Satisfiability in multi-valued circuits

Paweł M. Idziak
Department of Theoretical Computer Science
Faculty of Mathematics and Computer Science
Jagiellonian University
Kraków, Poland
idziak@tcs.uj.edu.pl

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

Abstract

Satisfiability of Boolean circuits is among the most known and important problems in theoretical computer science. This problem is NP-complete in general but becomes polynomial time when restricted 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 strictly connected with the problems of solving equations (or systems of equations) over finite algebras.

The research reported in this work was motivated by a desire to know for which finite algebras $\mathbf{A}$ there is a polynomial time algorithm that decides if an equation over $\mathbf{A}$ has a solution. We are also looking for polynomial time algorithms that decide if two circuits over a finite algebra compute the same function. Although we have not managed to solve these problems in the most general setting we have obtained such a characterization for a very broad class of algebras from congruence modular varieties. This class includes 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).

This paper seems to be the first systematic study of the computational complexity of satisfiability of non-Boolean circuits and solving equations over finite algebras. Our characterization is given in terms of nice structural properties of algebras for which the problems are solvable in polynomial time. Such algebras have to decompose into two factors: a nilpotent one and a factor that essentially behaves as a finite distributive lattice.

CCS Concepts • **Theory of computation → Complexity theory and logic**; **Problems, reductions and completeness**; *Circuit complexity*; *Constraint and logic programming*; • **Mathematics of computing** → *Combinatorial algorithms*;

Keywords circuit satisfiability, solving equations

ACM Reference Format:
Paweł M. Idziak and Jacek Krzaczkowski. 2018. Satisfiability in multi-valued circuits. In *LICS '18: LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, July 9–12, 2018, Oxford, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3209108.3209173

1 Introduction

One of the most celebrated NP-complete problem is SAT – the problem that takes on a Boolean expression and decides whether there is a $\{0, 1\}$-valuation of variables that satisfies this expression.

The most popular variant of this problem is CNF-SAT (often called SAT as well) in which the input is in Conjunctive Normal Form. A formula in CNF is a conjunction of clauses each of which is a disjunction of (e.g. at most 3) literals. These clauses (if ternary) can be treated as (ternary) relations on the set $\{0, 1\}$ and the SAT problem simply asks whether a conjunction of atomic formulas (in this new relational language) is satisfiable. This generalizes to any (finite) relational structure, say $\mathbb{D}$, where the problem lies in answering whether a conjunction of atomic formulas (in the language of $\mathbb{D}$) is satisfiable in $\mathbb{D}$. This is now known under the name of Constraint Satisfaction Problem, or CSP for short. A characterization of relational structures over $\{0, 1\}$ for which CSP is solvable in a polynomial time has been done in [33]. The structures for which a polynomial time algorithm is not provided in [33] have been shown there to be NP-complete with respect to CSP. The similar dichotomy conjecture for CSP over arbitrary finite domains has been stated by Feder and Vardi in [9]. With the help of deep algebraic tools two algorithmic paradigms have been shown to be fruitful in establishing polynomial time complexity of a wide range of relational structures. One of these paradigms generalizes Gaussian elimination method to the realm of algebras with few subpowers [24]. The other generalizes DATALOG programming to local consistency checking method [2]. Both of those methods were explored to their limits, so that a lot of effort has been put to find a new or hybrid approach. Very recently two independent proofs (one by D. Zhuk [37] and another one by A. Bulatov [5]) confirming the CSP dichotomy conjecture have been announced.

In contrast to CNF-SAT the problem o satisfiability of general Boolean expression is often called CIRCUITS SAT or CSAT for short. After restricting this NP-complete problem for example to the circuits that are either monotone (only AND and OR gates) or linear (only XOR gates) the problem becomes solvable in a polynomial time. Thus it is natural to isolate those collections of 2-valued gates that lead to circuits with polynomially solvable satisfiability problem. Actually such characterization of tractable families of 2-valued gates can be inferred from the results of [13].

In general, different collections of admissible gates (on a given set) give rise to algebras (in the universal algebraic sense). Thus we will talk about circuits over a fixed finite algebra $\mathbf{A}$, i.e. an algebra with finitely many elements and finitely many fundamental operations. In this language the output gates of such circuits can be represented by terms of an algebra $\mathbf{A}$ (or polynomials of $\mathbf{A}$, if

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

LICS '18, July 9–12, 2018, Oxford, United Kingdom

ACM ISBN 978-1-4503-5583-4/18/07…$15.00
https://doi.org/10.1145/3209108.3209173

values on some input gates are fixed). We also relax the notion of satisfiability of such circuits to be read:

> CSAT($\mathbf{A}$)
> given a circuit over $\mathbf{A}$ 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 $\mathbf{g}_1, \mathbf{g}_2$, i.e. $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x})$.

Note here, that in some cases (including 2-element Boolean algebra) the satisfiability of $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x})$ can be replaced by satisfiability of $\mathbf{g}(\overline{x}) = c$, where c is a constant and $\mathbf{g}$ is a new output gate that combines $\mathbf{g}_1$ and $\mathbf{g}_2$.

In a circuit that has more than two output gates it is also natural to state the following question. We will see that this very similar question has different taste.

> MCSAT($\mathbf{A}$)
> given a circuit over $\mathbf{A}$ with output gates $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$ is there a valuation of input gates $\overline{x}$ that gives the same output on all the $\mathbf{g}_i$'s, i.e. $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x}) = \ldots = \mathbf{g}_k(\overline{x})$.

From algebraic point of view problem CSAT($\mathbf{A}$) asks for the solutions of an equation over $\mathbf{A}$. The problem MCSAT($\mathbf{A}$) asks for solutions of a special system of equations over $\mathbf{A}$. But we can also ask for solutions of arbitrary systems of equations. This however has a more natural wording in purely algebraic terms.

