# Expressive Power, Satisfiability and Equivalence of Circuits over Nilpotent Algebras

**Paweł M. Idziak**

Jagiellonian University, Faculty of Mathematics and Computer Science,
Department of Theoretical Computer Science
Krakow, Poland
idziak@tcs.uj.edu.pl

**Piotr Kawałek**

Jagiellonian University, Faculty of Mathematics and Computer Science,
Department of Theoretical Computer Science
Krakow, Poland
piotr.kawalek@student.uj.edu.pl

**Jacek Krzaczkowski**

Jagiellonian University, Faculty of Mathematics and Computer Science,
Department of Theoretical Computer Science
Krakow, Poland
jacek.krzaczkowski@uj.edu.pl

## Abstract

Satisfiability of Boolean circuits is NP-complete in general but becomes polynomial time when restricted for example either to monotone gates or linear gates. We go outside Boolean realm and consider circuits built of any fixed set of gates on an arbitrary large finite domain. From the complexity point of view this is connected with solving equations over finite algebras. This in turn is one of the oldest and well-known mathematical problems which for centuries was the driving force of research in algebra. Let us only mention Galois theory, Gaussian elimination or Diophantine Equations. The last problem has been shown to be undecidable, however in finite realms such problems are obviously decidable in nondeterministic polynomial time.

A project of characterizing finite algebras $\mathbf{A}$ with polynomial time algorithms deciding satisfiability of circuits over $\mathbf{A}$ has been undertaken in [12]. Unfortunately that paper leaves a gap for nilpotent but not supernilpotent algebras. In this paper we discuss possible attacks on filling this gap.

**2012 ACM Subject Classification** Theory of computation → Complexity theory and logic, Theory of computation → Problems, reductions and completeness, Theory of computation → Circuit complexity, Theory of computation → Constraint and logic programming, Mathematics of computing → Combinatorial algorithms

**Keywords and phrases** circuit satisfiability, solving equations, Constraint Satisfaction Problem, structure theory

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2018.17

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

© Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski;
licensed under Creative Commons License CC-BY



## 1 Introduction

Solving equations (or systems of equations) over an algebra $\mathbf{A}$ is one of the oldest and well-known mathematical problems which for centuries was the driving force of research in algebra. Let us only mention Galois theory, Gaussian elimination or Diophantine Equations. In fact, for $\mathbf{A}$ being the ring of integers this is the famous 10th Hilbert Problem on Diophantine Equations, which has been shown to be undecidable [17]. In finite realms such problems are obviously decidable in nondeterministic polynomial time.

The decision version of solving systems of equations is strictly connected with Constraint Satisfaction Problem for relational structures. A bisimulation between these two problems has been presented in [15] and [5]. Due to this bisimulation and the recent dichotomy results for CSP [2, 20] one can translate the beautiful splitting conditions into the language of satisfiability of systems of equations. Unfortunately such a bisimulation between satisfiability of a single equation (denoted by PolSat) and CSP is not known. Therefore the search for a characterization of finite algebras with tractable PolSat seems to be challenging. There were isolated results for particular algebraic structures like groups [7, 10, 9], rings [8] or lattices [19]. However, already a study of PolSat for groups has shown [11] that for the alternating group $\mathbf{A}_4$ the problem has a polynomial time solution in the pure group language, while it is NP-complete after endowing $\mathbf{A}_4$ with binary commutator operation (which is obviously definable by group multiplication). Here the polynomial time complexity is a result of artificial inflation of the size of input – indeed the terms used in proving NP-completeness are exponentially longer in pure group language than in the language allowing group commutator. This unwanted phenomena should be eliminated when one wants to get a characterization of abstract algebras with polynomial time procedures for satisfiability of single equations. This project has been taken over in [12] where a measure of the size of input is based on circuits representing terms/polynomials. This eliminates the mentioned exponential inflation of the size of terms by replacing all terms by circuits that computes them. For a detailed discussion of this we refer to [12]. Here we only recall definitions (in this new setting) of two problems based on PolSat and its dual.

- Csat($\mathbf{A}$)
  given a circuit over the algebra $\mathbf{A}$ with two output gates $\mathbf{g}_1, \mathbf{g}_2$ is there a valuation of input gates $\overline{x} = (x_1, \ldots, x_n)$ that gives the same output on both $\mathbf{g}_1$ and $\mathbf{g}_2$, i.e. $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x})$ for some $\overline{x} \in A^n$.
- Ceqv($\mathbf{A}$)
  given a circuit over the algebra $\mathbf{A}$ is it true that for all inputs $\overline{x}$ we have the same values on given two output gates $\mathbf{g}_1, \mathbf{g}_2$, i.e. $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x})$.

This new approach proved itself to be very useful as the mentioned results for groups, rings and lattices have been extended in [12] to much more general setting. Actually the main result of [12] shows that both nilpotent groups and distributive lattices (isolated in [7] and [19] as those with Ptime algorithm for PolSat) form a paradigm for algebras with Csat solvable in polynomial time. Unfortunately nilpotent groups hide an extremely interesting phenomena, namely every finite nilpotent group is a direct product of groups of prime power order. In universal algebraic terms this means that nilpotent groups are supernilpotent. In fact, one of the the results contained in [12] (independently also shown in [14]) generalizes polynomial time algorithms from nilpotent groups to supernilpotent algebras. On the other hand, the results of [12] leave a gap: solvable algebras considered in [12] that are not nilpotent have NP-complete Csat and co-NP-complete Ceqv, while the complexity of Csat and Ceqv for nilpotent but not supernilpotent algebras is not known. Unfortunately to fill this gap the

PTIME algorithms presented in [1, 12, 14] are useless – they do not work in general nilpotent setting. They rely on the fact that if an equation has a solution then it must have one among relatively small set $S$ of tuples (although there may exist some other solutions outside the set $S$). More precisely: if $\mathbf{A}$ is a supernilpotent algebra (or a distributive lattice) then there is a constant $d$ so that for each natural number $n$ there is $S_n \subseteq A^n$ such that

- $|S_n|$ is $O(n^d)$,
- for two $n$-ary polynomials $\mathbf{s}$ and $\mathbf{t}$ the equation $\mathbf{s}(\overline{x}) = \mathbf{t}(\overline{x})$ has a solution $\overline{x} \in A^n$ iff it has a solution in $S_n$.

Also showing that two $n$-ary polynomials over an algebra determine the same $n$-ary function is reduced to checking this on a relatively small set $S_n$.

In this paper we show two things:

1. For nilpotent but not supernilpotent finite algebras one cannot expect polynomial time algorithms for CSAT or CEQV based on small search spaces $S_n$ described above, (unless $\mathsf{P} = \mathsf{NP}$).
2. On the other hand there are finite nilpotent but not supernilpotent algebras with tractable CSAT and CEQV problems.

The examples mentioned in the second item are expanded abelian groups $(\mathbb{Z}_{pq}; +, f)$ where $p, q$ are different primes, $+$ is the addition modulo $pq$ and $f(x)$ is a unary function that returns $x$ modulo $p$. Algorithms solving CEQV and CSAT for those algebras in polynomial time are described in sections 4 and 5, respectively. Those algorithms are based on a precise analysis of some kind of normal form of polynomials. The existence of such nice normal form is shown in section 3.

The evidence for the first item is provided by algebras with the very same clone of all polynomials as $(\mathbb{Z}_{pq}; +, f)$ but given by a different and infinite set of precisely chosen basic operations. In fact we show in section 6 that such algebras have co-NP-complete CEQV problem and NP-complete CSAT.

