Circuit Equivalence in 2-Nilpotent Algebras

Piotr Kawałek
Institute of Discrete Mathematics and Geometry, Vienna University of Technology, Austria
Institute of Computer Science, University of Maria Curie-Skłodowska, Lublin, Poland

Michael Kompatscher
Department of Algebra, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Jacek Krzaczkowski
Institute of Computer Science, University of Maria Curie-Skłodowska, Lublin, Poland

Abstract

The circuit equivalence problem $\textsc{Ceqv}(\mathbf{A})$ of a finite algebra $\mathbf{A}$ is the problem of deciding whether two circuits over $\mathbf{A}$ compute the same function or not. This problem not only generalises the equivalence problem for Boolean circuits, but is also of interest in universal algebra, as it models the problem of checking identities in $\mathbf{A}$. In this paper we prove that $\textsc{Ceqv}(\mathbf{A}) \in \mathsf{P}$, if $\mathbf{A}$ is a finite 2-nilpotent algebra from a congruence modular variety.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Complexity classes

Keywords and phrases circuit equivalence, identity checking, nilpotent algebra

Digital Object Identifier 10.4230/LIPIcs.STACS.2024.45

Funding *Piotr Kawałek*: Funded by the European Union (ERC, POCOCOP, 101071674). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.
This research was funded in whole or in part by National Science Centre, Poland #2021/41/N/ST6/03907. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.
Michael Kompatscher: This work has been supported by projects PRIMUS/24/SCI/008 and UNCE/SCI/022 of Charles University.
Jacek Krzaczkowski: This research was funded in whole or in part by National Science Centre, Poland #2022/45/B/ST6/02229.
For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

Acknowledgements We would like to thank Prof. Paweł Idziak for bringing us together in the spring of 2018 (by inviting the second author to Kraków), which ultimately lead to this publication.

1 Introduction

It is a common problem in mathematics to decide whether two formal expressions are equivalent. Some well-known examples are the word problem for groups and semigroups, checking whether a Boolean formula is a Tautology, or whether two polynomials over a given ring define the same operation. In this paper we study a class of problems that generalize the latter two examples.

In the *polynomial equivalence problem* $\textsc{PolEqv}(\mathbf{A})$ the input consists of two polynomials p and q of the same arity over a finite algebra $\mathbf{A}$, and the task is to decide whether the (universally quantified) identity $p(x_1, \ldots, x_n) \approx q(x_1, \ldots, x_n)$ holds in $\mathbf{A}$. For every fixed finite algebra $\textsc{PolEqv}(\mathbf{A})$ is clearly in $\mathsf{co\text{-}NP}$, since we can verify in polynomial time, whether

41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024).
Editors: Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov; Article No. 45; pp. 45:1–45:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the identity fails at a given tuple $(a_1, \dots, a_n) \in A^n$. So the chief question is to distinguish, for which algebras the problem is hard (i.e. co-NP-complete), tractable (in P), or possibly of some intermediate complexity. There are numerous papers investigating this question for algebras from concrete varieties, such as groups [8, 16, 10, 33, 20], rings [7, 19, 15], and semigroups [27, 4].

However, not much is known in general. One of the major obstacles in studying $\textsc{PolEqv}(\mathbf{A})$ systematically for all finite algebras is that the complexity strongly depends on the language of $\mathbf{A}$. For example, the alternating group $(A_4, \cdot)$ has a polynomial equivalence problem that is in P; but after adding the commutator $[x, y] = x^{-1}y^{-1}xy$ to the signature we obtain the problem $\textsc{PolEqv}((A_4, \cdot, [x, y]))$, which is co-NP-complete (see [18]). Roughly speaking, this follows from the fact that some polynomials in the extended language are exponentially inflated in length, when expressed by only the group operations.

To resolve this problem, in [25] it was proposed to encode an input equation by circuits instead of polynomials. This approach prevents an artificial “inflation” of the input. As a consequence, the complexity then only depends on the clone of polynomial operations of the algebra, which allows for the use of universal algebraic methods. Formally we define the *circuit equivalence problem* $\textsc{Ceqv}(\mathbf{A})$ as follows:

$\textsc{Ceqv}(\mathbf{A})$
Input: Two circuits g_1, g_2 over $\mathbf{A}$ with input gates $x_1, \dots, x_n$
Question: Is $g_1(a_1, \dots, a_n) = g_2(a_1, \dots, a_n)$ for all $(a_1, \dots, a_n) \in A^n$?

In [25], Idziak et al. set the goal to classify the computational complexity of $\textsc{Ceqv}(\mathbf{A})$ for all algebras from congruence modular varieties. On one hand, such a classification would subsume many of the previously known results (e.g. for groups [17], rings [19] and lattices [31, 12]). On the other hand, congruence modular varieties offer a structural advantage, since tools from tame congruence theory and commutator theory work particularly well in them. As it turns out, the complexity of $\textsc{Ceqv}$ in the congruence modular case is strongly linked to commutator theoretical properties. By results contained in [25] and [23] it is known that $\textsc{Ceqv}$ for non-nilpotent algebras from congruence modular variety is co-NP-complete. On the other hand, it was shown in [3] that $\textsc{Ceqv}$ for supernilpotent algebras from congruence modular varieties is in P.

Since in congruence modular varieties supernilpotence implies nilpotence (see e.g. [30]), results mentioned above leave only a gap for nilpotent, but not supernilpotent algebras (Problem 2 in [25]). It was shown in [22], [28], that, under the assumption of the Exponential Time Hypothesis, the complexity of $\textsc{Ceqv}(\mathbf{A})$ has quasipolynomial lower bounds $\Omega(2^{c(\log n)^{k-1}})$, if $\mathbf{A}$ is nilpotent and of supernilpotent rank k (where the supernilpotent rank, introduced in [22], is one of the generalizations of group-theoretical Fitting length notion). On the other hand, under the assumption of an open conjecture in circuit complexity theory, for every nilpotent but not supernilpotent algebra there actually is an algorithm solving $\textsc{Ceqv}$ that has quasipolynomial running time [29]. These two conditional results indicate that nilpotent algebras of supernilpotent rank greater than 2 have coNP-intermediate complexities. Interestingly, this mirrors the situation for the polynomial equivalence problem $\textsc{PolEqv}(\mathbf{G})$ for solvable groups $\mathbf{G} = (G, \cdot, e, ^{-1})$ of Fitting length k [33, 20].

As one can observe, many of recent results connected with complexity of $\textsc{PolEqv}$ and $\textsc{Ceqv}$ are obtained under assumptions of some known hypotheses. For example, hardness results from [22], [20], [28] and [33] are proved under the assumption of Exponential Time Hypothesis (or its randomized version). On the other hand upper bounds for algorithms complexity are often obtained under the assumption of some conjectures (e.g. Strong

Exponential Size Hypothesis, Constant Degree Hypothesis), which assume lower bounds for the size of circuits computing AND. Such results can be found e.g. in [22], [24], [20] and [29]. In contrast, this paper provides unconditional results.

Our paper is structured as follows: In Section 2, we introduce some standard notation and definitions. Section 3 contains basic structural results about 2-nilpotent algebras. In Section 4 we prove that all operations $f\colon U^n \to L$ between a cyclic group $(U,+) = \mathbb{Z}_{p^k}$ of prime power order and a coprime $(L,+)$ are already generated by all unary functions $g\colon U \to L$ (and the addition on U and L). It provides us with a useful normal form for all the functions of type $U^n \to L$. In Section 5 we prove that we can check in polynomial time, whether such a normal form induces a constant function. Results of Sections 4 and 5 might be of independent interest in the study of linearly closed clonoids.

Section 6 then contains the proof that $\textsc{Ceqv}(\mathbf{A}) \in \mathsf{P}$ for 2-nilpotent $\mathbf{A}$ from congruence modular varieties. Our algorithm mixes the two prevalent approaches for checking equivalence, as it first partially restricts the domain of a given circuit (as in [3]), while then doing a syntactic manipulation on the resulting circuits (as in [21]). The latter part is based on the algorithm from Section 5.

2 Preliminaries

In this paper, small bold letters always denote tuples. For instance, tuples of constants are denoted $\mathbf{a} = (a_1, \dots, a_n) \in A^n$ and tuples of variables are denoted $\mathbf{x} = (x_1, \dots, x_n)$. We are going to use standard notation and definitions from universal algebra (see e.g. [6]). We define *algebra type* (or *algebra signature*) to be a sequence of function symbols together with a corresponding arity for each symbol. For a signature F, an algebra $\mathbf{A}$ over F is a pair $(A, F^{\mathbf{A}})$, where A is a set (the *universe* of $\mathbf{A}$) and $F^{\mathbf{A}} = (f^{\mathbf{A}})_{f\in F}$ is a family of finitary operations $f^{\mathbf{A}}\colon A^{\mathrm{ar}(f)} \to A$. Each $f^{\mathbf{A}}$ is called a *basic operation* of $\mathbf{A}$. Sometimes we are not going to distinguish between the basic operation $f^{\mathbf{A}}$ and the corresponding function symbol f, but this should never cause confusion. We say $\mathbf{A}$ is a *finite algebra* if it has a finite universe A and finitely many basic operations. By $\mathrm{ar}(\mathbf{A})$ we denote the maximal arity of the basic operations in $\mathbf{A}$.