> SCSAT($\mathbf{A}$)
> Given polynomials
> $$\mathbf{g}_1(\overline{x}), \mathbf{h}_1(\overline{x}), \ \ldots \ , \mathbf{g}_k(\overline{x}), \mathbf{h}_k(\overline{x})$$
> of an algebra $\mathbf{A}$, is there a valuation of the variables $x_1, \ldots, x_n$ in A such that
> $$\begin{array}{rcl} \mathbf{g}_1(x_1, \ldots, x_n) & = & \mathbf{h}_1(x_1, \ldots, x_n) \\ & \vdots & \\ \mathbf{g}_k(x_1, \ldots, x_n) & = & \mathbf{h}_k(x_1, \ldots, x_n), \end{array}$$

With this natural approach via multi valued circuits also the problem TAUTOLOGY has its natural generalization:

> CEQV($\mathbf{A}$)
> given a circuit over $\mathbf{A}$ 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. $\mathbf{g}_1(\overline{x}) = \mathbf{g}_2(\overline{x})$.

In the algebraic setting this is simply the question of equivalence of two terms or polynomials. Here equivalence of k pairs of terms/polynomials reduces to k independent CEQV queries.

In Boolean realm the problem CEQV can be treated as the complement of CSAT and therefore is co-NP-complete. In general the closely related problem CEQV($\mathbf{A}$) is somehow independent from CSAT($\mathbf{A}$). This independence means that all four possibilities of tractability/intractability can be witnessed by some finite algebras. For example for the 2-element lattice $\mathbf{L}$ the problem CSAT($\mathbf{L}$) is in P while CEQV($\mathbf{L}$) is co-NP-complete. An example of a finite semigroup $\mathbf{S}$ with CEQV($\mathbf{S}$) $\in$ P and CSAT($\mathbf{S}$) being NP-complete can be inferred from [27].

It is worth to note that solving equations (or systems of equations) 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 the decision version of these problems one asks if an equation (or system of such equations) expressed in the language of a fixed algebra $\mathbf{A}$, has a solution in $\mathbf{A}$. 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 [31]. In finite realms such problems are obviously decidable in nondeterministic polynomial time. There are numerous results related to problems connected with solving equations and systems of equations over fixed finite algebras. Most of them concerns well known algebraic structures as groups [7], [11], [19], [21] rings [17], [6] or lattices [34] but there are also some more general results [1], [30].

The main goal of this paper is to attack the classification problems of the form: for which finite algebras $\mathbf{A}$ there is an algorithm that answers one of the problems CSAT($\mathbf{A}$), MCSAT($\mathbf{A}$), SCSAT($\mathbf{A}$) or CEQV($\mathbf{A}$) in polynomial time with respect to the size of the circuit, i.e. the size of the underlying graph of the circuit. It seems that the most natural way to look at these problems is to treat circuits over $\mathbf{A}$ (or in fact output gates of such circuits) as terms/polynomials of the algebra $\mathbf{A}$. This obvious translation makes our attack fruitful, as we can apply deep results and techniques developed by universal algebra such as *modular commutator theory* and *tame congruence theory*. These tools are especially useful in case of algebras generating congruence modular variety. This assumption covers many well known structures as groups, rings, modules or lattices. Our attempt to attack the classification problems has resulted in partial characterization of computational complexity of CSAT, MCSAT and CEQV for algebras generating congruence modular varieties. This partial characterization leaves some room to be filled before establishing a dichotomy. We will also briefly discuss the difficulties arising in filling the gap in our characterization.

2 The results

In this section we present the state of the art in more details and discuss our results and tools.

The first thing in which our research differs from what has been already considered is that we concentrate on circuits rather than on syntactic form of terms or polynomials. This difference is visible in how the size of the input is measured. We have seen how an output gate can be treated as a term or a polynomial. On the other hand, in an obvious way, every term over $\mathbf{A}$ can be treated as a circuit in which each gate is used as an input to at most one other gate. This leads to a circuit whose underlying graph is a tree. However circuits can have more compact representation than terms. For example, in groups the terms $\mathbf{t}_n(x_1, x_2, \ldots, x_n) = [\ldots[[x_1, x_2], x_3] \ldots x_n]$, (where $[x, y] = x^{-1}y^{-1}xy$ is the group commutator) expressed in the pure group language of $(\cdot, {}^{-1})$ have an exponential size in n, as the number of occurrences of variables doubles whenever we pass from n to $n+1$. On the other hand the size of a circuit realizing $\mathbf{t}_n$ has $6n - 5$ vertices as can be seen from Figure 1.

The consequences of this (exponential) disproportion in measuring the input size for terms and circuits are illustrated by the following example.

Example 2.1. *There are finite groups* $\mathbf{A}$ *such that* CSAT($\mathbf{A}$) *is NP-complete, while there are polynomial time algorithms for solving equations over* $\mathbf{A}$.

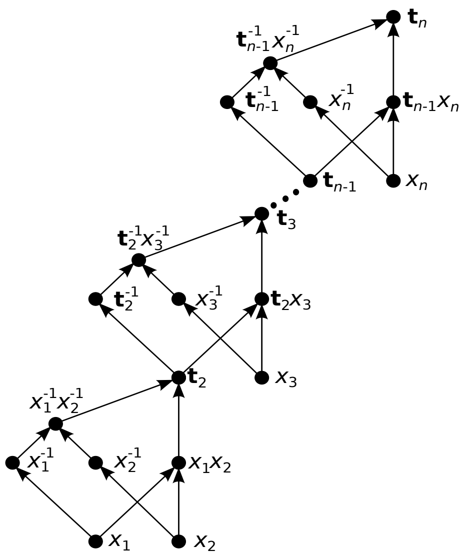

Figure 1

There are also finite groups $\mathbf{B}$ *such that* Ceqv($\mathbf{A}$) *is co-NP-complete, while there are polynomial time algorithms for checking the identities in* $\mathbf{B}$.