The reader should be warned here that leaving the safe realm, in which only finitely many basic operations are allowed, results in several fundamental problems. To start with, note that presenting an equation $\mathbf{s}(\overline{x}) = \mathbf{t}(\overline{x})$ we need to identify the basic operations that occur in $\mathbf{s}$ or $\mathbf{t}$. In fact, even the decidability of such redefined CSAT is not clear. One way to overcome this is to have an algebra $\mathbf{A} = (A; \mathbf{f}_0, \mathbf{f}_1, \ldots)$ encoded by a Turing machine $TM_{\mathbf{A}}$ which given the ($k$-ary) operation $\mathbf{f}_m$ and $a_1, \ldots, a_k \in A$ returns $TM_{\mathbf{A}}(m, a_1, \ldots, a_k) = \mathbf{f}_m(a_1, \ldots, a_k)$. Such approach puts extended CSAT($\mathbf{A}$) into NP whenever $TM_{\mathbf{A}}$ works in polynomial time. But it can be applied only to algebras with recursively enumerable set of fundamental operations.

The other way is to present an instance $\mathbf{s}(\overline{x}) = \mathbf{t}(\overline{x})$ of the problem together with the descriptions of all fundamental operations that occur in $\mathbf{s}$ or $\mathbf{t}$. Again such description may be done twofold:

- (CSAT$_{\mathrm{T}}$) by presenting the tables of the occurring basic operations, or
- (CSAT$_{\mathrm{TM}}$) by (polynomial time) algorithms $TM_{\mathbf{f}}$ computing the values $\mathbf{f}(a_1, \ldots, a_k)$.

It may seem that presenting operations by tables is more natural, as in many cases the complexity of such extended CSAT$_{\mathrm{T}}$ coincides with the complexity of its original (finite) version CSAT whenever the clone of an algebra is finitely generated (cf. Theorems 4.2, 5.2 and 6.2). On the other hand presenting the tables can again be treated as an artificial inflation of the input size. Indeed, Theorems 6.1 and 6.2 provide examples of nilpotent algebras with polynomial time CSAT$_{\mathrm{T}}$ and NP-complete CSAT$_{\mathrm{TM}}$. In fact item (1) above relies on such examples.

## 2 Background material

We use standard notation of universal algebra and computational complexity theory which can be found for example in [4], [18]. In particular, by an *algebra* we mean a pair $\mathbf{A} = (A; F)$, where $A$ is a nonempty set and $F$ is a family of finitary operations on $A$. Together with basic operations of $\mathbf{A}$, i.e. the operations from the set $F$, we often consider the derived operations – terms and polynomials of $\mathbf{A}$. By a polynomial of $\mathbf{A}$ we mean its term with some variables substituted by constants from $A$. We often refer to the syntactical side of the set $F$ of operations as the type of $\mathbf{A}$. If $F = \{f_1, \ldots, f_s\}$ we simply write $(A; f_1, \ldots, f_s)$ rather than $(A; F)$ and say that $\mathbf{A}$ is of finite type. We restrict ourselves to finite algebras, i.e. algebras with finite universe $A$ but we do allow infinite sets $F$ of basic operations. The set of all terms (or polynomials) of $\mathbf{A}$ is to be denoted by $\mathrm{Clo}\,\mathbf{A}$ (or $\mathrm{Pol}\,\mathbf{A}$). Two algebras $\mathbf{A}$ and $\mathbf{B}$ are said to be *polynomially equivalent* if they have the same universes and $\mathrm{Pol}\,\mathbf{A} = \mathrm{Pol}\,\mathbf{B}$.

This paper is also restricted to algebras that belong to congruence modular varieties, i.e. to algebras $\mathbf{A}$ that together with all subalgebras $\mathbf{D}$ of the powers $\mathbf{A}^n$ having modular lattices $\mathsf{Con}\,\mathbf{D}$ of their congruences. Congruence modular varieties include most known and well-studied algebras such as groups, rings, modules (and their generalizations like quasigroups, loops, near-rings, nonassociative rings, Lie algebras), lattices (and their extensions like Boolean algebras, Heyting algebras or other algebras connected with multi-valued logics including MV-algebras).

One reason of our restriction to algebras from congruence modular varieties is that the paper [12] deals with such algebras. The other one is that in congruence modular setting there is a pretty well working notion of commutator of congruences that nicely generalizes commutator of normal subgroups (in group theory) and multiplication of ideals (in ring theory). A deep study of this commutator is described in [6]. Here we only recall a couple of definitions needed later. For congruences $\alpha, \beta, \gamma \in \mathsf{Con}\,\mathbf{A}$ we say that $\alpha$ *centralizes* $\beta$ *modulo* $\gamma$ (and denote this by $C(\alpha, \beta; \gamma)$) if for every $n \geqslant 1$, every $(n+1)$-ary term $\mathbf{t}$ of $\mathbf{A}$, every $(a, b) \in \alpha$, and every $(c_1, d_1), \ldots, (c_n, d_n) \in \beta$ we have

$$\mathbf{t}(a, \overline{c}) \overset{\gamma}{\equiv} \mathbf{t}(a, \overline{d}) \quad \text{iff} \quad \mathbf{t}(b, \overline{c}) \overset{\gamma}{\equiv} \mathbf{t}(b, \overline{d}).$$

Now the *commutator* $[\alpha, \beta]$ of the congruences $\alpha, \beta \in \mathsf{Con}\,\mathbf{A}$ is the smallest congruence $\gamma \in \mathsf{Con}\,\mathbf{A}$ for which $C(\alpha, \beta; \gamma)$. With the help of the commutator one can define solvable, nilpotent and Abelian congruences. In this paper we are interested only in the two last concepts. To define nilpotency we first iterate the commutator by putting $\theta^{(0)} = \theta$ and $\theta^{(i+1)} = [\theta, \theta^{(i)}]$, whenever $\theta \in \mathsf{Con}\,\mathbf{A}$. Now we say that $\theta$ is *$k$-nilpotent* if $\theta^k$ is the identity relation $0_A$ on $\mathbf{A}$ and the algebra $\mathbf{A}$ is *nilpotent* if the largest congruence $1_A$ of $\mathbf{A}$ is $k$-nilpotent for some positive integer $k$. As in group theory, $\mathbf{A}$ is called *Abelian* if $1_A$ is 1-nilpotent. Abelian algebras from congruence modular varieties have been shown in [6] to have particularly nice structure. In fact they are *affine*, i.e. polynomially equivalent to unitary modules (over a ring with unit).

We will also need the following strengthening of the nilpotency. First, for a bunch of congruences $\alpha_1, \ldots, \alpha_k, \beta, \gamma \in \mathsf{Con}\,\mathbf{A}$ we say that $\alpha_1, \ldots, \alpha_k$ centralize $\beta$ modulo $\gamma$, and write $C(\alpha_1, \ldots, \alpha_k, \beta; \gamma)$, if for all polynomials $\mathbf{p} \in \mathrm{Pol}\,\mathbf{A}$ and all tuples $\overline{a}_1 \overset{\alpha_1}{\equiv} \overline{b}_1, \ldots, \overline{a}_k \overset{\alpha_k}{\equiv} \overline{b}_k$ and $\overline{u} \overset{\beta}{\equiv} \overline{v}$ such that

$$\mathbf{p}(\overline{x}_1, \ldots, \overline{x}_k, \overline{u}) \overset{\gamma}{\equiv} \mathbf{p}(\overline{x}_1, \ldots, \overline{x}_k, \overline{v})$$

for all possible choices of $(\overline{x}_1, \ldots, \overline{x}_k)$ in $\{\overline{a}_1, \overline{b}_1\} \times \ldots \times \{\overline{a}_k, \overline{b}_k\}$ but $(\overline{b}_1, \ldots . \overline{b}_k)$, we also have

$$\mathbf{p}(\overline{b}_1, \ldots, \overline{b}_k, \overline{u}) \overset{\gamma}{\equiv} \mathbf{p}(\overline{b}_1, \ldots, \overline{b}_k, \overline{v}).$$

This notion was introduced by A. Bulatov [3] and further developed by E. Aichinger and N. Mudrinski [1]. In particular they have shown that for all $\alpha_1, \dots, \alpha_k \in \mathsf{Con}\,\mathbf{A}$ there is the smallest congruence $\gamma$ with $C(\alpha_1, \dots, \alpha_k; \gamma)$ called the $k$-ary commutator and denoted by $[\alpha_1, \dots, \alpha_k]$. Such generalized commutator behaves especially well in algebras from congruence modular varieties. In particular this commutator is monotone, join-distributive and we have $[\alpha_1, [\alpha_2, \dots, \alpha_k]] \leqslant [\alpha_1, \dots, \alpha_k]$. Thus every $k$-supernilpotent algebra, i.e. algebra satisfying $[\underbrace{1, \dots, 1}_{k+1 \text{ times}}] = 0$, is $k$-nilpotent. The following properties, that can be easily inferred from the deep work of R. Freese and R. McKenzie [6] and K. Kearnes [13], have been summarized in [1].