An operation that can be constructed by composing basic operations is called a *term operation* of $\mathbf{A}$. If also constants from A are allowed in its construction we call it a *polynomial operation* of $\mathbf{A}$. If for example $\mathbf{A} = (A, +, 0, -, \cdot)$ is a ring, its polynomial operations are exactly the polynomial operations over the ring in the traditional sense. The clone of all polynomial operations of $\mathbf{A}$ is denoted by $\mathrm{Pol}(\mathbf{A})$. We say that $\mathbf{B}$ and $\mathbf{A}$ are *polynomially equivalent* iff there exists an algebra $\mathbf{B}'$ isomorphic to $\mathbf{B}$ with $\mathrm{Pol}(\mathbf{A}) = \mathrm{Pol}(\mathbf{B}')$.

A properly formed string defining a polynomial operation is called a *polynomial* over $\mathbf{A}$ (e.g. if $\mathbf{A} = (A, f^{\mathbf{A}}, g^{\mathbf{A}})$ such that f is ternary and g is binary, the expression $p(x,y,z) = g(g(x,a), f(x,x,y))$ for $a \in A$ is a polynomial over $\mathbf{A}$). It might seem that polynomials are the most natural way of encoding polynomial operations, however circuits offer some advantages. A *circuit* $p(x_1, \dots, x_n)$ over $\mathbf{A}$ is a finite directed acyclic graph, such that

- all the vertices of in-degree 0 (*fan-in* 0) are labeled by a variable x_i (*input gates*), or a constant from A (*constant gates*),
- all other vertices (*gates*) are labeled by a basic operation f of $\mathbf{A}$, and an enumeration of the $\mathrm{ar}(f)$-many incoming edges (thus fan-in must be $\mathrm{ar}(f)$).

The vertices with no outgoing edge are called *output-gates*. In this paper we will only consider circuits over $\mathbf{A}$ with one output gate; such circuits also naturally encode the polynomial operations of $\mathbf{A}$. Two circuits (or polynomials) $p(x_1, \dots, x_n), q(x_1, \dots, x_n)$ over

$\mathbf{A}$ are equivalent if they compute the same polynomial operation over $\mathbf{A}$. For short, let us then write $p(x_1,\dots,x_n) \approx q(x_1,\dots,x_n)$. Note that polynomials can be considered as those circuits, whose underlying digraph is a tree. Thus PolEqv($\mathbf{A}$) reduces to Ceqv($\mathbf{A}$).

As pointed out in [25] circuits are well suited to discuss computational problems in universal algebra, by the following folklore result:

▶ **Lemma 2.1.** *Let $\mathbf{A}$ and $\mathbf{B}$ be two finite algebras with the same universe. If $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$, then every circuit c_1 over $\mathbf{A}$ can be rewritten in logspace into an equivalent circuit c_2 over $\mathbf{B}$, so that c_1 and c_2 compute the same function.*

In particular, this implies that whenever $\mathrm{Pol}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{B})$, the problem Ceqv($\mathbf{A}$) reduces to Ceqv($\mathbf{B}$) in polynomial time.

3 The structure of 2-nilpotent algebras

In this section we discuss 2-nilpotent algebras from congruence modular varieties and the structure of their polynomial clones. In general, nilpotent algebras can be defined by having a finite central series of congruences, where centrality is defined via the so-called term condition:

▶ **Definition 3.1.** Let $\mathbf{A}$ be an algebra. For congruences $\alpha, \beta, \gamma \in \mathrm{Con}(\mathbf{A})$ we say that α *centralizes β modulo γ* (and write $C(\alpha,\beta;\gamma)$) if and only if for all polynomials $p(\mathbf{x},\mathbf{y}) \in \mathrm{Pol}(\mathbf{A})$, and all tuples $\mathbf{a},\mathbf{b} \in A^n$, $\mathbf{c},\mathbf{d} \in A^m$, such that $a_i \sim_\alpha b_i$ for $i = 1,\dots,n$ and $c_j \sim_\beta d_j$ for $j = 1,\dots,m$, the implication

$$\begin{aligned} &p(\mathbf{a},\mathbf{c}) \sim_\gamma p(\mathbf{a},\mathbf{d}) \\ \Rightarrow\ &p(\mathbf{b},\mathbf{c}) \sim_\gamma p(\mathbf{b},\mathbf{d}) \end{aligned}$$

holds.

An algebra $\mathbf{A}$ is called *n-nilpotent* if there is a central series of length n, i.e. a series of congruences $0_A = \alpha_0 \leq \alpha_1 \leq \cdots \leq \alpha_n = 1_A$, such that $C(\alpha_{i+1}, 1_A; \alpha_i)$ for $i = 0,\dots,n-1$. An algebra $\mathbf{A}$ is called *Abelian*, if it is 1-nilpotent, and is called *nilpotent* if it is n-nilpotent, for some natural number $n > 0$.

We refer to [11] for further background on commutator theory. For our purposes we however do not need this original definition of nilpotence, since in congruence modular varieties we have equivalent characterizations by properties of the polynomial clone. For Abelian algebras, a classical result of Herrmann states the following:

▶ **Theorem 3.2** ([14]). *Let $\mathbf{A}$ be an algebra from a congruence modular variety. Then $\mathbf{A}$ is Abelian if and only if it is polynomially equivalent to a module.*

Here R-modules are considered as algebras $(A,+,0,-,(r)_{r\in R})$, where every scalar $r \in R$ is identified with the unary operation $r(x) = r \cdot x$. Abelian algebras from congruence modular varieties are also called *affine*, since their polynomial operations are exactly the affine operations of some module.

2-nilpotent algebras from congruence modular varieties can be characterized as a special kind of wreath product (in the sense of [32]) of two affine algebras, which is defined as follows:

▶ **Definition 3.3.** Let $\mathbf{U}$ and $\mathbf{L}$ be two affine algebras of the same type F, and let $\widehat{F} = (\hat{f})_{f\in F}$ be a family of operations $\widehat{f}\colon U^k \to L$ such that the arity k of $\widehat{f}$ is the arity of the corresponding operation symbol $f \in F$. We then define $\mathbf{L} \otimes^{\widehat{F}} \mathbf{U}$ as the algebra of type F with universe $L \times U$ and basic operations

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

If $\widehat{F}$ is clear from the context, we also write $\mathbf{L} \otimes \mathbf{U}$.

By a result of Freese and McKenzie the following holds:

▶ **Theorem 3.4** (Corollary 7.2. in [11]). *An algebra* $\mathbf{A} = (A, F^{\mathbf{A}})$ *from a congruence modular variety is 2-nilpotent if and only if there are two affine algebras* $\mathbf{U}$, $\mathbf{L}$ *of type* F*, and a set* $\widehat{F}$ *such that* $\mathbf{A} \cong \mathbf{L} \otimes^{\widehat{F}} \mathbf{U}$.

In the following we are often going to identify 2-nilpotent $\mathbf{A}$ with such a wreath product, and write $\mathbf{A} = \mathbf{L} \otimes \mathbf{U}$ for short. We remark however, that this representation of $\mathbf{A}$ is in general not unique. Given a wreath product representation, we can use it to construct its polynomial expansion with some additional nice properties:

▶ **Lemma 3.5.** *For every finite 2-nilpotent algebra* $\mathbf{A}'$ *from a congruence modular variety there exists a finite 2-nilpotent* $\mathbf{A} = \mathbf{L} \otimes^{\widehat{F}} \mathbf{U}$ *such that*

1. $\mathrm{Pol}(\mathbf{A}') \subseteq \mathrm{Pol}(\mathbf{A})$*;*
2. $\mathbf{A}$ *contains Abelian group operations* $+, 0, -$*;*
3. *all other basic operations* $f^{\mathbf{A}}$ *of* $\mathbf{A}$ *are either*
   - *"scalar multiplications"* $f^{\mathbf{A}}((l,u)) = (\lambda \cdot l, 0)$ *or* $f^{\mathbf{A}}((l,u)) = (0, \rho \cdot u)$*, with respect to the modules equivalent to* $\mathbf{L}$ *and* $\mathbf{U}$*,*
   - *or of "hat type"* $f^{\mathbf{A}}((l_1,u_1),\dots,(l_k,u_k)) = (\widehat{f}(u_1,\dots,u_k), 0)$*.*

Proof. By Theorem 3.4 we know that $\mathbf{A}'$ is equal to a wreath product $\mathbf{L}' \otimes^{\widehat{F'}} \mathbf{U}'$ such that $\mathbf{L}'$ and $\mathbf{U}'$ are polynomially equivalent to two modules $(L, +, 0, -, (\lambda)_{\lambda \in R_L})$ and $(U, +, 0, -, (\rho)_{\rho \in R_U})$.

We define $\mathbf{A}$ also to have the universe $L \times U$. The group operations of $\mathbf{A}$ are defined by $(l_1,u_1) + (l_2,u_2) = (l_1 + l_2, u_1 + u_2)$, $0 = (0,0)$ and $-(l,u) = (-l,-u)$. We further define the "scalar multiplications" on $\mathbf{A}$ by $f^{\mathbf{A}}((l,u)) = (\lambda \cdot l, 0)$ for all $\lambda \in R_L$ and $f^{\mathbf{A}}((l,u)) = (0, \rho \cdot u)$ for all $\rho \in R_U$. Finally, for every $\widehat{f} \in \widehat{F'}$ we introduce a basic operation of "hat type" $(\widehat{f}(u_1,\dots,u_k), 0)$ in $\mathbf{A}$.