Proof. The first such example for the equation versus circuit satisfiability problem was the symmetric group $\mathbf{S}_3$ for which polynomial time algorithm was shown in [20], while the first author's observation on the NP-completeness is included in [12].

The papers [18, 21, 22] contain many other examples of solvable non-nilpotent groups which witness both statements in our example. □

Note that in case of SCsat there is no such disproportion in the size as every polynomial equation $\mathbf{s}(\overline{x}) = \mathbf{t}(\overline{x})$ can be replaced by a system of equations of the form $y = \mathbf{f}(x_1, \ldots, x_k)$ or $y = c$, where $\mathbf{f}$ is one of the basic operations and c is a constant. This replacement has linear size with respect to the circuit representing $\mathbf{t}(\overline{x})$. For example for the above term

$$\mathbf{t}_n(x_1, x_2, \ldots, x_n) = [\ldots[[x_1, x_2], x_3] \ldots x_n],$$

slightly abusing our conditions, we can use the following representation

$$\begin{array}{rcl} t_2 & = & x_1^{-1}x_2^{-1}x_1x_2 \\ t_3 & = & t_2^{-1}x_3^{-1}t_2x_3 \\ & \vdots & \\ t_n & = & t_{n-1}^{-1}x_n^{-1}t_{n-1}x_n, \end{array}$$

in which $t_2, \ldots, t_n$ are treated as variables.

However, even in the setting of a single equation, representing a polynomial $\mathbf{t}(\overline{x})$ by its corresponding circuit and looking at the size of this circuit (instead of the syntactic length of $\mathbf{t}$) allows us to harmlessly expand the original language of the algebra $\mathbf{A}$ by finitely many polynomials. In fact in our intractability proofs we will often expand the language of the original algebra $\mathbf{A}$ by finitely many polynomials of $\mathbf{A}$. This will allow us to code NP-complete problems in much more smooth way. Note that the possibility of such expansions shows that the characterizations we are looking for can be done up to polynomial equivalence of algebras; two algebras $\mathbf{A}_1$ and $\mathbf{A}_2$ are said to be polynomially equivalent if they have the same universes and each polynomial of one of them can be defined by composing the polynomials of the other one, i.e. $\mathrm{Pol}\,\mathbf{A}_1 = \mathrm{Pol}\,\mathbf{A}_2$, where $\mathrm{Pol}\,\mathbf{A}$ is the set of all polynomials of the algebra $\mathbf{A}$.

Fact 2.2. *Let* $\mathbf{A}_1$, $\mathbf{A}_2$ *be finite algebras such that* $\mathrm{Pol}\,\mathbf{A}_1 = \mathrm{Pol}\,\mathbf{A}_2$. *Then* Csat($\mathbf{A}_1$) *and* Csat($\mathbf{A}_2$) *are polynomial-time equivalent.*

Proof. We will show polynomial time reduction from Csat($\mathbf{A}_1$) to Csat($\mathbf{A}_2$). Since $\mathrm{Pol}\,\mathbf{A}_1 \subseteq \mathrm{Pol}\,\mathbf{A}_2$ we can express every fundamental operation (gate) of $\mathbf{A}_1$ using circuit over $\mathbf{A}_2$. Let T be a function which for every fundamental operation f of $\mathbf{A}_1$ returns a circuit $T(f)$ over $\mathbf{A}_2$ such that $T(f)$ computes function f. Now it easy to see that transformation which for a given circuit C over $\mathbf{A}_2$ returns circuit C' obtained by replacing every occurrence of the gate f in C by $T(f)$, is a polynomial time reduction from Csat($\mathbf{A}_1$) to Csat($\mathbf{A}_2$). □

In our proofs of NP-completeness we will often use the following corollary of Fact 2.2.

Corollary 2.3. *Let* $\mathbf{A} = (A, F)$ *be a finite algebra and let* $G \subseteq \mathrm{Pol}\,\mathbf{A}$ *be a finite set of its polynomials.*
If Csat($\mathbf{A}, F \cup G$) *is NP-complete, then* Csat($\mathbf{A}, F$) *is NP-complete.*

It turns out that quite a few results on the complexity of the problems Csat, MCsat, SCsat and Ceqv are already known for particular kinds of (finite) algebras.

Example 2.4. *Finite Groups:*

- *If* $\mathbf{A}$ *is abelian then* SCsat($\mathbf{A}$) $\in$ *P (by Gaussian elimination), and for all other groups* SCsat($\mathbf{A}$) *is NP-complete [11].*
- Csat($\mathbf{A}$) *is in P, whenever* $\mathbf{A}$ *is nilpotent [11] and NP-complete otherwise [11, 21].*
- Ceqv($\mathbf{A}$) *is in P, if* $\mathbf{A}$ *is nilpotent [7] and co-NP-complete otherwise [19, 21].*

Example 2.5. *Finite Rings:*

- *If* $\mathbf{A}$ *is essentially an abelian group (i.e. multiplication satisfies the identity* $xy = 0$*) then* SCsat($\mathbf{A}$) $\in$ *P (by Gaussian elimination), and for all other rings* SCsat($\mathbf{A}$) *is NP-complete [30].*
- Csat($\mathbf{A}$) *is in P, whenever* $\mathbf{A}$ *is nilpotent [17] and NP-complete otherwise [6].*
- Ceqv($\mathbf{A}$) *is in P, whenever* $\mathbf{A}$ *is nilpotent and NP-complete otherwise (see [23] for commutative rings and [6] for general case).*

Example 2.6. *Finite Lattices:*

- Csat($\mathbf{A}$) $\in$ *P if* $\mathbf{A}$ *is distributive and NP-complete otherwise [34].*
- *For all nontrivial lattices* $\mathbf{A}$, SCsat($\mathbf{A}$) *is NP-complete while* Ceqv($\mathbf{A}$) *is co-NP-complete (easy to see).*

The examples given above suggest that the existence of polynomial time algorithms for the considered circuits problems go hand in hand with nice structure theory of the underlying algebras. However there are only two results that can be considered general enough to be expressed in structural terms. These results are stated in the following two theorems.

