdot x_j).$$

Note that $\sum_{a \in F_q} |S_a| = |\mathbf{f}_i^{-1}(y)|$ and hence $|S_b| \leqslant \frac{|\mathbf{f}_i^{-1}(y)|}{q}$. Thus, $|\mathbf{f}_{i+1}^{-1}(y)| \leqslant \frac{|\mathbf{f}_i^{-1}(y)|}{q}$. Besides, $\deg \mathbf{f}_i \geqslant \deg \mathbf{f}_{i+1}$. It is easy to see that the sequence of polynomials constructed in the presented way fulfills the required conditions.

Now, we will prove that $\deg \mathbf{f} \geqslant n-l$. There are two cases: $\mathbf{f}_l$ is a polynomial in one variable and $|\mathbf{f}_l^{-1}(y)| = 1$. If $\mathbf{f}_l$ is an univariate polynomial then $n-l=1$ and since $\mathbf{f}$ is not a constant polynomial $\deg f \geqslant n-l$. The case when $|\mathbf{f}_l^{-1}(y)| = 1$ is a bit more complicated. Notice, that there is exactly one tuple $\overline{a} = (a_1,\ldots,a_{n-l}) \in F_q^{n-l}$ such that $\mathbf{f}_l(\overline{a}) = y$. Let

$$\mathbf{f}'(x_1,\ldots,x_{n-l}) = 1 - (\mathbf{f}_l(x_1+a_1,\ldots,x_{n-l}+a_{n-l}) - y)^{q-1}.$$

One can easily check that $\mathbf{f}'(\overline{x}) = 1$ iff $x = (0,\ldots,0)$ and otherwise it is equal zero. Obviously $\deg \mathbf{f}' \leqslant (q-1)\deg \mathbf{f}_l$. On the other hand, we can express $\mathbf{f}'$ in the following way:

$$\mathbf{f}'(x_1,\ldots,x_{n-l}) = \prod_{i=1}^{n-l}(1 - x_i^{q-1}).$$

The above polynomial has degree $(q-1)\cdot(n-l)$. This is the lowest possible degree as every polynomial over the field $\mathbf{F}_q$ has a unique representation as a sum of monomials modulo identities in the form $x_i^q = x_i$. Hence, $(q-1)\deg \mathbf{f}_l \geqslant \deg \mathbf{f}' \geqslant (q-1)(n-l)$ and, as a consequence, $\deg \mathbf{f} \geqslant \deg \mathbf{f}_l \geqslant n-l$.

Now, we are ready to do the final calculations. Denote $K = |\mathbf{f}^{-1}(y)|$. Let l_1 be the number of $\mathbf{f}_i$'s obtained by substituting one of the variables in $\mathbf{f}_{i-1}$ by a constant, and $l_2 = l - l_1$ i.e the number of $\mathbf{f}_i$'s we get substituting one of the variables of $\mathbf{f}_{i-1}$ by a linear combination of other variables. It is easy to see that $l_1 \leqslant \log_2 q^q = q\log_2 q$ and $l_2 \leqslant \log_q K$. Summarizing,

$$\deg \mathbf{f} \geqslant n - l = n - l_1 - l_2 \geqslant n - q\log_2 q - \log_q K.$$

Hence,

$$q^{\deg \mathbf{f}} \geqslant q^{n - q\log_2 q - \log_q K},$$

and finally

$$|\mathbf{f}^{-1}(y)| = K = q^{\log_q K} \geqslant q^{n - \deg \mathbf{f} - q\log_2 q},$$

which finishes the proof of the lemma.