Note that $\mathbf{A}$ is polynomially richer than $\mathbf{A}'$, since every basic operation $f^{\mathbf{A}'}((l_1,u_1),\dots,(l_k,u_k)) = (c + \sum_{i=1}^{k} \lambda_i l_i + \widehat{f}(u_1,\dots,u_k), d + \sum_{i=1}^{k} \rho_i u_i)$ of $\mathbf{A}'$ is also a polynomial operation of $\mathbf{A}$. Moreover, $\mathbf{A}'$ is finite, and 2-nilpotent by Theorem 3.4. ◀

For short, let us call the algebra $\mathbf{A}$ given by Lemma 3.5 a *group coordinatization* of $\mathbf{A}'$. A similar construction for arbitrary nilpotent algebras (from congruence modular varieties) was discussed in [1, Theorem 4.2].

By Lemma 2.1 and Lemma 3.5 it is enough to prove that the circuit equivalence problem of every 2-nilpotent group coordinatization is in P in order to prove it for all finite 2-nilpotent algebras from congruence modular varieties. The main advantage of working in a group coordinatization $\mathbf{A} = \mathbf{L} \otimes \mathbf{U}$ is, that every circuits/polynomials can be rewritten easily, by simplifying linear combinations over the modules $\mathbf{U}$, $\mathbf{L}$, and observing that the composition of two or more operations of "hat type" is always trivial:

▶ **Observation 3.6.** Let $\mathbf{A} = \mathbf{L} \otimes \mathbf{U}$ be a two nilpotent group coordinatisation. If, for a circuit $p(x_1,\dots,x_n)$ over $\mathbf{A}$ we identify every variable with $x_i = (l_i, u_i)$, then p can be rewritten in polynomial time to an expression

$$p^{\mathbf{A}}((l_1,u_1),\dots,(l_k,u_k)) = \left(p^{\mathbf{L}}(l_1,\dots,l_k) + \widehat{p}(u_1,\dots,u_k), p^{\mathbf{U}}(u_1,\dots,u_k)\right),$$

where $p^{\mathbf{L}}$ and $p^{\mathbf{U}}$ are affine combinations over the modules $\mathbf{L}$ or $\mathbf{U}$ respectively, and $\widehat{p}(u_1,\dots,u_n)$ is a sum of expressions of the form $\lambda\widehat{f}(\sum_{i=1}^{k} \rho_{1,i} u_i + c_1, \dots, \sum_{i=1}^{k} \rho_{1,m} u_i + c_m)$, such that $\widehat{f} \in \widehat{F}$, all $\rho_{i,j}$ are scalars of $\mathbf{U}$, $c_i \in U$, and λ is a scalar of $\mathbf{L}$.

Note that the expressions $\widehat{p}$ in Observation 3.6 are formed by closing the basic operations $\widehat{f} \in \widehat{F}$ under affine combinations in $\mathbf{U}$ (from the inside) and $\mathbf{L}$ (from the outside). In the language of [9] the induced functions $\widehat{p}\colon U^n \to L$ form the $(\mathbf{L}, \mathbf{U})$-*linearly closed clonoid*, which is generated by the operations $\widehat{F}$ (and their translations by constants).

Now clearly $p^{\mathbf{A}}$ is constant, if and only if the operations given by $p^{\mathbf{L}}$, $p^{\mathbf{U}}$ and $\widehat{p}$ are all constant. Since this task is easy to decide for the affine combinations $p^{\mathbf{L}}$ and $p^{\mathbf{U}}$, in this paper we focus mainly on the analysis of the functions $\widehat{p}\colon U^n \to L$.

In the special case that a finite nilpotent algebra $\mathbf{A}$ from a modular variety is additionally of prime power size, it has several nice additional properties. In particular any such algebra is *supernilpotent*, see e.g. [3, 30] for background. We are going to use the following result for such prime power size algebras:

▶ **Theorem 3.7** ([3, 1]). *Assume that* $\mathbf{A}$ *is a nilpotent algebra from a congruence modular variety that is finite and of prime power size. Then, there is a constant* $C \leq \mathrm{ar}(\mathbf{A})(|A|-1)^{\log_2 |A|-1}$ *such that, for every polynomial* $p(x_1, \dots, x_n)$ *and any constant* $0 \in A$*:*

$$p(\mathbf{x}) \approx 0 \Leftrightarrow p(\mathbf{a}) = 0 \text{ for all } \mathbf{a} \in A^n(C, 0),$$

where $A^n(C,0) := \{(a_1, \dots, a_n) \in A^n \colon |\{i \colon a_i \neq 0\}| \leq C\}$.

Theorem 3.7 was used in [3] to prove that PolEqv $\mathbf{A}$ is in P for supernilpotent algebras. This result implies also the existence of the polynomial time algorithm solving Ceqv($\mathbf{A}$).

4 A result on linearly closed clonoids

In this section we analyse functions $\widehat{p}\colon U^n \to L$ between Abelian groups $(U, +)$ and $(L, +)$ of coprime orders. We show that, in some cases, the unary functions between U and L already generate all such functions. In order to state our results, let us introduce the following notation:

▶ **Notation 4.1.** Let p be a prime and k be a natural number. Then, for two tuples $\mathbf{b} = (b_1, \dots, b_n), \mathbf{u} = (u_1, \dots, u_n) \in (\mathbb{Z}_{p^k})^n$ we are going to use the notation $\mathbf{b} \odot \mathbf{u} = \sum_{i=1}^n b_i \cdot u_i$ for the "inner product" of the two tuples in the ring $\mathbb{Z}_{p^k}$.

For $\mathbf{U} = \mathbb{Z}_{p^k}$, let us call a tuple $\mathbf{b} \in U^n$ *non-degenerate*, if one of its entries is a multiplicative invertible element of U. Furthermore, let us call a non-degenerate tuple *normalized*, if the first invertible element in $\mathbf{b}$ is equal to 1 and let us write $(U^n)^*$ for the set of all normalized tuples.

Note that, for a fixed tuple $\mathbf{b} \in U^n$ the map $\mathbf{u} \mapsto \mathbf{b} \odot \mathbf{u}$ is an affine operation and equal to a polynomial of $(U, +)$. In other words, the group $(U, +)$ can be regarded as a module over the ring $(U, +, \cdot)$. This explains the slight abuse of notation in the following, in which we use the inner product $\mathbf{b} \odot \mathbf{u}$, although talking about operations of the Abelian group $(U, +)$.

▶ **Theorem 4.2.** *Let* $(U, +) = \mathbb{Z}_{p^k}$ *for a prime power* p^k *and let* $(L, +)$ *be an Abelian group of order coprime to* $|U|$*. Then, for every function* $f\colon U^n \to L$ *there are unary functions* $m_{\mathbf{b}}\colon U \to L$ *for all* $\mathbf{b} \in (U^n)^*$ *such that*

$$f(\mathbf{u}) = \sum_{\mathbf{b} \in (U^n)^*} m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}). \tag{1}$$

Let us call (1) *a* normal form *of* f.

Using the terminology from [9], Theorem 4.2 says that the set of all operations $f\colon U^n \to L$ is the $((U,+),(L,+))$-linearly closed clonoid generated by all unary functions from U to L. Before we prove Theorem 4.2, note that the existence of a normal form for f is equivalent to the existence of a representation as sum $f(\mathbf{u}) = \sum_{\mathbf{b}\in U^n} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u})$, in which the coefficients range over *arbitrary* $\mathbf{b} \in U^n$. This follows directly from the fact that every $\mathbf{a} \in U^n$ can be uniquely written as $\mathbf{a} = c\cdot\mathbf{b}$, for $\mathbf{b} \in (U^n)^*$ and $c \in U$. Thus, if we define $m'_{\mathbf{b}}(u) = \sum_{c\in U} m_{c\mathbf{b}}(c\cdot u)$, for every $\mathbf{b} \in (U^n)^*$ we obtain a normal form $f(\mathbf{u}) = \sum_{\mathbf{b}\in(U^n)^*} m'_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u})$.

Proof of Theorem 4.2. We first show by induction on $n = 2,3,\dots$ that the existence of normal forms for all binary functions $f\colon U^2 \to L$ implies that also all n-ary function $f\colon U^n \to L$ have a normal form.

For $n = 2$ this is trivial. For an induction step $n \to n+1$, let $f\colon U^{n+1} \to L$ be an $n+1$-ary function. Then, for every $a \in U$, by induction hypothesis, there exist unary functions $m_{a,\mathbf{b}}\colon U \to L$ such that $f(\mathbf{u},a) = \sum_{\mathbf{b}\in U^n} m_{\mathbf{b},a}(\mathbf{b}\odot\mathbf{u})$. For every $\mathbf{b} \in U^n$, we can then define the binary function $s_{\mathbf{b}}(u,v) = m_{\mathbf{b},v}(u)$. By our assumption, every $s_{\mathbf{b}}$ has a normal form. Thus also

$$f(\mathbf{u},u_{n+1}) = \sum_{\mathbf{b}\in U^n} s_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u},u_{n+1})$$

has a normal form, which can be computed by substituting every binary $s_{\mathbf{b}}$ by its normal form and simplifying the resulting sum. This finishes the proof of our claim.

By the above, it is enough to prove the lemma for arity $n = 2$. Without loss of generality we can assume that $(L,+) = \mathbb{Z}_m$ is also a cyclic group (otherwise we take a direct decomposition $(L,+) \cong \prod_{i=1}^k (L_i,+)$ into cyclic groups. Then clearly f has a normal form, if all projections $\pi_i f$ have a normal form).