First note that E. Aichinger and N. Mudrinski [1] have shown the following theorem, a partial converse of which is our Theorem 2.15.

Theorem 2.7. *If* $\mathbf{A}$ *is a finite supernilpotent algebra from a congruence variety then* $\textsc{Ceqv}(\mathbf{A})$ *is in P.*

The second general result is that of B. Larose and L. Zádori [30]. After observing that SCsat has exactly the same expressive power as CSP they used mutual translation between SCsat and CSP to prove the first part of the next characterization, while the second one is a form of Gaussian elimination.

Theorem 2.8. *For a finite algebra* $\mathbf{A}$ *from a congruence modular variety:*

- *if* $\mathrm{SC}\textsc{sat}(\mathbf{A})$ *is not NP-complete then* $\mathbf{A}$ *is affine (i.e.* $\mathbf{A}$ *is polynomially equivalent to a module over a finite ring),*
- *if* $\mathbf{A}$ *is affine then* $\mathrm{SC}\textsc{sat}(\mathbf{A}) \in P$*.*

Not as much is known when one leaves the congruence modularity realm. It is worth to note however that an important extension of Theorem 2.8 to finite algebras from varieties omitting type **1** (in the sense of Tame Congruence Theory, see [16]) can be found in [36].

Also a number of results on semigroups do not fall in congruence modular setting but these results are still about particular type of algebras. The paper [28] gives a nice, but somewhat technical, characterization of finite monoids $\mathbf{A}$ for which $\mathrm{SC}\textsc{sat}(\mathbf{A}) \in \mathsf{P}$. There are also several results on the complexity of $\mathrm{SC}\textsc{sat}(\mathbf{A})$ for particular semigroups or classes of semigroups, but we are far from having a full characterization similar to that for monoids. In fact the paper [28] contains a proof that the expressive power of SCsat over semigroups is equivalent to the expressive power of CSP. However full understanding of semigroups with polynomially solvable SCsat requires a translation of the dichotomy borderline into structural condition for semigroups. Surprisingly another class of algebras for which SCsat coincides with expressive power of CSP is the class of algebras with unary operations only [3, 8].

When coming to a single equation we are still able to prove that the expressive power of Csat is no weaker that this of CSP, as expressed below.

Proposition 2.9. *For every finite relational structure* $\mathbb{D}$ *(with finitely many relations) there is a finite algebra* $\mathbf{A}[\mathbb{D}]$ *such that the problem* $\mathrm{CSP}(\mathbb{D})$ *is polynomially equivalent to* $\textsc{Csat}(\mathbf{A}[\mathbb{D}])$*.*

Unlike in the SCsat setting we do not know whether the expressive power of Csat is not bigger than the one of CSP.

Problem 1. *Is it true that for every finite algebra* $\mathbf{A}$ *there exists a relational structure* $\mathbb{D}[\mathbf{A}]$ *such that the problems* $\textsc{Csat}(\mathbf{A})$ *and* $\mathrm{CSP}(\mathbb{D}[\mathbf{A}])$ *are polynomially equivalent?*

The above difference between a single equation and a system of equations is probably a consequence of the presence of an external conjunction in systems of equations. Intuitively, to replace a system of equations by a single equation, one needs to squeeze many terms (or polynomials) into a single one. This requires an analogue of an internal conjunction (that can be expressed by a polynomial) present in Boolean algebras. Since such a squeeze is not always possible, more algebras may have polynomial time algorithms for Csat than for SCsat. Actually our work confirms this claim.

One of the main difficulties in characterizing finite algebras with $\mathrm{SC}\textsc{sat}(\mathbf{A}) \in \mathsf{P}$ is that this property does not carry over quotient algebras (unless $\mathsf{P} = \mathsf{NP}$). The paper [28] contains an example of a finite semigroup $\mathbf{A}$ and its congruence θ with $\mathrm{SC}\textsc{sat}(\mathbf{A}/\theta)$ being NP-complete while $\mathrm{SC}\textsc{sat}(\mathbf{A}) \in \mathsf{P}$. The example below shows that this unwanted phenomena occurs for the Csat problem, as well.

Example 2.10. *There is a finite algebra* $\mathbf{A}$ *and its congruence* θ *such that* $\textsc{Csat}(\mathbf{A}) \in P$ *while* $\textsc{Csat}(\mathbf{A}/\theta)$ *is NP-complete.*

Since passing to quotient algebras may not preserve polynomial time complexity for Csat, it is natural to work under the stronger assumption that not only $\textsc{Csat}(\mathbf{A}) \in \mathsf{P}$, but $\textsc{Csat}(\mathbf{A}/\theta) \in \mathsf{P}$ for all congruences θ of $\mathbf{A}$. Such assumption has also a natural interpretation. Given $\mathbf{A}$ we want a fast method to solve equations over $\mathbf{A}$, or at least decide if such equations have solutions. However such solutions may not exist in the original algebra $\mathbf{A}$. They obviously do exist in $\mathbf{A}/1_{\mathbf{A}}$, where $1_{\mathbf{A}}$ is the congruence collapsing everything. Thus the best we can do, is to determine (existence of) the solutions with best possible precision, i.e. modulo the smallest congruences possible. This however requires $\mathbf{A}$ to be regular enough so that $\textsc{Csat}(\mathbf{A}')$ is in P for all quotients $\mathbf{A}'$ of $\mathbf{A}$.