5 Deterministic algorithm

In this Section we prove Theorem 1.1. Let $\mathbf{A}$ be a fixed supernilpotent algebra of prime power order q^h and

$$\mathbf{p}(\overline{x}) = \mathbf{g}(\overline{x}) \tag{4}$$

be a given equation over $\mathbf{A}$. By Lemma 3.2 there exists a polynomial $\mathbf{f}$ over $\mathbf{F}_q$ of degree $d \leqslant |A|^{\log_q m+1}$ and arity hn, where m is bound on arity of a basic operation of $\mathbf{A}$, such that $\mathbf{f}(F_q^{hn}) \subseteq \{0,1\}$ and the equation

$$\mathbf{f}(x_1^1, \ldots, x_1^h, \ldots, x_n^1, \ldots, x_n^h) = 1 \tag{5}$$

has a solution iff equation (4) has a solution. We have even more, $\overline{a} \in A^n$ is a solution of equation (4) iff $\mathbf{f}(\pi_1(a_1), \ldots, \pi_h(a_1), \ldots, \pi_1(a_n), \ldots, \pi_h(a_n)) = 1$. Thus, it is enough to give an algorithm solving equation $\mathbf{f}(\overline{x}) = 1$.

Our algorithm treats circuits as a black-box and checks the set $S_{n,h} \subseteq F_q^{nh}$ of potential solutions of polynomial size in n with the property that if equation (5) has a solution it has a solution contained in S_{nh}. The algorithm returns "yes" if it finds the solution in a hitting set, and "no" otherwise. In the next paragraph we will show that such the set S_{nh} exists for every n and it can be computed in polynomial time. If $\mathbf{f}$ is a constant function, then the algorithm obviously returns the proper answer for every non-empty set of potential solutions as a hitting set. Hence, we can assume that f is not a constant function.

As every polynomial over $\mathbf{F}_q$ also the polynomial $\mathbf{f}$ can be presented as a sum of pairwise different monomials multiplied by nonzero constants from the field. Let t be a monomial taken from this presentation which contains the biggest number of different variables. From the fact that the degree of $\mathbf{f}$ is bounded by d we have that t depends on at most d variables. Now consider the polynomial $\mathbf{f}'$ formed by substituting variables not contained in t by $0 \in F_q$. Note that $\mathbf{f}'$ is not syntactically equal to any constant and hence it is not a constant function as every polynomial function over a finite filed has a unique representation (modulo equations $x^q = x$ for variables). Therefore, there exists a solution to the equation $\mathbf{f}'(\overline{x}) = 1$. Such a solution corresponds to a solution of equation (5) in which at most d variables are not equal 0. Hence, we obtain that equation (5) has a solution if it has a solution in which at most d variables are not equal 0. There are $O((q^h n)^d) = O(n^d)$ valuations of variables in which at most d variables are different than 0. Thus, to check if equation (5) has a solution it is enough to check $O(n^d)$ potential solutions and it can be done in time $O(n^d k)$, where k is the size of the circuit on the input.

It is worth noting, that the algorithm presented here does not compute the polynomial $\mathbf{f}$ from Lemma 3.2. We can translate the hitting set for (5) to a hitting set for the original equation over $\mathbf{A}$. That is why we are not concerned with the time complexity of computing $\mathbf{f}$.

6 Randomized algorithm

In this section we will prove Theorem 1.2 which says that there exists a linear time Monte Carlo algorithm solving Csat for supernilpotent algebras. More precisely, we will prove that if there exists a solution to the equation over a fixed supernilpotent algebra of prime power order, then by checking random assignment of variables with uniform distribution we will find the solution with probability at least c for some $c > 0$.

Let

$$\mathbf{p}(x_1, \ldots, x_n) = \mathbf{g}(x_1, \ldots, x_n)$$

be a given equation over supernilpotent algebra $\mathbf{A}$ of prime power order q^h. By Lemma 3.2 we get a function $\mathbf{f}$ which is an hn-ary polynomial over $\mathbf{F}_q$, such that $\overline{a} \in A$ is a solution to the above equation iff $\mathbf{f}(\pi_1 a_1, \ldots, \pi_h a_1, \ldots, \pi_1 a_n, \ldots, \pi_h a_n) = 1$. Moreover, the degree of $\mathbf{f}$ is bounded by a constant d which depends only on $\mathbf{A}$.