Note that it is further enough to prove that the function

$$w(u_1,u_2) := w_{(0,0)}(u_1,u_2) = \begin{cases} 1 \text{ if } (u_1,u_2) = (0,0) \\ 0 \text{ else,} \end{cases}$$

has a normal form. If this is the case, then all other binary functions $f\colon U^2 \to L$ also have a normal form by the equation $f(u_1,u_2) = \sum_{a_1,a_2\in U} f(a_1,a_2)\cdot w(u_1-a_1,u_2-a_2)$.

We prove that w has a normal form by induction on the exponent k of the prime power p^k. If $k = 1$, then note that

$$p\cdot w(u_1,u_2) = \sum_{i=0}^{p-1} w_0(u_1+iu_2) - \sum_{j=1}^{p-1} w_0(j+u_2).$$

where

$$w_0(u) = \begin{cases} 1 \text{ if } u = 0 \\ 0 \text{ else.} \end{cases}$$

This was shown before in [2, Lemma 5.3], and can easily be verified by a case distinction. Since L is coprime to U, p has a multiplicative inverse p^{-1} in L, and thus $w(u_1,u_2) = p^{-1}(\sum_{i=0}^{p-1} w_0(u_1+iu_2) - \sum_{j=1}^{p-1} w_0(j+u_2))$, which can be rewritten to a normal form.

For an induction step $k-1 \to k$, let us first define the auxiliary function

$$t(u_1,u_2) = \sum_{i=0}^{p^k-1} w_0(u_1+iu_2) + \sum_{i=0}^{p^{k-1}-1} w_0(piu_1+u_2).$$

We claim that $t(u_1, u_2) = p^k w(u_1, u_2) + \min(\frac{p^{k-1}}{|pu_1|}, \frac{p^{k-1}}{|pu_2|})$, where $|u|$ denotes the order of the group element $u \in U$. To prove this claim, note that for the two sums defining t we have:

$$\sum_{i=0}^{p^k-1} w_0(u_1 + iu_2) = \begin{cases} \frac{p^k}{|u_2|} \text{ if } |u_2| \geq |u_1| \\ 0 \text{ else,} \end{cases} \quad \text{and} \quad \sum_{i=0}^{p^{k-1}-1} w_0(piu_1 + u_2) = \begin{cases} \frac{p^{k-1}}{|pu_1|} \text{ if } |pu_1| \geq |u_2| \\ 0 \text{ else.} \end{cases}$$

Observe further that $u \neq 0$ is equivalent to $1 < |u| = p \cdot |pu|$. Thus, if $u_1 \neq 0$, then $t(u_1, u_2) = \frac{p^{k-1}}{|pu_2|}$ for $|u_2| \geq |u_1|$ and $t(u_1, u_2) = \frac{p^{k-1}}{|pu_1|}$ for $|u_2| < |u_1|$; in other words $t(u_1, u_2) = \min(\frac{p^{k-1}}{|pu_1|}, \frac{p^{k-1}}{|pu_2|})$. If $u_1 = 0$ and $u_2 \neq 0$, then $t(u_1, u_2) = \frac{p^k}{|u_2|} = \min(p^{k-1}, \frac{p^{k-1}}{|pu_2|})$. Finally if $u_1 = 0$ and $u_2 = 0$, then $t(u_1, u_2) = p^k + p^{k-1} = p^k + \min(p^{k-1}, p^{k-1})$.

Thus we have verified that $t(u_1, u_2) = p^k w(u_1, u_2) + \min(\frac{p^{k-1}}{|pu_2|}, \frac{p^{k-1}}{|pu_1|})$. If we define $r\colon (pU)^2 \to L$ as the function $r(pu_1, pu_2) = \min(\frac{p^{k-1}}{|pu_2|}, \frac{p^{k-1}}{|pu_1|})$, then, by the induction assumption (and $pU \cong \mathbb{Z}_{p^{k-1}}$) r has a normal form. Since also t has (by definition) a normal form, it follows that $w(u_1, u_2) = p^{-k} \cdot (t(u_1, u_2) - r(pu_1, pu_2))$ has a normal form. This finishes the proof. ◀

As a direct consequence of Theorem 4.2 we obtain the following version of it for direct products:

▶ **Corollary 4.3.** *Let $(U, +) = \mathbb{Z}_{p^k}$ for a prime power p^k, and $(L, +)$ be an Abelian group of coprime order. Then, for any set V and any function $f\colon U^n \times V \to L$ there are functions $m_{\mathbf{b}}\colon U \times V \to L$ for all $\mathbf{b} \in U^n$ such that $f(\mathbf{u}, v) = \sum_{\mathbf{b} \in (U^n)^*} m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}, v)$.*

Proof. For any fixed value $a \in V$, there is a normal form $f(\mathbf{u}, a) = \sum_{\mathbf{b} \in (U^n)^*} m_{\mathbf{b},a}(\mathbf{b} \odot \mathbf{u})$ by Theorem 4.2. Thus the functions $m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}, v) = m_{\mathbf{b},v}(\mathbf{b} \odot \mathbf{u})$ give us the above normal form. ◀

This corollary is in particular of interest, if we consider the direct products of cyclic groups $\mathbb{Z}_{p^k}$. Let us then use the following notation:

▶ **Notation 4.4.** For a list of prime powers $\mathbf{p} = (p_1^{k_1}, p_2^{k_2}, \dots, p_m^{k_m})$, let us define the ring $\mathbb{Z}_{\mathbf{p}} = \prod_{i=1}^m (\mathbb{Z}_{p_i^{k_i}})$. Moreover, for a list of positive integers $\mathbf{n} = (n_1, n_2, \dots, n_m)$, let us define the $\mathbf{n}$-th power $\mathbb{Z}_{\mathbf{p}}$ as $(\mathbb{Z}_{\mathbf{p}})^{\mathbf{n}} = \prod_{i=1}^m (\mathbb{Z}_{p_i^{k_i}})^{n_i}$. For short, let us also write $\mathbb{Z}_{\mathbf{p}}$ for $\mathbb{Z}_{\mathbf{p}}^{(1,1,\dots,1)}$ and $\mathbb{Z}_{\mathbf{p}}^{\mathbf{n}}$ for $(\mathbb{Z}_{\mathbf{p}})^{\mathbf{n}}$. For every index $i = 1, \dots, m$, let $\mathbf{u}^{(i)}$ denote the projection of $\mathbf{u} \in \mathbb{Z}_{\mathbf{p}}^{\mathbf{n}}$ to $(\mathbb{Z}_{p_i^{k_i}})^{n_i}$.

For two tuples $\mathbf{b}, \mathbf{u} \in \mathbb{Z}_{\mathbf{p}}^{\mathbf{n}}$ we define their “inner product”

$$\mathbf{b} \odot \mathbf{u} = (\mathbf{b}^{(1)} \odot \mathbf{u}^{(1)}, \mathbf{b}^{(2)} \odot \mathbf{u}^{(2)}, \dots, \mathbf{b}^{(m)} \odot \mathbf{u}^{(m)}) \in \mathbb{Z}_{\mathbf{p}}.$$

Note that, for a fixed $\mathbf{b} \in \mathbb{Z}_{\mathbf{p}}^{\mathbf{n}}$ the map $\mathbf{u} \mapsto \mathbf{b} \odot \mathbf{u}$ is a linear map from $\mathbb{Z}_{\mathbf{p}}^{\mathbf{n}}$ to $\mathbb{Z}_{\mathbf{p}}$. In particular, for $\mathbb{Z}_{\mathbf{p}}^{(n,n,\dots,n)} \cong (\mathbb{Z}_{\mathbf{p}})^n$ it can be considered as an n-ary polynomials of the affine algebra $(\mathbb{Z}_{\mathbf{p}}, +, \pi_1, \dots, \pi_m)$, where $\pi_i((u^{(1)}, \dots, u^{(n)})) = (0, \dots, 0, u^{(i)}, 0, \dots, 0)$.

Let us call a tuple $\mathbf{b} \in \mathbb{Z}_{\mathbf{p}}^{\mathbf{n}}$ *non-degenerate/normalized*, if $\mathbf{b}^{(i)}$ is non-degenerate/normalized for every component $i = 1, \dots, m$, and let us write $(\mathbb{Z}_{\mathbf{p}}^{\mathbf{n}})^*$ for the set of normalized tuples.

▶ **Corollary 4.5.** *Let $(U, +) = \mathbb{Z}_{\mathbf{p}}$ for a list of prime powers $\mathbf{p} = (p_1^{k_1}, p_2^{k_2}, \dots, p_m^{k_m})$ and let $(L, +)$ be an Abelian group of order coprime to $|U|$. Then, for any $\mathbf{n} \in \mathbb{N}^m$ and any $f\colon U^{\mathbf{n}} \to L$ there are functions $m_{\mathbf{b}}\colon U \to L$ for all $\mathbf{b} \in (U^{\mathbf{n}})^*$ such that*

$$f(\mathbf{u}) = \sum_{\mathbf{b} \in (U^{\mathbf{n}})^*} m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}) = \sum_{\mathbf{b} \in (U^{\mathbf{n}})^*} m_{\mathbf{b}}(\mathbf{b}^{(1)} \odot \mathbf{u}^{(1)}, \dots, \mathbf{b}^{(m)} \odot \mathbf{u}^{(m)}).$$

We call this representation a normal form *of f.*

Proof. This follows directly from iteratively applying Corollary 4.3 to all the components of the direct product $\mathbb{Z}_{\mathbf{p}} = \prod_{i\in 1}^{m} \mathbb{Z}_{p_i^{k_i}}$. ◀

At last, we show that for coprime modules $\mathbf{U}$ and $\mathbf{L}$, all functions in a finitely generated $(\mathbf{L},\mathbf{U})$-clonoid can be rewritten into such a normal form in polynomial time. Note here, that whenever some $m_{\mathbf{b}}$ is equal to the constant $\mathbf{0}$ function, we can just skip it in the representation of the normal form. In this way we can avoid summing over the entire $(U^{\mathbf{n}})^*$ (which has exponential size).