After fixing the setting we are working in, we can state our main result in the next theorem. This result shows that the structure of algebra $\mathbf{A}$ with tractable $\textsc{Csat}(\mathbf{A})$ has to be nice. Such an algebra has to decompose nicely into two factors: a nilpotent one and a factor that resembles a finite distributive lattice (DL-like for short). To express this we need a notion of a subdirect product. We say that an algebra $\mathbf{A}$ is a subdirect product of the family of algebras $(\mathbf{A}_i)_{i\in I}$ if $\mathbf{A}$ is a subalgebra of the direct product $\prod_{i\in I}\mathbf{A}_i$ and $\mathbf{A}$ projects surjectively onto each of the stalks $\mathbf{A}_i$. The concept of subdirect product allows to define subdirectly irreducible algebra as an algebra that in every representation by a subdirect product has be isomorphic to one of the stalks. One can easily show that an algebra $\mathbf{A}$ is subdirectly irreducible if $\mathbf{A}$ has the smallest non-zero congruence, called the monolith of $\mathbf{A}$.

Now we are ready to state our characterization theorem.

Theorem 2.11. *Let* $\mathbf{A}$ *be a finite algebra from a congruence modular variety.*

1. *If* $\mathbf{A}$ *has no quotient* $\mathbf{A}'$ *with* $\textsc{Csat}(\mathbf{A}')$ *being NP-complete then* $\mathbf{A}$ *is isomorphic to a direct product* $\mathbf{N}\times\mathbf{D}$*, where* $\mathbf{N}$ *is a nilpotent algebra and* $\mathbf{D}$ *is a subdirect product of* 2*-element algebras each of which is polynomially equivalent to the* 2*-element lattice.*
2. *If* $\mathbf{A}$ *decomposes into a direct product* $\mathbf{N}\times\mathbf{D}$*, where* $\mathbf{N}$ *is a supernilpotent algebra and* $\mathbf{D}$ *is a subdirect product of* 2*-element algebras each of which is polynomially equivalent to the* 2*-element lattice, then for every quotient* $\mathbf{A}'$ *of* $\mathbf{A}$ *the problem* $\textsc{Csat}(\mathbf{A}')$ *is solvable in polynomial time.*

To understand the above result first note that the congruence modularity assumption covers most algebraic structures considered in classical mathematics. In particular it includes groups (and their extensions like rings, fields), and lattices (and their extensions like Boolean algebras or other algebras connected with multi-valued logics). This assumption does not cover however semigroups (or even semilattices) or multiunary algebras.

The conditions (1) and (2) show that the nilpotent groups and rings as well as distributive lattices mentioned in Examples 2.4, 2.5 and 2.6 are in fact paradigms for CSAT tractability in congruence modular realm. In fact the structural conditions described in Theorem 2.11, when specialized to groups, rings or lattices, gives the already known characterizations presented in Examples 2.4, 2.5 and 2.6.

The decomposition enforced in (1) is a result of almost a dozen of constructions interpreting NP-complete problems (mostly SAT and k-COLORABILITY) into CSAT($\mathbf{A}$), whenever $\mathbf{A}$, or some of its quotients, fails to satisfy one of the structural conditions that finally lead to this nice decomposition.

The second factor, $\mathbf{D}$, of this decomposition is easier to understand than the first one. It essentially behaves like a finite distributive lattice, but the algebra $\mathbf{D}$ does not need to actually have (explicit) lattice operations. Instead $\mathbf{D}$ is composed of 2-element algebras each of which does have lattice operations expressible by polynomials, while all of their operations are monotone with respect to this lattice order.

The next example shows that a subdirect product of 2-element algebras each of which is polynomially equivalent to the 2-element lattice need not be polynomially equivalent to a distributive lattice.

Example 2.12. *Let* $\mathbf{A} = (A, \mathbf{m})$ *be a subreduct of* $(\{0,1\}, \wedge, \vee)^3$, *with*

- $A = \{(1,1,1), (0,1,1), (1,0,1), (1,1,0)\}$
- *and* $\mathbf{m}$ *being the majority operation* $\mathbf{m}(x,y,z) = (x \vee y) \wedge (y \vee z) \wedge (z \vee x)$.

Then

- $\mathbf{A}$ *belongs to congruence distributive (and therefore to congruence modular) variety*
- $\mathbf{A}$ *is a subdirect product of algebras polynomially equivalent to two element lattices,*
- $\mathbf{A}$ *is not polynomially equivalent to a distributive lattice.*

Proof. The first two items are obvious. To see the third one note that, up to isomorphism, there are only two four element lattices:

- the four element chain,
- the four element Boolean lattice.

On the other hand, for three pairwise different elements $a, b, c \in A$ we have $\mathbf{m}(a,b,c) = \overline{1}$, where $\overline{1} = (1,1,1)$. Sending isomorphically, say by h, all possible 3-element tuples from A into one of the above 4-element lattices we simply cannot find a room for $h(\overline{1})$ under the assumption that $\mathbf{m}$ preserves lattice order. □

The first factor, $\mathbf{N}$, of the decomposition described in Theorem 2.11 requires the general algebraic notion of nilpotency in congruence modular setting that goes back to the late 1970's when Smith [35], Hagemann and Herrmann [15], Gumm [14] and finally Freese and McKenzie [10] developed necessary deep tools of *modular commutator theory*. In fact a notion of the commutator multiplication $[\alpha, \beta]$ of congruences α, β of arbitrary algebras was defined in a way that extends multiplication of ideals in ring theory and commutator multiplication of normal subgroups in group theory. With the help of such commutator one can define abelian, solvable and nilpotent congruences and algebras.