Assume $|\mathbf{f}^{-1}(1)| > 0$. Now, by Lemma 1.4 as $\mathbf{f}$ is nh-ary we obtain that $|\mathbf{f}^{-1}(1)| \geqslant q^{nh - \deg \mathbf{f} - q \log_2 q} \geqslant q^{nh - d - q \log_2 q}$. Observe that $\frac{|\mathbf{f}^{-1}(1)|}{|A|^n}$, the fraction of assignments of variables for which $\mathbf{f}$ is equal 1, is at least $c = \frac{q^{nh-d-g\log_2 q}}{q^{nh}} = q^{-d-q\log_2 q}$. This bound does not depend on $\mathbf{f}$ or n. Hence, the linear time randomized algorithm which picks the assignments of variables with uniform distribution and checks if picked assignments are a solution to the equation is a c-correct true-biased Monte Carlo algorithm solving Csat($\mathbf{A}$).

7 Conclusions

The main idea of the presented deterministic black-box algorithm correctness proof is translating polynomials of a nilpotent algebra $\mathbf{A}$ of prime power order to polynomials over $\mathbf{F}_q$ of small degree $d \leqslant |A|^{\log_q m + 1}$. This allowed us to create the hitting sets for Csat($\mathbf{A}$) by translating hitting sets for bounded degree polynomial equations over $\mathbf{F}_q$. It is worth to emphasize that this reasoning works for any hitting set. This means that any black-box algorithm for polynomials over $\mathbf{F}_q$ of degree at most d translates to an algorithm solving equations over supernilpotent algebras of prime power order. As each variable from $\mathbf{A}$ (in the reduction from Csat($\mathbf{A}$) to polynomial equations) is factored to at most $\log(|A|)$ variables, the reduction does not affect the time complexity too much. For instance, if we have some black-box algorithm for polynomial equations with hitting set of size $O(n^c)$, the same upper bound holds for Csat($\mathbf{A}$).

On the other hand it is easy to prove the dual theorem. For any polynomial equation over $\mathbf{F}_q$ of degree at most $d = \frac{|A|^{\log_q m}}{m}$ there is a nilpotent algebra $\mathbf{A}$ of size q^h and maximal arity of basic operation m such that any black-box algorithm for the algebra $\mathbf{A}$ translates to black box algorithm for solving equations over $\mathbf{F}_q$ of degree at most d. To see it, we will consider the following example.

▶ **Example 7.1.** For $h, m \in \mathbb{N}$, let $\mathbf{A}[h, m] = (A_h, +, p_1, \ldots, p_{h-1})$ be an algebra, such that:

- $(A_h, +) = \mathbf{Z}_q^h$,
- $\pi_i p_i(x_1, \ldots, x_m) = \prod_{j=1}^{k} \pi_{i+1} x_j$
- $\pi_j p_i(x_1, \ldots, x_m) = 0$ for $j \neq i$

Note that by results of [7] the algebra $\mathbf{A}$ from Example 7.1 is supernilpotent and belongs to a congruence modular variety. It is easy to see that every equation between polynomials over $\mathbf{F}_q$ of degree bounded by $d = m^{h-1} = \frac{m^{\log_q |A|}}{m} = \frac{|A|^{\log_q m}}{m}$ can be easily translated into

an equation over $\mathbf{A}$. Moreover, the projections on the first coordinate of elements of any hitting set for $\textsc{Csat}(\mathbf{A})$ is a hitting set for solving equations of polynomials over $\mathbf{F}_q$ of degree bounded by d.

In the light of above paragraphs, to obtain an efficient black-box algorithm solving Csat over supernilpotent algebras it is enough to produce a black-box algorithm for solving bounded degree equations for polynomials over fields and translate it to the black-box algorithm for supernilpotent algebras since any other black-box algorithm for supernilpotent algebras cannot be much more efficient (in terms of the size of the algebra and the maximal arity of an operation). So it seems that the right approach to find asymptotically optimal deterministic algorithms for supernilpotent algebras is to find optimal algorithms for polynomials of bounded degree.