▶ **Lemma 4.6.** *Let $\mathbf{U}$ and $\mathbf{L}$ be two finite modules over coprime domains, let $\mathbf{p}$ be list of prime powers $\mathbf{p} = (p_1^{k_1}, p_2^{k_2}, \dots, p_m^{k_m})$ such that $(U,+) = \mathbb{Z}_{\mathbf{p}}$ is the group reduct of $\mathbf{U}$. Let $\widehat{F}$ be a finite set of operations from U to L. Then any n-ary function $\widehat{p}$ in the $(\mathbf{L},\mathbf{U})$-clonoid generated by $\widehat{F}$ can be rewritten in polynomial time into a normal form*

$$\widehat{p}(\mathbf{u}) = \sum_{\mathbf{b}\in X} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u})$$

where $\mathbf{n} = (n,n,\dots,n) \in \mathbb{N}^m$ and $X \subseteq (U^{\mathbf{n}})^*$.

Proof. Recall that any $\widehat{p}(u_1,\dots,u_n)$ in an $(\mathbf{L},\mathbf{U})$-clonoid is a sum of expressions $\lambda\widehat{f}(\sum_{i=1}^k \rho_{1,i}u_i + c_1, \dots, \sum_{i=1}^k \rho_{m,i}u_i + c_m)$, with $\widehat{f} \in \widehat{F}$. It is thus enough to prove, that every such summand can be rewritten into a normal form.

The naive way to do this, would be to substitute every $\widehat{f} \in \widehat{F}$ by its normal form, and simplify the resulting sum. There is however a catch: The ring R of the module $\mathbf{U} = (U,+,0,-,(\rho)_{\rho\in R})$ is possibly different from $(\mathbb{Z}_{\mathbf{p}},+,0,-,\cdot,1)$. This problem can be resolved by computing normal forms for all functions

$$\widehat{f}(\sum_{\rho\in R}\rho u_{1,\rho}, \sum_{\rho\in R}\rho u_{2,\rho}, \dots, \sum_{\rho\in R}\rho u_{n,\rho}), \tag{2}$$

for $\widehat{f} \in \widehat{F}$ and distinct variables $u_{i,\rho}$ for all indices i and coefficients $\rho \in R$.

To rewrite an expression $\lambda\widehat{f}(\sum_{i=1}^k \rho_{1,i}u_i + c_1, \dots, \sum_{i=1}^k \rho_{1,m}u_i + c_m)$, we then collect in every argument of $\widehat{f}$ all variables according to their coefficients from R, and then substitute the normal form for the expression (2). ◀

5 A recursive principle

Let $(U,+) = \mathbb{Z}_{\mathbf{p}}$ and $(L,+)$ be two finite Abelian groups of coprime order. By Corollary 4.5 we know that every function $f\colon U^{\mathbf{n}} \to L$ is equal to the sum of operations $m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u})$. In this section we prove that we can check in polynomial time whether an f given by such a normal form is constant. Our algorithm is based on the fact that a normal form is constant, if and only if it can be partitioned in certain constant subsums, where the partition is formed with respect to the following equivalence relation $\sim$:

▶ **Definition 5.1.** Let $(U,+) = \mathbb{Z}_{p^k}$. Then, for two tuples $\mathbf{a},\mathbf{b} \in (U^n)^*$, let us write $\mathbf{a} \sim \mathbf{b}$, if $\mathbf{a} - \mathbf{b} \in (pU)^n$.

Note that, if $\mathbf{a} \sim \mathbf{b}$, then an entry a_i is invertible (i.e. not a multiple of p) if and only if b_i is invertible.

▶ **Proposition 5.2.** *Let $(U,+) = \mathbb{Z}_{p^k}$ and let $(L,+)$ be of order coprime to U. Let $f\colon U^n \to L$ be an operation given by the normal form*

$$f(\mathbf{u}) = \sum_{\mathbf{b}\in(U^n)^*} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}).$$

Then f is constant if and only if for every $\mathbf{a}\in(U^n)^*$ the sum

$$f_{\mathbf{a}}(\mathbf{u}) = \sum_{\mathbf{b}\in[\mathbf{a}]_\sim} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u})$$

is constant.

Proof. If $f_{\mathbf{a}}$ is constant for all $\mathbf{a}\in(U^n)^*$, then obviously f is also constant, since we can pick the transversal $\mathbf{a}_1,\dots,\mathbf{a}_s$ of $\sim$ and the statement can be inferred from $f(\mathbf{u}) = \sum_{i=1}^{s} f_{\mathbf{a}_i}(\mathbf{u})$.

For the other direction, we first assume that $\mathbf{a} = (1,0,\dots,0)$. Now in the case where $f(\mathbf{u}) = \sum_{\mathbf{b}\in(U^n)^*} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u})$ is constant, we are going to prove an even stronger statement, namely that for every $i = 0,\dots,k$ we have

$$\sum_{\mathbf{b}\in[\mathbf{a}]_\sim}\sum_{c\in p^iU} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}+c) \text{ is constant.} \tag{3}$$

Note that the case $i=k$ in (3) says, that the expression $\sum_{\mathbf{b}\in[\mathbf{a}]_\sim}\sum_{c\in p^iU} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}) = f_{\mathbf{a}}(\mathbf{u})$ represents a constant function.

We prove (3) by induction on $i = 0,1,\dots,k$. For $i=0$, the statement is true, since then the inner sum $\sum_{c\in U} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}+c) = \sum_{c\in U} m_{\mathbf{b}}(c)$ is constant for every $\mathbf{b}$.

For an induction step $i \to i+1$, let us define $C = p^{i+1}U \times (p^iU)^{n-1}$. Then, the sum

$$\sum_{\mathbf{c}\in C} f(\mathbf{u}+\mathbf{c}) = \sum_{\mathbf{b}\in(U^n)^*}\sum_{\mathbf{c}\in C} m_{\mathbf{b}}(\mathbf{b}\odot(\mathbf{u}+\mathbf{c})), \tag{4}$$

is constant, since f is constant.

Note that for every $\mathbf{g}\in(U^n)^*$, which is not equivalent to $\mathbf{a} = (1,0,\dots,0)$, there is an index $j\neq 1$, such that g_j is invertible. Therefore, if we restrict the sum (4) to only summands from the equivalence class of such a $\mathbf{g}$, we obtain

$$\sum_{\mathbf{b}\in[\mathbf{g}]_\sim}\sum_{\mathbf{c}\in C} m_{\mathbf{b}}(\mathbf{b}\odot(\mathbf{u}+\mathbf{c})) = \frac{|C|}{p^{k-i}}\sum_{\mathbf{b}\in[\mathbf{g}]_\sim}\sum_{c_j\in p^iU} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}+c_j),$$

which is constant by induction assumption.

This, together with (4) being constant implies that also

$$\sum_{\mathbf{b}\in[\mathbf{a}]_\sim}\sum_{\mathbf{c}\in C} m_{\mathbf{b}}(\mathbf{b}\odot(\mathbf{u}+\mathbf{c})) = \frac{|C|}{p^{k-i-1}}\sum_{\mathbf{b}\in[\mathbf{a}]_\sim}\sum_{c_1\in p^{i+1}U} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}+c_1)$$

is constant. Since $\frac{|C|}{p^{k-i-1}}$ is a power of p, it has an inverse in L. Thus also $\sum_{\mathbf{b}\in[\mathbf{a}]_\sim}\sum_{c_1\in p^{i+1}U} m_{\mathbf{b}}(\mathbf{b}\odot\mathbf{u}+c_1)$ is constant. This finishes the proof of (3), and therefore also the proof of the proposition in case $\mathbf{a} = (1,0,\dots,0)$.

Now, to make this proof work also for $\mathbf{a}\neq(1,0,\dots,0)$, we need the following.

▷ Claim 1. For a prime p and a natural number k, let $U = \mathbb{Z}_{p^k}$ and let $\mathbf{a}\in(U^n)^*$. There exist two linear maps $T'_{\mathbf{a}}(\mathbf{u})$ and $T_{\mathbf{a}}(\mathbf{u})$ of type $U^n \mapsto U^n$ such that

1. $T_{\mathbf{a}}$ maps $(a_1, \ldots, a_n) \to (1, 0, \ldots, 0)$,
2. both $T'_{\mathbf{a}}$ and $T_{\mathbf{a}}$ are linear bijections from $U^n \to U^n$,
3. function $T_{\mathbf{a}}$ preserves the equivalence relation $\sim$,
4. $\mathbf{d} \odot T'_{\mathbf{a}}(\mathbf{u}) = T_{\mathbf{a}}(\mathbf{d}) \odot \mathbf{u}$

Let j denote the first coordinate such that $a_j = 1$ and let $S_j = \{1.., n\} \setminus \{1, j\}$. One can check that the following definitions of

$$T'_{\mathbf{a}}(\mathbf{u}) = (u_j, u_2, u_3, \ldots, u_{j-1}, u_1 - a_1 u_j - \sum_{i \in S_j} a_i u_i, u_{j+1}, \ldots, u_n)$$

and

$$T_{\mathbf{a}}(\mathbf{d}) = (d_j, d_2 - d_j \cdot a_2, d_3 - d_j \cdot a_3, \ldots, d_{j-1} - d_j \cdot a_{j-1}, d_1 - (d_j) \cdot a_1, d_{j+1} - d_j \cdot a_{j+1}, \ldots, d_n - d_j \cdot a_n)$$

satisfy these four conditions (for $j = 1$ formulas should be interpreted as $T'_{\mathbf{a}}(\mathbf{u}) = (u_1 - \sum_{i=2}^{n} a_i u_i, u_2, \ldots, u_n)$ and $T_{\mathbf{a}}(\mathbf{d}) = (d_1, d_2 - d_1 \cdot a_2, d_3 - d_1 \cdot a_3, \ldots, d_n - d_1 \cdot a_n)$).