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

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

Commutator polynomials mentioned in condition (3) of Theorem 2.1 are the paradigms for the failure of supernilpotency and are easily seen to be useful in coding a $k$-ary conjunction. We say that $\mathbf{t}(x_1, \dots, x_{k-1}, z) \in \operatorname{Pol}_k\mathbf{A}$ is a commutator polynomial of rank $k$ if

- $\mathbf{t}(a_1, \dots, a_{k-1}, b) = b$ whenever $b \in \{a_1, \dots, a_{k-1}\} \subseteq A$,
- $\mathbf{t}(a_1, \dots, a_{k-1}, b) \neq b$ for some $a_1, \dots, a_{k-1}, b \in A$.

As we have mentioned in the Introduction this paper is intended to give a better understanding of the problems Csat and Ceqv for nilpotent but not supernilpotent algebras. Like in the groups, in nilpotent setting the inputs $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x})$ of Csat or Ceqv can be restricted to the ones in which one of the polynomials is constant, i.e. of the form $\mathbf{g}(\overline{x}) = c$. Indeed, with the help of Corollary 7.4 in [6] it suffices to choose any $c \in A$ and put $\mathbf{g}(\overline{x}) = \mathbf{m}(\mathbf{g}_1(\overline{x}), \mathbf{g}_2(\overline{x}), c)$, where $\mathbf{m}(x, y, z)$ is a Mal'cev term for $\mathbf{A}$, i.e. a term satisfying $\mathbf{m}(x, x, y) = y = \mathbf{m}(y, x, x)$ for all $x, y \in A$.

## 3 The structure of 2-nilpotent algebras

To fill the nilpotent versus supernilpotent gap mentioned in the Introduction we need to understand the structure of nilpotent algebras. Since all algebras considered in this paper are 2-nilpotent we will use the description of their structure presented in Chapter VII of [6]. It reduces to an action of one abelian algebra over the other abelian one. Thus, after fixing the set $F$ of operations we need two affine algebras:

- an upper one, say $\mathbf{U}$, which is polynomially equivalent to a module $(U; \oplus)$ over a ring $\mathbf{R}_U$, and
- a lower one, say $\mathbf{L}$, which is polynomially equivalent to a module $(L; +)$ over a ring $\mathbf{R}_L$,

and for each basic operation $f \in F$, say $k$-ary, we need a function $\widehat{f} : U^k \longrightarrow L$. This allows us to construct an algebra $\mathbf{L} \otimes^F \mathbf{U}$ of type $F$ on the set $L \times U$ by putting

$$f^{\mathbf{L}\otimes^F\mathbf{U}}((l_1, u_1), \dots, (l_k, u_k)) = (f^{\mathbf{L}}(l_1, \dots, l_k) + \widehat{f}(u_1, \dots, u_k), f^{\mathbf{U}}(u_1, \dots, u_k)). \tag{1}$$

The usefulness of this construction is described in Corollary 7.2 in [6], where every 2-nilpotent algebra of type $F$ is shown to be of the form $\mathbf{L} \otimes^F \mathbf{U}$ for some triple $\mathbf{U}, \mathbf{L}, \left\{\widehat{f} : f \in F\right\}$. The rest of this section is devoted to presenting a nice normal form of arbitrary polynomial $\mathbf{p}$ of $\mathbf{A}$ that will be useful to construct polynomial time algorithms for Ceqv($\mathbf{A}$) and Csat($\mathbf{A}$).

To start with, note that the equation (1) remains valid for $f$ being not just a basic operation but an arbitrary polynomial of $\mathbf{A}$, where $\widehat{f}$ is appropriately chosen. Moreover having in mind that both $\mathbf{L}$ and $\mathbf{U}$ are affine we know that $f^{\mathbf{L}}(l_1,\dots,l_k)$ and $f^{\mathbf{U}}(u_1,\dots,u_k)$ are affine combinations $\sum_{i=1}^{k}\lambda_i l_i + l_0$ and $\bigoplus_{i=1}^{k}\alpha_i u_i \oplus u_0$. Thus for a polynomial $\mathbf{p}$ we have

$$\mathbf{p}^{\mathbf{L}\otimes^F\mathbf{U}}((l_1,u_1),\dots,(l_k,u_k)) = \left(\sum_{i=1}^{k}\lambda_i l_i + \widehat{\mathbf{p}}(u_1,\dots,u_k), \bigoplus_{i=1}^{k}\alpha_i u_i + u_0\right),$$

where $l_0$ is absorbed by $\widehat{\mathbf{p}}$. Moreover, $\widehat{\mathbf{p}}(u_1,\dots,u_k)$ can be presented as a sum of elements of the form

$$\mu\cdot\widehat{g}\left(\bigoplus_{i=1}^{k}\beta_i^{(1)}u_i\oplus u_0^{(1)},\dots,\bigoplus_{i=1}^{k}\beta_i^{(s)}u_i\oplus u_0^{(s)}\right),$$

with $\widehat{g}$ ranging over basic operations (and its occurrences) used to build $\mathbf{p}$.

The normal form, we have just started to build, has particularly nice shape in the cases where the $\widehat{g}$'s can be presented as affine combinations of unary $\widehat{f}$'s. One of such case is presented in the following Lemma.

▶ **Lemma 3.1.** *Let* $\mathbf{U}$ *and* $\mathbf{L}$ *be algebras polynomially equivalent to 1-dimentional vector spaces over prime fields of different characteristics. Moreover let* $f: U\longrightarrow L$ *be such that* $f(0_{\mathbf{U}})=0_{\mathbf{L}}$ *and* $\sum_{u\in U}f(u)\neq 0_{\mathbf{L}}$*. Then, every function* $g\colon U^k\longrightarrow L$ *can be expressed by*

$$g(x_1,\dots,x_k) = \sum_{(\overline{\beta},u)\in F_U^k\times U}\mu_{\overline{\beta},u}\cdot f\left(\bigoplus_{i=1}^{k}\beta_i x_i\oplus u\right). \tag{2}$$

**Proof.** Let $h^k_{a_1,\dots,a_k,a}\colon U^k\longrightarrow L$ be constantly $0_{\mathbf{L}}$ except $h^k_{a_1,\dots,a_k,a}(a_1,\dots,a_k)=a$. Observe that for every function $g\colon U^k\longrightarrow L$ we have

$$g(x_1,\dots,x_k) = \sum_{(a_1,\dots,a_k)\in U^k} h^k_{a_1,\dots,a_k,g(a_1,\dots,a_k)}(x_1,\dots,x_k).$$

Moreover we can express one spike function by any other one by putting

$$h^k_{a_1,\dots,a_k,a}(x_1,\dots,x_k) = a\cdot b^{-1}\cdot h^k_{b_1,\dots,b_k,b}(x_1-a_1+b_1,\dots,x_k-a_k+b_k).$$

In the above we use 1-dimensionality so that the universes of $\mathbf{L}$ and $\mathbf{F}_L$ coincide, so that the vectors $a,b\in L$ can be also treated as scalars from $\mathbf{F}_L$.