If α, β, γ are congruences of an algebra then we say that α *centralizes* β *modulo* γ, denoted $C(\alpha, \beta; \gamma)$, if for every $n \geqslant 1$, every $(n+1)$-ary term $\mathbf{t}$, every $(a,b) \in \alpha$, and every $(c_1, d_1), \ldots, (c_n, d_n) \in \beta$ we have

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

Obviously among all congruences γ such that $C(\alpha, \beta; \gamma)$ there is the smallest one and it is denoted by $[\alpha, \beta]$ and called the commutator of α and β.

By means of the commutator it is possible to define notions of abelian, solvable and nilpotence for arbitrary algebras. First, for a congruence θ and $i = 1, 2, \ldots$ we put

$$\begin{array}{rclcrcl} \theta^{(1)} & = & \theta & \qquad & \theta^{[1]} & = & \theta \\ \theta^{(i+1)} & = & [\theta, \theta^{(i)}] & & \theta^{[i+1]} & = & [\theta^{[i]}, \theta^{[i]}]. \end{array}$$

Now, a congruence θ of $\mathbf{A}$ is called *k-step nilpotent* [or *k-step solvable*] if $\theta^{(k+1)} = 0_{\mathbf{A}}$ [$\theta^{[k+1]} = 0_{\mathbf{A}}$] and the algebra $\mathbf{A}$ is *nilpotent* [*solvable*] if 1_A is k-step nilpotent [k-step solvable] for some finite k. In particular θ [or $\mathbf{A}$] is abelian if $\theta^{(2)} = \theta^{[2]} = 0_{\mathbf{A}}$ [or $1_{\mathbf{A}}^{(2)} = 0_{\mathbf{A}}$].

Fuller discussions of the generalized commutator may be found in [10], [32, Section 4.13] and [16, Chapter 3].

Finite nilpotent groups (and rings) behave very nicely. In particular they decompose into direct products of groups (or rings) of prime power order. Unfortunately such nice decomposition of nilpotent algebras in congruence modular varieties does not hold in general. However, in this general setting, nilpotent algebras that have this nice decomposition (and have only finitely many basic operations) are exactly those that are supernilpotent. In fact supernilpotency has been introduced by another universal algebraic generalization of commutator multiplication of congruences.

For a bunch of congruences $\alpha_1, \ldots, \alpha_k, \beta, \gamma \in \mathrm{Con}\,\mathbf{A}$ we say that $\alpha_1, \ldots, \alpha_k$ centralize β modulo γ, and write $C(\alpha_1, \ldots, \alpha_k, \beta; \gamma)$, if for all polynomials $\mathbf{f} \in \mathrm{Pol}\,\mathbf{A}$ 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

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

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

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

This notion was introduced by A. Bulatov [4] and further developed by E. Aichinger and N. Mudrinski [1]. In particular they have shown that for all $\alpha_1, \ldots, \alpha_k \in \mathrm{Con}\,\mathbf{A}$ there is the smallest congruence γ with $C(\alpha_1, \ldots, \alpha_k; \gamma)$ called the k-ary commutator and denoted by $[\alpha_1, \ldots, \alpha_k]$. Such generalized commutator behaves especially well in algebras from congruence modular varieties. In particular this commutator is monotone, join-distributive and we have

$$[\alpha_1, [\alpha_2, \ldots, \alpha_k]] \leqslant [\alpha_1, \ldots, \alpha_k]$$

Thus every k-supernilpotent algebra, i.e. algebra satisfying

$$[\overbrace{1, \ldots, 1}^{k+1 \text{ times}}] = 0,$$

is k-nilpotent.

To illustrate the precise difference between nilpotency and superniloptency for an algebra $\mathbf{A}$ with finitely many elements, from congruence modular variety, note that due to [10] and [26] the following two conditions are equivalent

- $\mathbf{A}$ is k-supernilpotent,

- $\mathbf{A}$ is k-nilpotent, decomposes into a direct product of algebras of prime power order and the clone $\mathrm{Clo}\,\mathbf{A}$ is generated by finitely many operations.

The following example shows two ways in which nilpotent algebras with finitely many elements may fail to be supernilpotent.

Example 2.13. *For positive integers m, n and prime numbers p, q put*

- *f_m to be unary operation modulo m,*
- *$+_m$ to be addition modulo m,*
- *$p^n(x_1, \dots, x_n) = p \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_n$ modulo p^2.*

Then, the algebras

- *$\mathbf{Z}_m = (Z_m, +_m)$ are abelian,*
- *$\mathbf{A}^{(k)}_{p^2} = (Z_{p^2}, +_{p^2}, p^2, \dots, p^k)$, for finite k, are supernilpotent,*
- *$\mathbf{A}^{(\infty)}_{p^2} = (Z_{p^2}, +_{p^2}, \{p^i\}_{i=2}^{\infty})$ are nilpotent but not supernilpotent,*
- *$\mathbf{B}_{pq} = (Z_{pq}, +_{pq}, f_p)$ are nilpotent but not supernilpotent.*

The nilpotent/supernilpotent gap that occurs in Theorem 2.11 resists to be easily filled. This is because in supernilpotent case there is a bound on the arity of the so called commutator polynomials. These commutator polynomials can imitate the behavior of the long conjunction. In nilpotent (but not supernilpotent) case arbitrary long conjunctions are expressible. But this can be probably done at the expense of exponentially large (with respect to the arity) circuits needed to represent those conjunctions. This expected exponential size probably prevents polynomial time reduction of NP-complete problems to CSAT in nilpotent but not supernilpotent case.

Using Theorems 2.11 and 2.8 we are able to infer the following corollary.