Hence, if we now define $f'(\mathbf{u}) := f(T'_{\mathbf{a}}(\mathbf{u}))$ we can see that f' is a function with a normal form defined by $m'_{\mathbf{b}} = m_{(T_{\mathbf{a}})^{-1}(\mathbf{b})}$. This shows, that the Proposition is true for the pair (f, $[\mathbf{a}]_\sim$) iff it is true for the pair (f', $[(1, 0, \ldots, 0)]_\sim$) . This finishes the proof, as we already considered the case when $\mathbf{a} = (1, 0, \ldots, 0)$. ◀

As we work not only with $(U, +) = \mathbb{Z}_{p^k}$, but also with more general groups $(U, +) = \mathbb{Z}_{\mathbf{p}}$, we need to adjust our definitions accordingly.

▶ **Definition 5.3.** For a list of prime powers $\mathbf{p} = (p_1^{k_1}, \ldots, p_m^{k_m})$, let $(U, +) = \mathbb{Z}_{\mathbf{p}}$ and let $\mathbf{n} \in \mathbb{N}^m$. For an $i \in \{1, \ldots, m\}$ let us say that two elements the $\mathbf{a}, \mathbf{b} \in (U^{\mathbf{n}})^*$ are in equivalence relation $\mathbf{a} \sim_i \mathbf{b}$ if and only if they satisfy $\mathbf{a}^{(i)} \sim \mathbf{b}^{(i)}$ in $\mathbb{Z}_{p_i^{k_i}}$.

Here is a very important corollary which is a direct consequence of Proposition 5.2 applied to the direct component $\mathbb{Z}_{p_i^{k_i}}$ of $\mathbb{Z}_{\mathbf{p}}$.

▶ **Corollary 5.4.** *For a list of prime powers $\mathbf{p} = (p_1^{k_1}, \ldots, p_m^{k_m})$, let $(U, +) = \mathbb{Z}_{\mathbf{p}}$ and $(L, +)$ be a finite Abelian group of coprime order. For $\mathbf{n} \in \mathbb{N}^m$ let $f \colon U^{\mathbf{n}} \to L$ be an operation given by a normal form*

$$f(\mathbf{u}) = \sum_{\mathbf{b} \in (U^n)^*} m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}).$$

Then for every $i \in \{1, \ldots, m\}$: function f is constant if and only if for every $\mathbf{a} \in (U^n)^*$ we have that

$$f_{i,\mathbf{a}}(\mathbf{u}) = \sum_{\mathbf{b} \in [\mathbf{a}]_{\sim_i}} m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u})$$

is constant.

Now, we are going to use Corollary 5.4 to check in polynomial time, if a function in a normal form is constant.

▶ **Lemma 5.5.** *For a fixed tuple of prime powers $\mathbf{p} = (p_1^{k_1}, \ldots, p_m^{k_m})$, let $(U, +) = \mathbb{Z}_{\mathbf{p}}$, and $(L, +)$ be a finite Abelian group of coprime orders. Then, for $\mathbf{n} \in \mathbb{N}^m$ and any function $f \colon U^{\mathbf{n}} \to L$ that is given by a normal form*

$$f(\mathbf{u}) = \sum_{\mathbf{b} \in X} m_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}),$$

for some $X \subseteq (U^{\mathbf{n}})^*$, we can decide in time $O(|f|^C)$ whether f is constant or not (with C depending only on $\mathbf{p}$).

Proof. The algorithm, that we are going to present, is based on a recursion. In each recursive call, both the total number of variables on all coordinates (i.e. $n_1 + n_2 + \ldots + n_m$), as well as the sum of all exponents (i.e. $k_1 + \ldots + k_m$) are decreased by at least one.

Note that, in the case when for all $i = 1, ..., m$ we have either $n_i = 0$ or $k_i = 0$, the function f is constant. So first, we pick an appropriate coordinate i with $n_i \neq 0$ and $k_i \neq 0$. Now we take a transversal $\mathbf{a}_1, \ldots, \mathbf{a}_l$ of $\sim_i \cap (X \times X)$. From Corollary 5.4 we know that the function f is constant iff. all the $f_{i,\mathbf{a}}$'s are constant, for all $\mathbf{a} \in \{\mathbf{a}_1, \ldots, \mathbf{a}_l\}$.

Now, we will slightly transform those $f_{i,\mathbf{a}}$'s by applying linear maps to their arguments. We will use linear maps from Claim 1 applied to the i-th component of U, that is: $T'_{\mathbf{a}^{(i)}}(u_1^{(i)}, \ldots, u_{n_i}^{(i)})$, $T_{\mathbf{a}^{(i)}}(d_1^{(i)}, \ldots, d_{n_i}^{(i)})$ in order to define linear maps on the entire U as follows:

$$(T'_{\mathbf{a}}(\mathbf{u}))^{(k)} = \begin{cases} T'_{\mathbf{a}^{(i)}}(u_1^{(i)}, \ldots, u_{n_i}^{(i)}) & \text{if } k = i \\ (u_1^{(k)}, \ldots, u_{n_k}^{(k)}) & \text{otherwise} \end{cases}$$

as well as

$$(T_{\mathbf{a}}(\mathbf{d}))^{(k)} = \begin{cases} T_{\mathbf{a}^{(i)}}(d_1^{(i)}, \ldots, d_{n_i}^{(i)}) & \text{if } k = i \\ (d_1^{(k)}, \ldots, d_{n_k}^{(k)}) & \text{otherwise} \end{cases}$$

Note, that those maps only act on the variables from i-th component of U and keep other variables untouched. Since $T'_{\mathbf{a}}(\mathbf{u})$ is just a permutation of $U^{\mathbf{n}}$, instead of checking that $f_{i,\mathbf{a}}(\mathbf{u})$ is constant we can check that $f'_{i,\mathbf{a}}(\mathbf{u}) = f_{i,\mathbf{a}}(T'_{\mathbf{a}}(\mathbf{u}))$ is constant. Moreover, we can see that $\mathbf{d} \odot T'_{\mathbf{a}}(\mathbf{u}) = T_{\mathbf{a}}(\mathbf{d}) \odot \mathbf{u}$ (like in the Claim 1), so we can actually compute the normal form of each such $f'_{i,a}$, as it is given by the formula:

$$f'_{i,\mathbf{a}}(\mathbf{u}) = \sum_{\mathbf{b} \in X'} m'_{\mathbf{b}}(\mathbf{b} \odot \mathbf{u}),$$

where $m'_{\mathbf{b}} = m_{(T_{\mathbf{a}})^{-1}(\mathbf{b})}$ and $X' = T_{\mathbf{a}}([\mathbf{a}]_{\sim_i} \cap X)$.

In order to check that such created $f'_{i,\mathbf{a}}$'s are constant we will substitute constants $c \in \mathbb{Z}_{p_i^{k_i}}$ for the variable $u_1^{(i)}$ in $f'_{i,\mathbf{a}}$ and recursively check that such created $f'_{i,\mathbf{a}}[u_1^{(i)} = c]$ are constant. Additionally, we have to also make sure, that the returned constants are equal for all different $c \in \mathbb{Z}_{p_i^{k_i}}$. For this purpose, it is enough to assign to all variables the value 0 and check that the set $\{f'_{i,\mathbf{a}}[u_1^{(i)} = c](0, \ldots, 0) : c \in \mathbb{Z}_{p_i^{k_i}}\}$ has size one.