The last two displays yield that to finish the proof of the Lemma it suffices to represent, for each $k$, one spike function in the form described in (2). The rest of the proof shows how this can be done in the presence of a unary function $f$ described in the Lemma, which is to be called $(\mathbf{U},\mathbf{L})$-normal in the rest of the paper. Our construction of such spike functions is based on counting partitions (into special sums) of elements in vector spaces over finite fields.

Being left with representing one spike function of arbitrary large arity by an expression of the form (2) we start with letting $p$ and $q$ to be characteristics of the fields $\mathbf{F}_U$ and $\mathbf{F}_L$, respectively. Moreover let $f\colon U\longrightarrow L$ be $(\mathbf{U},\mathbf{L})$-normal . Put

$$t_s(x_1,\dots,x_s) = \sum_{\emptyset\neq I\subseteq\{1,\dots,s\}}(-1)^{|I|}f\left(\bigoplus_{i\in I}x_i\right)$$

and note that $t_s(x_1,\dots,x_s)=0_{\mathbf{L}}$ whenever $0_{\mathbf{U}}\in\{x_1,\dots,x_s\}$. We fix an enumeration of the set $U=\{0_{\mathbf{U}},u_1,\dots,u_{p-1}\}$ and define

$$w_k(x_1,\dots,x_k)=t_{k(p-1)}(x_1-u_1,\dots,x_k-u_1,x_1-u_2,\dots,x_k-u_2,\dots,x_1-u_{p-1},\dots,x_k-u_{p-1}).$$

Observe that $w_k(\overline{x})=0_{\mathbf{L}}$ whenever at least one of the $x_i$'s is non-zero. To prove that $w_k$ is a spike function it remains to show $w_k(0_{\mathbf{U}},\dots,0_{\mathbf{U}})\neq 0_{\mathbf{L}}$. Our first claim is

$$w_k(0_{\mathbf{U}},\dots,0_{\mathbf{U}}) \quad = \quad t_{p-1}(u_1,\dots,u_{p-1}). \tag{3}$$

Since $t_s$ is fully symetric, we have

$$w_k(0_{\mathbf{U}},\dots,0_{\mathbf{U}})=t_{k(p-1)}(\overbrace{u_1,\dots,u_1}^{k\text{ times}},\overbrace{u_2,\dots,u_2}^{k\text{ times}},\dots,\overbrace{u_{p-1},\dots,u_{p-1}}^{k\text{ times}}).$$

Without loss of generality we may assume that $k=q^a$, as for any other $k'$ to get a $k'$-ary spike we choose the smallest power $q^a\geqslant k'$ and replace $q^a-k'$ arguments with $0_{\mathbf{U}}$. From the definition of $t_{k(p-1)}$ we get

$$\begin{aligned}&t_{k(p-1)}(u_1,\dots,u_1,u_2,\dots,u_2,\dots,u_{p-1},\dots,u_{p-1})=\\&=\sum_{k=1}^{(p-1)\cdot q^a}(-1)^k\sum_{\substack{k_1+\dots+k_{p-1}=k\\k_i\leqslant q^a}}\binom{q^a}{k_1}\cdot\ldots\cdot\binom{q^a}{k_{p-1}}f\left(\bigoplus_{i=1}^{k_1}u_1\oplus\ldots\oplus\bigoplus_{i=1}^{k_{p-1}}u_{p-1}\right).\end{aligned}$$

Observe that $\binom{q^a}{k_i}$ is divisible by $q$ whenever $k_i\notin\{0,q^a\}$. Thus the only summands that do not vanish are those with $k_i\in\{0,q^a\}$ for all $i$ so that $\binom{q^a}{k_1}\cdot\ldots\cdot\binom{q^a}{k_{p-1}}=1$. Thus, by changing notation, we have

$$\begin{aligned}&t_{k(p-1)}(u_1,\dots,u_1,u_2,\dots,u_2,\dots,u_{p-1},\dots,u_{p-1})=\\&=\sum_{k'=1}^{p-1}(-1)^{k'\cdot q^a}\sum_{\substack{k'_1+\dots+k'_{p-1}=k'\\k'_i\in\{0,1\}}}f\left(\bigoplus_{i=1}^{k'_1\cdot q^a}u_1\oplus\ldots\oplus\bigoplus_{i=1}^{k'_{p-1}\cdot q^a}u_{p-1}\right)=\\&=\sum_{k'=1}^{p-1}(-1)^{k'\cdot q^a}\sum_{\substack{S\subseteq\{1,\dots,p-1\}\\|S|=k'}}f\left(\bigoplus_{i\in S}u_i\right),\end{aligned}$$

where the last equality follows from the fact that the multisets

$$\left\{\left\{\bigoplus_{i=1}^{k'_1\cdot q^a}u_1\oplus\ldots\oplus\bigoplus_{i=1}^{k'_{p-1}\cdot q^a}u_{p-1}\colon k'_1+\ldots+k'_{p-1}=k',k'_i\in\{0,1\}\right\}\right\}$$

and

$$\left\{\left\{\bigoplus_{i\in S}u_i\colon S\subseteq\{1,\dots,p-1\},|S|=k'\right\}\right\}.$$

are equal. To complete the proof of (3) observe that $(-1)^{k'\cdot q^a}=(-1)^{k'}$. This is obvious for $q$ being odd, while otherwise it follows from the fact that $x=-x$.

We will conclude our proof by showing that

$$w_k(0_{\mathbf{U}},\dots,0_{\mathbf{U}})\neq 0_{\mathbf{L}}. \tag{4}$$

From the definition of $t_s$ we know that

$$t_{p-1}(u_1,\dots,u_{p-1}) = \sum_{k=1}^{p-1}(-1)^k \sum_{\substack{S\subseteq\{1,\dots,p-1\}\\ |S|=k}} f\left(\bigoplus_{i\in S} u_i\right).$$

Now, if $\ell^k_{u_i}$ denotes a number of partition of the element $u_i$ into a sum of $k$ non-zero pairwise different elements from $U$ then by appropriately grouping the $f\left(\bigoplus_{i\in S} u_i\right)$'s and noting that $\mathbf{f}(0_\mathbf{U}) = 0_\mathbf{L}$ we can replace the last sum by

$$\sum_{k=1}^{p-1}(-1)^k \sum_{i=1}^{p-1} \ell^k_{u_i} f(u_i)$$

The numbers $\ell^k_{u_i}$ were calculated in [16] to be $\ell^k = \ell^k_{u_i} = \frac{1}{p}\left(\binom{p-1}{k} + (-1)^{k+1}\right)$ independently of $u_i \neq 0_\mathbf{U}$. This gives that

$$t_{p-1}(u_1,\dots,u_{p-1}) = \sum_{k=1}^{p-1}(-1)^k \ell^k \sum_{i=1}^{p-1} f(u_i) = \left(\sum_{k=1}^{p-1}(-1)^k \ell^k\right)\left(\sum_{i=1}^{p-1} f(u_i)\right).$$

Using the explicit formulas for the $\ell^k$'s we get

$$\begin{aligned}
\sum_{k=1}^{p-1}(-1)^k \ell^k &= \frac{1}{p}\sum_{k=1}^{p-1}(-1)^k\left[\binom{p-1}{k} + (-1)^{k+1}\right]\\
&= \frac{1}{p}\sum_{k=1}^{p-1}\left[(-1)^k\binom{p-1}{k} + (-1)^{2k+1}\right]\\
&= \frac{1}{p}\left[\sum_{k=1}^{p-1}(-1)^k\binom{p-1}{k} + \sum_{k=1}^{p-1}(-1)\right]\\
&= \frac{1}{p}\left[-1-(p-1)\right] = -1
\end{aligned}$$

This gives us that

$$w_k(0_\mathbf{U},\dots,0_\mathbf{U}) = t_{p-1}(u_1,\dots,u_{p-1}) = -\sum_{i=1}^{p-1} f(u_i)$$

is not equal to $0_\mathbf{L}$ as $f$ is $(\mathbf{U},\mathbf{L})$-normal . Thus the claim (4) holds, proving that $w_k$ is a spike function, as required. ◀

Lemma 3.1 yields that, in its setting, every polynomial $\mathbf{p}$ of $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ can be represented as

$$\mathbf{p}((l_1,u_1),\dots,(l_k,u_k)) = \left(\sum_{i=1}^{k}\lambda_i l_i + \sum_{\substack{\overline{\beta}\in R_U^k\\ u\in U}} \mu_{\overline{\beta},u}\cdot f\left(\bigoplus_{i=1}^{k}\beta_i u_i \oplus u\right), \bigoplus_{i=1}^{k}\alpha_i u_i \oplus u_0\right) \quad (5)$$

where $f$ is a $(\mathbf{U},\mathbf{L})$-normal function. Representations of the above form are to be called $f$-normal. The size of such normal form is essentially the number of non-zero coefficients among the $\lambda_i$'s, $\mu_{\overline{\beta},u}$'s and $\alpha_i$'s.