Corollary 2.14. *Let $\mathbf{A}$ be a finite algebra from a congruence modular variety.*

1. *If $\mathbf{A}$ has no quotient $\mathbf{A}'$ with MCSAT($\mathbf{A}'$) being NP-complete then $\mathbf{A}$ is isomorphic to a direct product $\mathbf{M} \times \mathbf{D}$, where $\mathbf{M}$ is an affine algebra and $\mathbf{D}$ is a subdirect product of 2-element algebras each of which is polynomially equivalent to the 2-element lattice.*
2. *If $\mathbf{A}$ decomposes into a direct product $\mathbf{M} \times \mathbf{D}$, where $\mathbf{M}$ is an affine algebra and $\mathbf{D}$ is a subdirect product of 2-element algebras each of which is polynomially equivalent to the 2-element lattice, then for every quotient $\mathbf{A}'$ of $\mathbf{A}$ the problem MCSAT($\mathbf{A}'$) is solvable in polynomial time.*

Our constructions used to show that lack of nice structure of the algebra $\mathbf{A}$ leads to intractability of CSAT($\mathbf{A}$) can be also modified to work for intractability of CEQV($\mathbf{A}$) so that we are able to prove a partial converse to Theorem 2.7.

Theorem 2.15. *Let $\mathbf{A}$ be a finite algebra from a congruence modular variety. If $\mathbf{A}$ has no quotient $\mathbf{A}'$ with CEQV($\mathbf{A}'$) being co-NP-complete then $\mathbf{A}$ is nilpotent.*

3 The methods

The reductions we have produced to show intractability of the considered problems are based on the local behavior described by another deep tool of universal algebra known as *tame congruence theory*. This theory, created and described by D. Hobby and R. McKenzie in [16], is a perfect tool for studying the local structure of finite algebras. Instead of considering the whole algebra and all of its operations at once, tame congruence theory allows us to localize to small subsets on which the structure is much simpler to understand and to handle. According to this theory there are only five possible ways a finite algebra can behave locally. The local behavior must be one of the following:

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

For an algebra $\mathbf{A}$ the set $\mathrm{typ}\{\mathbf{A}\} \subseteq \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}\}$ consists of types describing local behavior in $\mathbf{A}$.

Now, if from our point of view a local behavior of an algebra is 'bad' then we can often show that the algebra itself behaves 'badly'. For example, since CSAT or CEQV is intractable in 2-element Boolean algebra one can argue that in any finite algebra with tractable CSAT or CEQV type $\mathbf{3}$ cannot occur.

Theorem 3.1. *If $\mathbf{A}$ is finite algebra from a congruence modular variety such that $\mathbf{3} \in \mathrm{typ}\{\mathbf{A}\}$, then CSAT($\mathbf{A}$) is NP-complete and CEQV($\mathbf{A}$) is co-NP-complete.*

For a finite algebra $\mathbf{A}$ from a congruence modular variety we have $\mathrm{typ}\{\mathbf{A}\} \subseteq \{\mathbf{2}, \mathbf{3}, \mathbf{4}\}$. Thus in view of theorem 3.1 we are left with an analysis of different kinds of interactions between local behaviors of types $\mathbf{2}$ and $\mathbf{4}$. First we can separate these two types: the abelian type $\mathbf{2}$ and the lattice (non-abelian) type $\mathbf{4}$.

Theorem 3.2. *Let $\mathbf{A}$ be a finite subdirectly irreducible algebra from a congruence modular variety with an abelian monolith. Then $\mathbf{A}$ is solvable i.e. $\mathrm{typ}\{\mathbf{A}\} \subseteq \{\mathbf{2}\}$ or CSAT($\mathbf{A}$) is NP-complete.*

Theorem 3.3. *Let $\mathbf{A}$ be a finite subdirectly irreducible algebra from a congruence modular variety with a non-abelian monolith. Then $\mathrm{typ}\{\mathbf{A}\} \subseteq \{\mathbf{4}\}$ or CSAT($\mathbf{A}$) is NP-complete.*

Slightly more technical (but equivalent) conditions than those described in Theorems 3.2 and 3.3 form an important step towards the description described in Theorem 2.11.

Theorem 3.4. *If $\mathbf{A}$ is finite algebra from a congruence modular variety then either $\mathbf{A}$ is isomorphic to a direct product $\mathbf{S} \times \mathbf{D}$, where $\mathrm{typ}\{\mathbf{S}\} \subseteq \{\mathbf{2}\}$ and $\mathrm{typ}\{\mathbf{D}\} \subseteq \{\mathbf{4}\}$ or CSAT($\mathbf{A}$) is NP-complete.*

Now in view of Theorem 3.4 it suffices to understand solvable algebras (i.e. those with typset $\{\mathbf{2}\}$) and algebras with typset $\{\mathbf{4}\}$. With the help of another interpretation we can show that solvable algebra have in fact to be nilpotent.

Theorem 3.5. *If a finite algebra $\mathbf{A}$ from a congruence modular variety is solvable but not nilpotent then $\mathbf{A}$ has a homomorphic image $\mathbf{A}'$ with CSAT($\mathbf{A}'$) being NP-complete.*

The control of type $\mathbf{4}$ behavior with tractable CSAT is much more involved. We show that every subdirectly irreducible algebra of type $\mathbf{4}$ is simple and in fact has only 2 elements. Thus we have:

Theorem 3.6. *Let $\mathbf{A}$ be a finite algebra from a congruence modular variety and $\mathrm{typ}\{\mathbf{A}\} = \{\mathbf{4}\}$. Then either $\mathbf{A}$ is a subdirect product of 2-element algebras each of which is polynomially equivalent to the 2-element lattice, or $\mathbf{A}$ has a subdirectly irreducible homomorphic image $\mathbf{A}'$ such that CSAT($\mathbf{A}'$) is NP-complete.*

Now combining Theorem 3.4 with Theorem 3.5 and Theorem 3.6 we get the first part of Theorem 2.11.