Before the recursive call, we made a substitution for the variable and thus reduced the number of variables (on i-th coordinate) by one. But the effort taken to compute this $f'_{i,\mathbf{a}}$, instead of applying substitutions directly to f, will now provide us with an additional benefit. It turns out, that as a side effect of this substitution, we have also implicitly reduced the size of the domain U. To see it, recall that $T_{\mathbf{a}}(\mathbf{a})^{(i)} = (1, 0, \ldots, 0)$ and $T_{\mathbf{a}}$ preserves the $\sim_i$ relation (by Claim 1). It means that all the $\mathbf{b} \odot \mathbf{u}$ that occur in the normal form of $f'_{i,\mathbf{a}}$ on the i-th coordinate have a very special form: $\mathbf{b}^{(i)} \odot \mathbf{u}^{(i)} = u_1^{(i)} + p_i \cdot (\mathbf{d}_b \odot (u_2^{(i)}, \ldots, u_{n_i}^{(i)}))$, for some $\mathbf{d}_b \in (\mathbb{Z}_{p_i^{k_i}})^{n_i - 1}$. So now, when we substitute constant c for $u_1^{(i)}$, the normal form of $f'_{i,\mathbf{a}}$ transforms into the expression:

$$\sum_{\mathbf{b} \in X'} m'_{\mathbf{b}}(\mathbf{b}^{(1)} \odot \mathbf{u}^{(1)}, \ldots, c + p_i \cdot (\mathbf{d}_b \odot (u_2^{(i)}, \ldots, u_{n_i}^{(i)})), \ldots, \mathbf{b}^{(m)} \odot \mathbf{u}^{(m)}),$$

Here, linear combinations of variables $u_j^{(i)}$ are in fact computable in $\mathbb{Z}_{p_i^{k_i - 1}}$, since $p_i \cdot \mathbb{Z}_{p_i^{k_i}} \equiv \mathbb{Z}_{p_i^{k_i - 1}}$ (where $\equiv$ denotes an additive group isomorphism). So we can just reinterpret the variables to the new domain, and normalize the obtained form, so that now we can recursively check if $f'_{i,\mathbf{a}}[u_1^{(i)} = c]$ is constant, when treated as a function over the domain $U' = (\mathbb{Z}_{p_1^{k_1}}) \times \ldots, \times(\mathbb{Z}_{p_i^{k_i-1}}) \times \ldots \times (\mathbb{Z}_{p_m^{k_m}})$. For completeness, notice, that while going from f to the normalized form of $f'_{i,\mathbf{a}}[u_1^{(i)} = c]$, some variable other than $u_1^{(i)}$ can disappear. To handle it, we can just decrease the number of variables appropriately before the recursive call. The described procedure can be summarized as follows.

Algorithm 1 For a fixed Abelian $(L, +)$, this algorithm takes as input a list of prime powers $\mathbf{p} = (p_1^{k_1}, \ldots, p_m^{k_m})$ coprime to $|L|$, a list of arities $\mathbf{n} = (n_1, \ldots, n_m)$, and checks whether a function $f\colon (\mathbb{Z}_\mathbf{p})^\mathbf{n} \to L$ given by a normal form $f(\mathbf{u}) = \sum_{\mathbf{b} \in X} (\mathbf{b} \odot \mathbf{u})$ is constant.

```
procedure IsConstant($\mathbf{p}, \mathbf{n}$, $f\colon (\mathbb{Z}_\mathbf{p})^\mathbf{n} \to L$)
    if for all $i = 1, \ldots, m$: $n_i = 0$ or $p_i^{k_i} = 1$ then return True
    else
        Let $i$ be the minimal value such that $n_i, k_i \neq 0$
        Let $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s$ be a transversal of $\sim_i$ inside $X \times X$
        for all $\mathbf{a} \in \{\mathbf{a}_1, \ldots, \mathbf{a}_s\}$ do
            for all $c \in \mathbb{Z}_{p_i^{k_i}}$ do
                Compute a normal form of $f'_{i,\mathbf{a}}[u_1^{(i)} = c]$
                Compute new domain $\mathbf{p}'$ and new arities $\mathbf{n}'$
                if $\neg$ IsConstant($\mathbf{p}', \mathbf{n}'$, $f'_{i,\mathbf{a}}[u_1^{(i)} = c]$) then return False
            if $|\{f'_{i,\mathbf{a}}[u_1^{(i)} = c](0, \ldots, 0) \ : c \in \mathbb{Z}_{p_i^{k_1}}\}| \neq 1$ then return False
        return True
```

Now we analyse the running time of the above algorithm. Procedure $\textsc{IsConstant}(\mathbf{p}, \mathbf{n}, f)$ first computes at most $p_i^{k_i}|f|$-many functions $f'_{i,\mathbf{a}}[u_1^{(i)} = c]$, whose normal forms have sizes bounded by $|f|$. To obtain them we need to regroup the normal form of f into $\sim_i$ classes and apply linear map T', which can be done with a naive quadratic algorithm. For the obtained functions, it compares values $f'_{i,\mathbf{a}}[u_1^{(i)} = c](0, \ldots, 0)$, which takes linear time in $|f|$. Moreover, all the normalizations can be done in a linear time. The recursion depth of $\textsc{IsConstant}$ is at most $\sum_{i=1}^m k_i$, thus we obtain a running time of $O(|f|^2 \cdot |f|^{(k_1 + \cdots + k_m)})$.

A careful reader can see, that actually the sum of lengths of expressions that are computed during the runtime of the above algorithm is bounded by a linear function in $|f|$. Using this observation one can prove an even more accurate result, namely that the presented algorithm is in fact quadratic in the worst case. ◀

6 Proof of the main theorem

We are now ready to prove the main theorem:

▶ **Theorem 6.1.** *Let $\mathbf{A}$ be a finite 2-nilpotent algebra from a congruence modular variety. Then we can decide in time $\mathcal{O}(n^C)$ whether an n-ary circuit $p(x_1, \ldots, x_n)$ over $\mathbf{A}$ represents a constant function, where C depends only on $\mathbf{A}$. In particular, this implies that $\textsc{CEqv}(\mathbf{A}) \in \mathsf{P}$.*

Proof. By Lemma 2.1 and Lemma 3.5, we can without loss of generality assume that $\mathbf{A} = \mathbf{L} \otimes^{\widehat{F}} \mathbf{U}$ is a group extension. If we identify every variable x_i of the circuit $p(x_1, \ldots, x_n)$ with a pair $x_i = (l_i, u_i)$ of variables over L and U, then, by Observation 3.6, we can rewrite it in polynomial time to an expression

$$p^\mathbf{A}((l_1, u_1), \ldots, (l_k, u_k)) = \big(p^\mathbf{L}(l_1, \ldots, l_k) + \widehat{p}(u_1, \ldots, u_k), p^\mathbf{U}(u_1, \ldots, u_k)\big)\,,$$

where $p^{\mathbf{L}}(l_1, \dots, l_k) = c + \sum_{i=1}^{k} \lambda_i l_i$ and $p^{\mathbf{U}}(u_1, \dots, u_k) = d + \sum_{i=1}^{k} \rho_i u_i$ are affine combinations in the modules $\mathbf{L}$ and $\mathbf{U}$ respectively, and $\widehat{p}(u_1, \dots, u_n)$ is a sum of expressions $\lambda \widehat{f}(\sum_{i=1}^{k} \rho_{1,i} u_i + c_1, \dots, \sum_{i=1}^{k} \rho_{1,m} u_i + c_m)$, for $\widehat{f} \in \widehat{F}$.

Clearly $p^{\mathbf{A}}$ is constant if and only if $p^{\mathbf{L}}$, $p^{\mathbf{U}}$ and $\widehat{p}$ are constant. For the affine operations $p^{\mathbf{L}}$, $p^{\mathbf{U}}$ we can check this, by simply checking whether all coefficients λ_i and ρ_i are equal to 0. Thus the problem reduces to checking, whether the expression $\widehat{p}(u_1, \dots, u_n)$ defines a constant function.

Since $\mathbf{L}$ is a module, it has a direct decomposition $\mathbf{L} = \prod_{i=1}^{r} \mathbf{L}_i$ into factors of prime power size. If π_i denotes the projection of L to L_i, then $\widehat{p}$ is constant if and only if $\pi_i \circ \widehat{p}$ is constant for every $i = 1, \dots, m$. Thus, without loss of generality, we can assume that the size of $\mathbf{L}$ is a power of some prime q. Also $\mathbf{U}$ can be directly decomposed into $\mathbf{U} = \mathbf{U}_1 \times \mathbf{U}_2$ such that $|U_1|$ is a power of q, and $|U_2|$ is coprime to q. Let us then identify every variable u over U with its direct decomposition $(u^{(1)}, u^{(2)})$ with respect to $U_1 \times U_2$.

Now we want to check that $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ is constant. Note, that $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ is an expression of $(\mathbf{L},\ (\mathbf{U}_1 \times \mathbf{U}_2))$-clonoid. However, by fixing $\mathbf{u}^{(2)}$ to some constant $\mathbf{a}^{(2)} \in (U_2)^n$ we create $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{a}^{(2)})$, which is an expression of $(\mathbf{L}, \mathbf{U}_1)$-clonoid. This clonoid is generated by all the functions $\widehat{f}_{\mathbf{b}^{(2)}}(\mathbf{u}^{(1)}) = \widehat{f}(\mathbf{u}^{(1)}, \mathbf{b}^{(2)})$ and is of prime power order. Hence, we can associate this $(\mathbf{L}, \mathbf{U}_1)$-clonoid with a 2-nilpotent algebra of prime power order (q is the prime here). By Theorem 3.7 there is a set S of polynomial size $O(n^C)$ (and independent of $\mathbf{a}^{(2)}$), such that $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{a}^{(2)})$ is constant iff it is constant on the set S. So now, going back to $(\mathbf{L},\ (\mathbf{U}_1 \times \mathbf{U}_2))$-clonoid, our expression $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ represents a constant function iff all the $\widehat{p}(\mathbf{a}^{(1)}, \mathbf{u}^{(2)})$ represent the same constant function c for all $a^{(1)} \in S$. So, in order to check that $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ is constant, it is enough to pick arbitrary tuple $\mathbf{0} \in \mathbf{U}_2$, check that $\{\mathbf{p}(a^{(1)}, \mathbf{0}) : a^{(1)} \in S\}$ is one element set, and then check that each $\widehat{p}(\mathbf{a}^{(1)}, \mathbf{u}^{(2)})$ is constant.