Most of our arguments in this paper refer to $f$-normal forms where the modules hidden in $\mathbf{U}$ and $\mathbf{L}$ are actually 1-dimensional vector spaces over prime fields of different characteristics. However $f$-normal forms could be useful in more general settings as shown in the following Lemma as well as in Lemma 4.1.

▶ **Lemma 3.2.** *Let* $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ *be a finite 2-nilpotent algebra. Then for each* $f\colon U \longrightarrow L$ *there exists a polynomial time procedure that returns* $f$*-normal form for a polynomial* $\mathbf{g}$ *of* $\mathbf{A}$ *presented together with* $f$*-normal forms of all the basic operations occurring in* $\mathbf{g}$*.*

**Proof.** To prove the lemma we start with two polynomials $\mathbf{h}_1, \mathbf{h}_2$ of $\mathbf{A}$ presented in their $f$-normal forms to carefully compute $f$-normal form of the superposition

$$\mathbf{g}(x_1, \dots, x_{k_1+k_2-1}) = \mathbf{h}_2(\mathbf{h}_1(x_1, \dots, x_{k_1}), x_{k_1+1}, \dots, x_{k_1+k_2-1})$$

in a polynomial time. Let

$$\mathbf{h}_j((l_1, u_1), \dots, (l_{k_j}, u_{k_j})) = \left( \sum_{i=1}^{k_j} \lambda_i^{(j)} l_i + \sum_{(\overline{\beta}, u) \in \Gamma_j} \mu^{(j)}_{\overline{\beta}, c} \cdot f\left( \bigoplus_{i=1}^{k_j} \beta_i u_i \oplus u \right), \bigoplus_{i=1}^{k_j} \alpha_i^{(j)} u_i \oplus u_0^{(j)} \right)$$

for appropriate $\Gamma_j \subseteq \mathbf{R}_U^{k_j} \times U$.
Now the second and the first coordinates of $\mathbf{g}((l_1, u_1), \dots, (l_{k_1+k_2-1}, u_{k_1+k_2-1}))$ are easily seen to be

$$\bigoplus_{i=1}^{k_1} \alpha_1^{(2)} \alpha_i^{(1)} u_i \oplus \bigoplus_{i=2}^{k_2} \alpha_i^{(2)} u_{i+k_1-1} \oplus \alpha_1^{(2)} u_0^{(1)} \oplus u_0^{(2)},$$

and

$$\sum_{i=1}^{k_1} \lambda_1^{(2)} \lambda_i^{(1)} l_i + \sum_{i=2}^{k_2} \lambda_i^{(2)} l_{i+k_1-1} + \lambda_1^{(2)} \cdot \sum_{(\overline{\beta}, u) \in \Gamma_1} \mu^{(1)}_{\overline{\beta}, u} \cdot f\left( \bigoplus_{i=1}^{k_1} \beta_i u_i \oplus u \right) + g'(\overline{u}),$$

where $g'(u_1, \dots, u_{k_1+k_2-1})$ is obtained from $\sum_{(\overline{\beta}, u) \in \Gamma_2} \mu^{(2)}_{\overline{\beta}, u} \cdot f\left( \beta_1 u_1 \oplus \bigoplus_{i=2}^{k_2} \beta_i u_{k_1+i-1} \oplus u \right)$ by substituting every occurrence of $u_1$ by $\bigoplus_{i=1}^{k_1} \alpha_i^{(1)} u_i \oplus u_0^{(1)}$.

It should be obvious that both these coordinates give $f$-normal form of $\mathbf{g}$ and that they can be computed in a polynomial time in size of $f$-normal forms of $\mathbf{h}_1$ and $\mathbf{h}_2$. Actually a careful inspection of our argument allows us to bound the number of non-zero summands so that the $f$-normal forms of more complicated superpositions of polynomials of $\mathbf{A}$ can be also computed efficiently. ◀

Combining Lemmas 3.1 and 3.2 we get

▶ **Corollary 3.3.** *Let* $\mathbf{L}$ *and* $\mathbf{U}$ *be algebras polynomially equivalent to one dimensional vectors spaces over prime fields of different characteristics and* $f\colon U \longrightarrow L$ *be* $(\mathbf{U}, \mathbf{L})$*-normal . Then every polynomial operation of* $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ *has* $f$*-normal forms and, if the type* $F$ *of* $\mathbf{A}$ *is finite then one of these* $f$*-normal forms can be computed in polynomial time.*

## 4 Equivalence

The last paragraph of Section 2 shows that in the nilpotent setting in CEQV it suffices to consider equivalence of two polynomials one of which is constant.

▶ **Lemma 4.1.** *Let* $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ *with* $\mathbf{L}$ *and* $\mathbf{U}$ *being polynomially equivalent to one-dimensional vector spaces over finite fields of different characteristics. Then there exists a polynomial time algorithm which for polynomials* $\mathbf{p}$ *of* $\mathbf{A}$ *given in some* $f$*-normal form decides if* $\mathbf{p}$ *is a constant function.*

**Proof.** Let

$$\mathbf{p}((l_1,u_1),\dots,(l_k,u_k)) = \left(\mathbf{p}^{\mathbf{L}}(l_1,\dots,l_k) + \sum_{\substack{\overline{\beta}\in\mathbf{F}_U^k\\ u\in U}} \mu_{\overline{\beta},u}\cdot f\left(\bigoplus_{i=1}^k \beta_i u_i \oplus u\right), \mathbf{p}^{\mathbf{U}}(u_1,\dots,u_k)\right)$$

be an $f$-normal form of $\mathbf{p}$. Obviously such polynomial $\mathbf{p}$ is constant iff both coordinates of the right-hand side above are constant. This can be efficiently checked for $\mathbf{p}^{\mathbf{U}}(u_1,\dots,u_k)$ as it is simply a polynomial of a vector space. Since both summands of the first coordinate on the right-hand side depend on disjoint sets of variables (the $l_i$'s and the $u_i$'s) to keep their sum constant we need to keep both summands constant. Again, checking that for $\mathbf{p}^{\mathbf{L}}$ (in a vector space) is fast so that we are left with the expression of the form

$$\widehat{\mathbf{p}}(u_1,\dots,u_k) = \sum_{\substack{(\overline{\beta},u)\in\mathbf{F}_U^k\times U\\ \overline{\beta}\neq\overline{0}}} \mu_{\overline{\beta},u}\cdot f\left(\bigoplus_{i=1}^k \beta_i u_i \oplus u\right). \tag{6}$$

This in turn can be done with the help of the following claim.

($\star$) $\widehat{\mathbf{p}}$ is constant iff for each $\overline{\beta}\in\mathbf{F}_U^k\setminus\{\overline{0}\}$ the function $S_{\overline{\beta}}(x) = \sum_{(\kappa,u)\in F_U^*\times U}\mu_{\kappa\cdot\overline{\beta},u}\cdot f(\kappa x\oplus u)$ is constant on $U$.