There is a big disproportion between the computational complexity of deterministic and probabilistic algorithms presented in this paper. Hence, it would not be surprising if there was an effective derandomization of our Monte Carlo algorithm which would result in a new fast deterministic algorithm solving Csat. The results of this paper also imply, that there is one probabilistic algorithm for all supernilpotent algebras that is probabilistic FPT in terms of the algebras' signature. It is a nontrivial result, because if we were allowed to present a signature of a supernilpotent algebra on the input, such a problem would be NP-complete (to prove it, we can use the construction of the algebra shown in Example 7.1 to encode q-coloring).

References

1 Erhard Aichinger. Solving systems of equations in supernilpotent algebras. arXiv preprint arXiv:1901.07862, 2019.

2 Erhard Aichinger and Nebojša Mudrinski. Some applications of higher commutators in Mal'cev algebras. *Algebra universalis*, 63(4):367–403, 2010.

3 Andrei Bulatov. On the number of finite Mal'tsev algebras. *Contributions to general algebra*, 13:41–54, 2000.

4 Stanley Burris and Hanamantagouda P. Sankappanavar. *A Course in Universal Algebra*. Dover Publications, 2014.

5 Attila Földvári. The complexity of the equation solvability problem over nilpotent groups. *Journal of Algebra*, 495:289–303, 2018.

6 Attila Földvári and Gábor Horváth. The complexity of the equation solvability and equivalence problems over finite groups. *International Journal of Algebra and Computation*, 30(3):1–17, 2019.

7 Ralph Freese and Ralph McKenzie. *Commutator theory for congruence modular varieties*, volume 125. CUP Archive, 1987.

8 Mikael Goldmann and Alexander Russell. The complexity of solving equations over finite groups. *Information and Computation*, 178(1):253–262, 2002.

9 Tomasz Gorazd and Jacek Krzaczkowski. Term equation satisfiability over finite algebras. *International Journal of Algebra and Computation*, 20(8):1001–1020, 2010.

10 Tomasz Gorazd and Jacek Krzaczkowski. The complexity of problems connected with two-element algebras. *Reports on Mathematical Logic*, 46:91–108, 2011.

11 Johan Håstad. Satisfying degree-d equations over GF[2]. *Theory of Computing*, 9:845–862, 2013.

12 Gábor Horváth. The complexity of the equivalence and equation solvability problems over nilpotent rings and groups. *Algebra universalis*, 66(4):391–403, 2011.

13 Gábor Horváth. The complexity of the equivalence and equation solvability problems over meta-abelian groups. *Journal of Algebra*, 433:208–230, 2015.

14 Gábor Horváth and Csaba Szabó. The extended equivalence and equation solvability problems for groups. *Discrete Mathematics and Theoretical Computer Science*, 13(4):23–32, 2011.

15 Gábor Horváth and Csaba Szabó. The complexity of checking identities over finite groups. *International Journal of Algebra and Computation*, 16(5):931–940, 2006.

16 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Expressive power, satisfiability and equivalence of circuits over nilpotent algebras. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

17 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*, 2020.

18 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Stratifying algebras by supernilpotent intervals. Manuscript, 2020.

19 Paweł M. Idziak and Jacek Krzaczkowski. Satisfiability in multi-valued circuits. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 550–558. ACM, 2018.

20 Gyula Károlyi and Csaba Szabó. The complexity of the equation solvablity problem over nilpotent rings. Manuscript available at http://web.cs.elte.hu/~csaba/publications, 2015.

21 Piotr Kawałek, Michael Kompatscher, and Jacek Krzaczkowski. Circuit equivalence in 2-nilpotent algebras. arXiv preprint arXiv:1909.12256, 2019.

22 Keith Kearnes. Congruence modular varieties with small free spectra. *Algebra Universalis*, 42(3):165–181, 1999.

23 Michael Kompatscher. The equation solvability problem over supernilpotent algebras with Mal'cev term. *International Journal of Algebra and Computation*, 28(6):1005–1015, 2018.

24 Michael Kompatscher. CC-circuits and the expressive power of nilpotent algebras. arXiv preprint arXiv:1911.01479, 2019.

25 Bernhard Schwarz. The complexity of satisfiability problems over finite lattices. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 31–43, 2004.

26 Joel VanderWerf. Wreath decomposition of algebras. PhD thesis, University of California, Berkeley, 1995.