Since the set S is of polynomial size, this is a polynomial-time Turing reduction from the problem over the $(\mathbf{L},\ (\mathbf{U}_1 \times \mathbf{U}_2))$-clonoid to the problem over the $(\mathbf{L},\ \mathbf{U}_2)$-clonoid, where this $(\mathbf{L},\ \mathbf{U}_2)$-clonoid is generated by all operations $\widehat{f}_{\mathbf{b}^{(1)}}(\mathbf{u}^{(2)}) = \widehat{f}(\mathbf{b}^{(1)}, \mathbf{u}^{(2)})$, for $\widehat{f} \in \widehat{F}$ and $\mathbf{b}^{(1)} \in (U_1)^{\mathrm{ar}(f)}$. Since $|L|$ and $|U_2|$ are coprime, by Lemma 4.6 we know that $\widehat{p}(\mathbf{a}^{(1)}, \mathbf{u}^{(2)})$ can be rewritten into a normal form $\sum_{\mathbf{b}^{(2)} \in X} m_{\mathbf{b}^{(2)}}(\mathbf{b}^{(2)} \odot \mathbf{u}^{(2)})$ in polynomial time (where $X \subseteq ((U_2)^{\mathbf{n}})^*$, with $\mathbf{n} = (\mathrm{ar}(f), \dots, \mathrm{ar}(f))$. By Lemma 5.5 we can check in polynomial time, whether this normal form represents a constant function. Thus we can check in polynomial time whether $\widehat{p}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)})$ is constant.

In order to obtain an algorithm for CEQV($\mathbf{A}$), note that any identity $p \approx q$ is equivalent to $p + (-q) \approx 0$ over $\mathbf{A}$ (recall that $\mathbf{A}$ is a group extension), so CEQV($\mathbf{A}$) can be solved by checking whether $p + (-q)$ is constant and evaluates to 0 at some tuple. ◀

7 Conclusions and open problems

As it was mentioned in the introduction there is a characterization (under assumptions of ETH and CDH) of algebras from a congruence modular variety for which CEQV can be solved in randomized polynomial time. Moreover, we are not far from obtaining a similar characterization of algebras for which CEQV can be solved in deterministic polynomial time. The only case we have to consider to obtain such a characterization is algebras of supernilpotent rank 2, i.e for every algebra $\mathbf{A}$ having supernilpotent congruence α such that $\mathbf{A}/\alpha$ is also supernilpotent. Note that in this paper we show a deterministic polynomial time algorithm for every algebra $\mathbf{A}$ having an abelian congruence α such that $\mathbf{A}/\alpha$ is also abelian. The interesting question is if we can extend our recursive principle to all algebras with supernilpotent rank 2. Note that there are many structural similarities between 2-nilpotent algebras and algebras with supernilpotent rank 2.

This leads us to the following question.

▶ **Problem 1.** *Let* $\mathbf{A}$ *be a finite algebra from congruence modular variety with supernilpotent rank* 2*.*

Is there a deterministic polynomial time algorithm solving CEQV $\mathbf{A}$*?*

Note that the probabilistic algorithm solving CSAT for nilpotent algebras of supernilpotent rank 2 relies on Constant Degree Hypothesis (introduced in [5]), i.e. the conjecture that there are no subexponential size $\mathrm{AND}_d \circ \mathrm{MOD}_m \circ \mathrm{MOD}_p$-circuits computing AND_n function of arbitrarly large arity, where d and m are some constant integers and p is a prime number. Despite the fact that proving CDH will not automatically give us a deterministic polynomial time algorithm solving CEQV for algebras of supernilpotent rank 2, it is hard to believe that such an algorithm can exist in case CDH fails. It leads us to a natural question.

▶ **Problem 2.** *Does Constant Degree Hypothesis hold?*

Although CDH is a quite a long-standing hypothesis, we strongly believe that it holds. It is already proven in some restricted settings, for instance when the number of connections between AND_d gates and MOD_m gates is restricted [13]. Recently, Kawałek and Weiss [26] have shown that if there exist circuits witnessing that CDH fails they have to be non-symmetric.

The natural next step after characterizing algebras from congruence modular varieties for which CEQV can be solved in polynomial time is to study the computational complexity of CEQV for algebras outside congruence modular variety. The most notable example of such algebras are semigroups.

▶ **Problem 3.** *For which semigroups can* CEQV *be solved in (deterministic) polynomial time?*

References

1 Erhard Aichinger. Bounding the free spectrum of nilpotent algebras of prime power order. *Israel Journal of Mathematics*, 230(2):919–947, 2019. doi:10.1007/s11856-019-1846-x.

2 Erhard Aichinger and Peter Mayr. Polynomial clones on groups of order pq. *Acta Mathematica Hungarica*, 114(3):267–285, 2006. doi:10.1007/s10474-006-0530-x.

3 Erhard Aichinger and Nebojša Mudrinski. Some applications of higher commutators in Mal'cev algebras. *Algebra universalis*, 63(4):367–403, 2010. doi:10.1007/s00012-010-0084-1.

4 Jorge Almeida, Mikhail V. Volkov, and Svetlana V. Goldberg. Complexity of the identity checking problem for finite semigroups. *Journal of Mathematical Sciences*, 158(5):605–614, 2009. doi:10.1007/s10958-009-9397-z.

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

6 Clifford Bergman. *Universal Algebra: Fundamentals and selected topics*. CRC Press, 2011.

7 Stanley Burris and John Lawrence. The equivalence problem for finite rings. *Journal of Symbolic Computation*, 15(1):67–71, 1993. doi:10.1006/jsco.1993.1004.

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

9 Stefano Fioravanti. Closed sets of finitary functions between finite fields of coprime order. *Algebra universalis*, 81:52, 2020. published online. doi:10.1007/s00012-020-00683-5.

10 Attila Földvári and Gábor Horváth. The complexity of the equation solvability and equivalence problems over finite groups. *International Journal of Algebra and Computation*, 30(03):607–623, 2020. doi:10.1142/S0218196720500137.

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

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

13 Vince Grolmusz and Gábor Tardos. Lower bounds for (MOD_p-MOD_m) circuits. *SIAM Journal on Computing*, 29(4):1209–1222, 2000. doi:10.1137/S0097539798340850.

14 Christian Herrmann. Affine algebras in congruence modular varieties. *Acta Universitatis Szegediensis*, 41:119–125, 1979.

15 Gábor Horváth. The complexity of the equivalence problem over finite rings. *Glasgow Mathematical Journal*, 54(1):193–199, 2012. doi:10.1017/S001708951100053X.

16 Gábor Horváth and Csaba Szabó. The complexity of checking identities over finite groups. *International Journal of Algebra and Computation*, 16(5):931–939, 2006. doi:10.1142/S0218196706003256.

17 Gábor Horváth and Csaba Szabó. The extended equivalence and equation solvability problems for groups. *Discrete Mathematics & Theoretical Computer Science*, 13(4):23–32, 2011. doi:10.46298/dmtcs.536.

18 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. doi:10.1016/j.jpaa.2012.02.007.

19 Harry B. Hunt III and Richard Edwin Stearns. The complexity of equivalence for commutative rings. *Journal of Symbolic Computation*, 10(5):411–436, 1990. doi:10.1016/S0747-7171(08)80053-3.

20 Paweł Idziak, Piotr Kawałek, Jacek Krzaczkowski, and Armin Weiß. Equation satisfiability in solvable groups. *Theory of Computing Systems*, 2022. doi:10.1007/s00224-022-10082-z.

21 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Expressive power, satisfiability and equivalence of circuits over nilpotent algebras. In *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:15. EATCS, 2018. doi:10.4230/LIPIcs.MFCS.2018.17.

22 Pawel M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Intermediate problems in modular circuits satisfiability. In *Proceedings of the 35th Annual Symposium on Logic in Computer Science (LICS)*, pages 578–590, 2020. doi:10.1145/3373718.3394780.

23 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Satisfiability of Circuits and Equations over Finite Malcev Algebras. In *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*, volume 219 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2022.37.

24 Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. Nonuniform deterministic finite automata over finite algebraic structures, 2024. Manuscript.

25 Paweł M Idziak and Jacek Krzaczkowski. Satisfiability in multivalued circuits. *SIAM Journal on Computing*, 51(3):337–378, 2022. doi:10.1137/18M122019.

26 Piotr Kawałek and Armin Weiß. Violating constant degree hypothesis requires breaking symmetry, 2023. arXiv:2311.17440.

27 Ondřej Klíma. Complexity issues of checking identities in finite monoids. *Semigroup Forum*, 79(3):435–444, 2009. doi:10.1007/s00233-009-9180-y.

28 Michael Kompatscher. CSAT and CEQV for nilpotent Maltsev algebras of Fitting length > 2, 2021. arXiv:2105.00689.

29 Michael Kompatscher. CC-circuits and the expressive power of nilpotent algebras. *Logical Methods in Computer Science*, 18(2), 2022. doi:10.46298/lmcs-18(2:12)2022.

30 Andrew Moorhead. Higher commutator theory for congruence modular varieties. *Journal of Algebra*, 513:133–158, 2018. doi:10.1016/j.jalgebra.2018.07.026.

31 Bernhard Schwarz. The complexity of satisfiability problems over finite lattices. In *Proceedings of the 21st Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 31–43, 2004. `doi:10.1007/978-3-540-24749-4_4`.

32 Joel VanderWerf. Wreath products of algebras: generalizing the Krohn-Rhodes theorem to arbitrary algebras. *Semigroup Forum*, 52(1):93–100, 1996. `doi:10.1007/BF02574084`.

33 Armin Weiß. Hardness of Equations over Finite Solvable Groups Under the Exponential Time Hypothesis. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 102:1–102:19, 2020. `doi:10.4230/LIPIcs.ICALP.2020.102`.