Indeed, having ($\star$) we argue as follows. If for a particular $\overline{\beta}$ all the $\mu_{\kappa\cdot\overline{\beta},u}$'s are zero then $S_{\overline{\beta}}(x)=0$ for all $x\in U$. For any other $\overline{\beta}$ we simply check if $S_{\overline{\beta}}$ is constant by computing all of the $|U|$ values for the $x$'s. This is fast as we have at most $|U|^2$ summands. Finally, the number of the $\overline{\beta}$'s of the second kind is linear in the number of non-zero coefficients $\mu_{\overline{\beta},u}$. This in turn is bounded by the length of the expression (6) which is the part of the input.

Thus we are left with the proof of ($\star$). To see the 'only if' direction for $\overline{\beta}\in\mathbf{F}_{\mathbf{U}}^k\setminus\{\overline{0}\}$ and $a\in U$ define $\mathbf{O}_{a,\overline{\beta}} = \{\overline{u}\in U^k : \bigoplus_{i=1}^k\beta_i u_i = a\}$. Observe that the size of the solution set $\mathbf{O}_{a,\overline{\beta}}$ of a nontrivial linear equation is always $|U|^{k-1}$ independently of the choice of $(a,\overline{\beta})$. Now, since $\widehat{\mathbf{p}}$ is constant, for each $\overline{\beta}\in F_U^k\setminus\{0\}^k$ and $a,b\in L$ we have

$$\begin{aligned}
0 &= \sum_{\overline{u}\in\mathbf{O}_{a,\overline{\beta}}}\widehat{\mathbf{p}}(\overline{u}) - \sum_{\overline{u}\in\mathbf{O}_{b,\overline{\beta}}}\widehat{\mathbf{p}}(\overline{u})\\
&= \sum_{\overline{u}\in\mathbf{O}_{a,\overline{\beta}}}\sum_{\substack{(\overline{\gamma},u)\in\mathbf{F}_U^k\times U\\ \overline{\gamma}\neq\overline{0}}}\mu_{\overline{\gamma},u}\cdot f\left(\bigoplus_{i=1}^k\gamma_i u_i\oplus u\right) - \sum_{\overline{u}\in\mathbf{O}_{b,\overline{\beta}}}\sum_{\substack{(\overline{\gamma},u)\in\mathbf{F}_U^k\times U\\ \overline{\gamma}\neq\overline{0}}}\mu_{\overline{\gamma},u}\cdot f\left(\bigoplus_{i=1}^k\gamma_i u_i\oplus u\right)\\
&= \sum_{\substack{(\overline{\gamma},u)\in\mathbf{F}_U^k\times U\\ \overline{\gamma}\neq\overline{0}}}\mu_{\overline{\gamma},u}\cdot\left(\sum_{\overline{u}\in\mathbf{O}_{a,\overline{\beta}}} f\left(\bigoplus_{i=1}^k\gamma_i u_i\oplus u\right) - \sum_{\overline{u}\in\mathbf{O}_{b,\overline{\beta}}} f\left(\bigoplus_{i=1}^k\gamma_i u_i\oplus u\right)\right)\\
&= \sum_{\substack{(\overline{\gamma},u)\in\mathbf{F}_U^k\times U\\ \overline{\gamma}\neq\overline{0}}}\mu_{\overline{\gamma},u}\cdot\left(t(a,\overline{\beta},\overline{\gamma},u) - t(b,\overline{\beta},\overline{\gamma},u)\right),
\end{aligned}$$

where $t(x,\overline{\beta},\overline{\gamma},u) = \sum_{\overline{u}\in\mathbf{O}_{x,\overline{\beta}}} f\left(\bigoplus_{i=1}^k\gamma_i u_i\oplus u\right)$. Now, if the vectors $\overline{\beta},\overline{\gamma}\in\mathbf{F}_U^k$ are linearly dependent, i.e. $\overline{\gamma}=\kappa\cdot\overline{\beta}$ we have $t(x,\overline{\beta},\overline{\gamma},u) = |U|^{k-1}\cdot f(\kappa\cdot x\oplus u)$. Otherwise, if $\overline{\beta}$ and $\overline{\gamma}$ are linearly independent then $t(x,\overline{\beta},\overline{\gamma},u) = |U|^{k-2}\sum_{d\in U} f(d)$, as the system of the following

two equations

$$\begin{cases} \bigoplus_{i=1}^{k} \beta_i u_i = x \\ \bigoplus_{i=1}^{k} \gamma_i u_i \oplus u = d \end{cases}$$

has exactly $|U|^{k-2}$ solutions. Summing up in the big display above the summands with $\overline{\gamma}$'s that are linearly independent with $\overline{\beta}$ diminishes so that this display reduces to

$$\begin{aligned} 0 &= \sum_{\substack{(\overline{\gamma},u)\in \mathbf{F}_U^k \times U \\ \overline{\gamma}\neq\overline{0}}} \mu_{\overline{\gamma},u} \cdot \big(t(a,\overline{\beta},\overline{\gamma},u) - t(b,\overline{\beta},\overline{\gamma},u)\big) \\ &= |U|^{k-1} \cdot \left( \sum_{(\kappa,u)\in \mathbf{F}_U^* \times U} \mu_{\kappa\cdot\overline{\beta},u} f(\kappa\cdot a \oplus u) - \sum_{(\kappa,u)\in \mathbf{F}_U^* \times U} \mu_{\kappa\cdot\overline{\beta},u} f(\kappa\cdot b \oplus u) \right). \end{aligned}$$

Since $|U|$ and $|L|$ are coprime, the difference in the parenthesis is zero which shows the 'only if' direction of $(\star)$.

To prove the 'if' direction observe that $R = \{(\overline{\beta},\kappa\overline{\beta})\colon \overline{\beta}\in F_U^k \setminus \{\overline{0}\}, \kappa\in \mathbf{F}_U^*\}$ is an equivalence relation. Let $\overline{\beta}^{(1)},\overline{\beta}^{(2)},\dots,\overline{\beta}^{(m)}$ be the transversal of $R$. Then for $\overline{x}\in U^k$ we have

$$\begin{aligned} \widehat{\mathbf{p}}(x_1,\dots,x_k) &= \sum_{\substack{(\overline{\gamma},u)\in F_{\mathbf{U}}^k \times U \\ \overline{\gamma}\neq\overline{0}}} \mu_{\overline{\gamma},u} \cdot f\left(\bigoplus_{i=1}^{k} \gamma_i \cdot x_i \oplus u\right) \\ &= \sum_{j\in\{1,\dots,m\}} \sum_{(\kappa,u)\in F_{\mathbf{U}}^* \times U} \mu_{\kappa\cdot\overline{\beta}^{(j)},u} \cdot f\left(\kappa\cdot\bigoplus_{i=1}^{k} \beta_i^{(j)} x_i \oplus u\right) \\ &= \sum_{j\in\{1,\dots,m\}} S_{\overline{\beta}^{(j)}}\left(\bigoplus_{i=1}^{k} \beta_i^{(j)} x_i\right). \end{aligned}$$

Now our assumption that all the $S_{\overline{\beta}^{(j)}}$ are constant shows that $\widehat{\mathbf{p}}$ is constant, as well. ◀

Corollary 3.3 together with Lemma 4.1 immediately give the following theorem.

▶ **Theorem 4.2.** *Let* $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ *with* $\mathbf{L}$ *and* $\mathbf{U}$ *being polynomially equivalent to* 1*-dimensional vector spaces over prime fields of different characteristics. Then* $\textsc{Ceqv}(\mathbf{A})$ *is in P.*

## 5 Satisfiability

Again, in nilpotent realm the last paragraph of Section 2 allows us to fix one side of the equations considered in Csat to be a constant polynomial.

▶ **Lemma 5.1.** *Let* $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ *with* $\mathbf{L}$ *and* $\mathbf{U}$ *being polynomially equivalent to one-dimensional vector spaces over finite fields of different characteristics. Then there exists a polynomial time algorithm which for a constant* $c\in A$ *and polynomials* $\mathbf{p}$ *of* $\mathbf{A}$ *given in some* $f$*-normal form decides if the equation* $\mathbf{p}(\overline{x}) = c$ *has a solution.*

**Proof.** Since our polynomial is given in $f$-normal form we start with the following equation

$$\left( p^{\mathbf{L}}(l_1,\dots,l_k) + \sum_{(\overline{\beta},u)\in \mathbf{F}_U^k \times U} \mu_{\overline{\beta},u} \cdot f\left(\bigoplus_{i=1}^{k} \beta_i \cdot u_i \oplus u\right), p^{\mathbf{U}}(u_1,\dots,u_k) \right) = (c_L, c_U). \quad (7)$$

We start with observing that since $\mathbf{L}$ and $\mathbf{U}$ are polynomially equivalent to one-dimensional vector spaces the range of $p^{\mathbf{L}}$ (and $p^{\mathbf{U}}$) is either one element or the entire $L$ (or $U$). Now, if $p^{\mathbf{L}}$ is not constant then it suffices to check whether $p^{\mathbf{U}}(u_1,\dots,u_k) = c_U$ has a solution in $U$, as the solution in first coordinate always exists. Thus we assume that $p^{\mathbf{L}}$ is constant and put $d = c_L - p^{\mathbf{L}}(0_{\mathbf{L}},\dots,0_{\mathbf{L}})$. Moreover we may assume that $p^{\mathbf{U}}$ is not constant or equal to $c_U$, as otherwise our equation has no solution.

We want to reduce our equation in $\mathbf{A}$ to an equivalent equation of the form

$$\sum_{(\overline{\beta},u)\in\mathbf{F}_U^k\times U} \nu_{\overline{\beta},u}\cdot f\left(\bigoplus_{i=1}^{k}\beta_i\cdot u_i\oplus u\right) \;=\; d \tag{8}$$

in $\mathbf{L}$, but with the $u_i$'s taking values in $U$. Now, if $p^{\mathbf{U}}$ is constant (and therefore equal to $c_U$) then we are done with $\nu_{\overline{\beta},u} = \mu_{\overline{\beta},u}$ for all $\overline{\beta}$'s and $u$'s. Otherwise $p^{\mathbf{U}}(u_1,\dots,u_k) = c_U$ reduces to something of the form $u_j = \bigoplus_{i\neq j}\delta_i u_i \oplus c_U$. This allows us to replace all occurrences of the $u_j$ by the sum $\bigoplus_{i\neq j}\delta_i u_i\oplus c_U$ and after recalculating the coefficients $\mu_{\overline{\beta},u}$ we get the desired $\nu_{\overline{\beta},u}$'s as required in (8).

Denote the left-hand side of (8) by $p'(u_1,\dots,u_k)$ and note that since the set $L$ carries the multiplication inherited from the field $\mathbf{F}_L$ the equation $p'(u_1,\dots,u_k) = d$ has a solution iff

$$\prod_{d'\in L-\{d\}} (p'(u_1,\dots,u_k) - d') = 0_{\mathbf{L}}$$

is not an identity. Note however that the product above is not directly expressible as a polynomial of $\mathbf{A}$. Nevertheless distributing over the factors we can replace the product by the sum

$$\sum_{(\overline{\beta}^{(1)},u^{(1)})}\sum_{(\overline{\beta}^{(2)},u^{(2)})}\cdots\sum_{(\overline{\beta}^{(m)},u^{(m)})} \nu_{\mathcal{I}}\cdot f\left(\bigoplus_{i=1}^{k}\beta_i^{(1)}u_i\oplus u^{(1)}\right)\cdot\ldots\cdot f\left(\bigoplus_{i=1}^{k}\beta_i^{(m)}u_i\oplus u^{(m)}\right),$$

where $\mathcal{I} = \left(\overline{\beta}^{(1)},u^{(1)},\dots,\overline{\beta}^{(m)},u^{(m)}\right)$ and $m = |U|-1$. Lemma 3.1 supplies us with a representation of the product $f(x_1)\cdot\ldots\cdot f(x_m)$ sending $U^m$ into $L$ by an expression of the right-hand side of (2). This leads to the representation of the form

$$\prod_{d'\in L-\{d\}} (p'(u_1,\dots,u_k) - d') = \sum_{(\overline{\beta},u)\in\mathbf{F}_U^n\times U} \mu'_{\overline{\beta},u}\cdot f\left(\bigoplus_{i=1}^{k}\beta_i\cdot u_i\oplus u\right)$$

which, with the help of Lemma 4.1, can be efficiently checked not to be constantly $0_{\mathbf{L}}$. This in turn is equivalent for the starting equation to have a solution. ◀

Corollary 3.3 and Lemma 5.1 yield the following result.

▶ **Theorem 5.2.** *Let* $\mathbf{A} = \mathbf{L}\otimes^F\mathbf{U}$ *with* $\mathbf{L}$ *and* $\mathbf{U}$ *being polynomially equivalent to one dimensional vectors spaces over prime fields of different characteristics. Then* $\textsc{Csat}(\mathbf{A})$ *is in P.*

## 6 Algebras with infinitely many operations

The reasons we consider infinite languages in this paper are twofold. One motivation comes from the fact that a desire for a simple extension of our polynomial time algorithms from supernilpotent algebras to nilpotent ones is hopeless. Indeed, as it has been already mentioned

in the Introduction, those algorithms are based on a reduction to a small search space. Its size and shape is bounded as a result of the bound for the essential arity of commutator terms (they serve as multi-ary internal conjunctions replacing lack of an external one). At first glance the existence of polynomial time algorithms for nilpotent but not supernilpotent algebras is not so obvious as they do have commutator terms of arbitrary large arity so that one can try to interpret NP-complete problems like in solvable but nonnilpotent case. However all known commutator terms have exponential size with respect to the number of variables whenever they are produced from finitely many operations. In fact the circuits representing those known commutator polynomials are also of exponential size. However, allowing infinitely many operations, we do have the possibility to express arbitrary large conjunctions by short polynomials. This phenomena is presented in the next Theorem.

▶ **Theorem 6.1.** *For two different prime numbers $p, q$ there exists a 2-nilpotent algebra $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ with $\mathbf{U}$ and $\mathbf{L}$ being polynomially equivalent to one dimensional vector spaces over the fields $GF(p)$ and $GF(q)$ respectively, such that $\textsc{Ceqv}_{\textsc{tm}}(\mathbf{A})$ is co-NP-complete and $\textsc{Csat}_{\textsc{tm}}(\mathbf{A})$ is NP-complete.*

**Proof.** We start with choosing $a \in U - \{0\}$ and $b \in L - \{0\}$ to define the following family of functions:

$$f_k((l_1^1, u_1^1), (l_2^1, u_2^1), (l_3^1, u_3^1), \ldots, (l_1^k, u_1^k), (l_2^k, u_2^k), (l_3^k, u_3^k)) = (\widehat{f_k}(u_1^1, u_2^1, u_3^1, \ldots, u_1^k, u_2^k, u_3^k), 0),$$

where

$$\widehat{f_k}(u_1^1, u_2^1, u_3^1, \ldots, u_1^k, u_2^k, u_3^k) = \begin{cases} b, & \text{if } a \in \{u_1^i, u_2^i, u_3^i\} \text{ for each } i, \\ 0, & \text{otherwise.} \end{cases}$$

Obviously the values of the $f_k$'s can be computed by a single Turing machine in $O(k)$ time.

Now we define $\mathbf{A}$ to be $(L \times U; +_{\mathbf{A}}, \{f_k\}_{k=1}^{\infty})$, where $(L; +)$ and $(U; \oplus)$ are the groups of order $q$ and $p$, respectively, and $(l_1, u_1) +_{\mathbf{A}} (l_2, u_2) = (l_1 + l_2, u_1 \oplus u_2)$.

To see that $\textsc{Csat}_{\textsc{tm}}(\mathbf{A})$ is NP-complete observe that a 3-CNF formula

$$(\ell_1^1 \vee \ell_2^1 \vee \ell_3^1) \wedge \cdots \wedge (\ell_1^k \vee \ell_2^k \vee \ell_3^k) \tag{9}$$

is satisfiable if and only if the following equation has a solution in $\mathbf{A}$

$$f_k(z_1^1, z_2^1, z_3^1, \ldots, z_1^k, z_2^k, z_3^k) = (b, 0),$$

where $z_i^j = x_i^j$ if $\ell_i^j$ is a positive literal and $z_i^j = (0, a) - x_i^j$ otherwise.

Similarly a formula (9) is not satisfiable iff the following equation holds in $\mathbf{A}$.

$$f_k(z_1^1, z_2^1, z_3^1, \ldots, z_1^k, z_2^k, z_3^k) = (0, 0).$$

This shows co-NP-completeness of $\textsc{Ceqv}_{\textsc{tm}}(\mathbf{A})$. ◀

Note here that the examples with co-NP-complete $\textsc{Ceqv}_{\textsc{tm}}$ and NP-complete $\textsc{Csat}_{\textsc{tm}}$ do exist even for $p = q$. Actually they are provided (but without detailed description of the input size) in [14]. However, if $p = q$ the resulting algebras must have infinitely many basic operations (as otherwise they would be supernilpotent), while for $p \neq q$ the algebras have finitely generated clone of operations but are presented with infinitely many basic operations only to (artificially) compress the size of the input.

Note also that in fact Theorem 6.1 actually establishes much more than just hardness of $\textsc{Csat}_{\textsc{tm}}$ and $\textsc{Ceqv}_{\textsc{tm}}$. Indeed, these examples show that (unless P = NP) for nilpotent but

not supernilpotent finite algebras one cannot expect polynomial time algorithms for CSAT or CEQV based on small search spaces $S_n$ described in the Introduction. This is because the existence of such search spaces do not depend on the finiteness of the language.

Representing functions by Turing machines gives us a way to compress the input. However one can consider this as a drawback. That is because in this approach we can no longer treat basic operations occurring in the input as parameters in the way we do it for the finite set of operations. This probably denies the intuition behind what should an algorithm parameterized by an algebra mean. That is why $\mathrm{CEQV}_{\mathrm{T}}(\mathbf{A})$ and $\mathrm{CSAT}_{\mathrm{T}}(\mathbf{A})$, as described in the Introduction, appear to be more natural candidates for transferring these problems to the realm with possibly infinitely many basic operations. In contrast to Theorem 6.1 we have the following.

▶ **Theorem 6.2.** *Let* $\mathbf{A} = \mathbf{L} \otimes^F \mathbf{U}$ *with* $\mathbf{L}$ *and* $\mathbf{U}$ *being polynomially equivalent to one dimensional vector spaces over prime fields of different characteristics. Then* $\mathrm{CEQV}_{\mathrm{T}}(\mathbf{A})$ *and* $\mathrm{CSAT}_{\mathrm{T}}(\mathbf{A})$ *are in P.*

**Proof.** First note that from Corollary 3.3 every polynomial over $\mathbf{A}$ can be represented in some $f$-normal form. In view of Lemmas 3.2, 4.1 and 5.1 it suffices to show that obtaining $f$-normal forms of basic operations can be done in time polynomial in size of their tables.

To represent the basic operation $g$ in the form of the right-hand side of (5) we need to compute all the $\lambda_i$'s, $\mu_{\overline{\beta},u}$'s, $\alpha_i$'s and $u_0$ from the table of $g$. For $x \in L \times U$ we will use $\Pi_L(x)$ and $\Pi_U(x)$ to denote the first and second coordinate of $x$.

Now, to compute the $\alpha_i$'s and $u_0$ it suffices to solve the following system of $k+1$ linear equations

$$\bigoplus_{i=1}^{k} \alpha_i u_i \oplus u_0 = \Pi_U(g((0,u_1),\dots,(0,u_k)))\,,$$

where $(u_1,\dots,u_k)$ ranges over the set $\{(0,\dots,0),(1,0,\dots,0),(0,1,0,\dots,0)\dots,(0,\dots,0,1)\} \subseteq U^k$ and $\Pi_U(g((0,u_1),\dots,(0,u_k)))$ can be read from the table of $g$. Similarly, the $\lambda_i$'s are the solutions of the following system of $k$ linear equations

$$\sum_{i=1}^{k} \lambda_i l_i + \Pi_L(g((0,0),\dots,(0,0))) = \Pi_L(g((l_1,0),\dots,(l_k,0)))\,,$$

where again $(l_1,\dots,l_k)$ ranges over the set $\{(1,0,\dots,0),(0,1,0,\dots,0)\dots,(0,\dots,0,1)\} \subseteq L^k$. Finally, the $\mu_{\overline{\beta},u}$'s can be recovered from the following system of linear equations

$$\sum_{(\overline{\beta},u)\in \mathbf{F}_U^k \times U} \mu_{\overline{\beta},u} \cdot f\left(\bigoplus_{i=1}^{k} \beta_i u_i \oplus u\right) = \Pi_L(g((0,u_1),\dots,(0,u_k)))\,.$$

This time the system consists of $|U|^k$ equations (one for each $(u_1,\dots,u_k) \in U^k$) but this number is linear in the size of the table of $g$, as $g$ is $k$-ary. ◀

## References


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

2 A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, Oct. 2017. doi:10.1109/FOCS.2017.37.

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

**4** Stanley Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1981.

**5** Tomás Feder, Florent Madelaine, and Iain A. Stewart. Dichotomies for classes of homomorphism problems involving unary functions. *Theoret. Comput. Sci.*, 314(1-2):1–43, 2004.

**6** Ralph Freese and Ralph McKenzie. *Commutator theory for congruence modular varieties*, volume 125 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1987.

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

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

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

**10** Gábor Horváth and Csaba Szabó. The Complexity of Checking Identities over Finite Groups. *Internat. J. Algebra Comput.*, 16(5):931–940, 2006. doi:10.1142/S0218196706003256.

**11** Gábor Horváth and Csaba Szabó. Equivalence and equation solvability problems for the alternating group $A_4$. *Journal of Pure and Applied Algebra*, 216(10):2170–2176, 2012.

**12** Paweł M. Idziak and Jacek Krzaczkowski. Satisfiability in multi-valued circuits. In *2018 Thirty-Third Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2018.

**13** K.A. Kearnes. Congruence modular varieties with small free spectra. *Algebra Universalis*, 42(3):165–181, Oct 1999. doi:10.1007/s000120050132.

**14** Michael Kompatscher. The equation solvability problem over nilpotent mal'cev algebras. *arXiv*, 2017. arXiv:1710.03083.

**15** Benoit Larose and László Zádori. Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. *Internat. J. Algebra Comput.*, 16(3):563–581, 2006.

**16** Jiyou Li and Daqing Wan. On the subset sum problem over finite fields. *Finite Fields and Their Applications*, 14(4):911–929, 2008. doi:10.1016/j.ffa.2008.05.003.

**17** Ju. V. Matijasevič. The Diophantineness of enumerable sets. *Dokl. Akad. Nauk SSSR*, 191:279–282, 1970.

**18** Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

**19** Bernhard Schwarz. The Complexity of Satisfiability Problems over Finite Lattices. In *2004 21st Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, page 31–43, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

**20** D. Zhuk. A proof of CSP dichotomy conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 331–342, Oct. 2017. doi:10.1109/FOCS.2017.38.