The proof of the second part of Theorem 2.11 splits into two cases. We show that for both factors of **A**, namely **N** and **D**, the problem has polynomial time solution. Actually we will show that in both cases if the polynomial equation $\mathbf{t}(\overline{x}) = \mathbf{s}(\overline{x})$ has a solution $\overline{x} = (x_1, \dots, x_n) \in A^n$ then it has a solution in a relatively small subset S of A^n, namely in a subset with size bounded by a polynomial in n. The reader should be however warned here that we are not claiming that all solutions are contained in this small set S.

The size of set $S \subseteq A^n$ to which we reduce our search for a solution to $\mathbf{t}(x_1, \dots, x_n) = \mathbf{s}(x_1, \dots, x_n)$ is bounded by a polynomial depending on n, i.e. on the number of variables in **t** and **s**, and in other words on the number of inputs gates in corresponding circuit. In fact the size of term/polynomials **t** and **s** or the corresponding circuit can be arbitrary larger then the number n (of variables or input gates). Thus our arguments do not depend whether the size of the input is the size of circuit (graph) or the size of corresponding polynomial (length of the syntactic expression).

First we show how to find this small sets in the factor **D**, i.e. we prove the following.

Theorem 3.7. *Let* **D** *be a subdirect product of finitely many* 2*-element algebras each of which is polynomially equivalent to the* 2*-element lattice. Then* CSat(**D**) *is solvable in polynomial time.*

Proof. The basic observation is that for the 2-element lattice **L**, and therefore for every algebra polynomially equivalent to the 2-element lattice, the problem CSat(**L**) is solvable in polynomial time by a very special algorithm.

Indeed, if $\mathbf{t}, \mathbf{s} \in \operatorname{Pol} \mathbf{L}$ the equation $\mathbf{t}(\overline{x}) = \mathbf{s}(\overline{x})$ has a solution, say $(a_1, \dots, a_n)$, then both $\mathbf{t}(a_1, \dots, a_n)$ and $\mathbf{s}(a_1, \dots, a_n)$ have the same value a. But for a polynomial **t** over the 2-element lattice one can easily show, that if $\mathbf{t}(a_1, \dots, a_n) = a$ then $\mathbf{t}(a, \dots, a) = a$. Indeed, by the monotonicity of the polynomials of **L** we have

$$\mathbf{t}(0, \dots, 0) \leqslant \mathbf{t}(a_1, \dots, a_n) \leqslant \mathbf{t}(1, \dots, 1)$$

and if $\mathbf{t}(a_1, \dots, a_n) = 0$ then $\mathbf{t}(0, \dots, 0)$ has to be 0 as well. Similarly $\mathbf{t}(a_1, \dots, a_n) = 1$ implies $\mathbf{t}(1, \dots, 1) = 1$.

Therefore, to determine if $\mathbf{t}(\overline{x}) = \mathbf{s}(\overline{x})$ has a solution over **L** it suffices to show whether $\mathbf{t}(a, \dots, a) = \mathbf{s}(a, \dots, a)$ for some $a \in L$.

We say that an algebra **A** has Uniform Solution Property, or USP for short, if for every polynomial $\mathbf{t}(\overline{x}) \in \operatorname{Pol} \mathbf{A}$ and $a \in A$

$$\big(\exists \overline{x} \;\; \mathbf{t}(x_1, \dots, x_n) = a\big) \Rightarrow \mathbf{t}(a, \dots, a) = a$$

What we have just shown is that the 2-element lattice has USP, and that CSat(**A**) is polynomially time solvable for every finite algebra **A** with USP.

Now we can conclude the proof by noting that a subdirect product of algebras with USP, has USP itself. Actually USP is preserved under forming homomorphic images, subalgebras, products or reducts. □

The reduction of searching a solution of an equation in supernilpotent realm to a relatively small set is much more involved than in lattice case. Our proof is modeled after the Ramsey type argument introduced by Mikael Goldmann and Alexander Russell in [11] for nilpotent groups, and later cleaned up by Gábor Horváth [17] in the realm of nilpotent groups and nilpotent rings.

Theorem 3.8. *Let* **A** *be a finite supernilpotent algebra from a congruence modular variety. Then* CSat(**A**) *is solvable in polynomial time.*

Proof. The idea of our construction of the set S relies on the following property of supernilpotent algebras

(∗) For every finite supernilpotent algebra **A** and $a \in A$ there is a positive integer d such that every equation of the form $\mathbf{w}(\overline{x}) = a$ has a solution iff it has a solution such that $|\{i \colon x_i \neq a\}| \leqslant d$.

Given (∗) we simply check if $\mathbf{w}(x_1, \dots, x_n) = a$ has a solution among $\binom{n}{d} \cdot |A|^d$ possible evaluations of the x_i's with $|\{i \colon x_i \neq a\}| \leqslant d$. Unfortunately the degree d of the polynomial bounding the run time of the algorithm can be really huge, as it is obtained by a Ramsey type argument applied to the numbers:

- k – the degree of supernilpotency of the algebra **A**,
- $C = |A|^{k \cdot |A|}$,
- $m = (k-1)! \cdot |A|$

to get that:

(∗∗) There is a positive integer d such that for every set S with $|S| \geqslant d$ and every coloring of all at most $(k-1)$-element subsets of S with C colors there exists m-element subset T of S such that all at most $(k-1)$-element subsets of T with the same number of elements have the same color.

□

Very recently the authors have been informed by M. Kompatscher about his independent proof of Theorem 3.8 (see [29])

4 Conclusions and open problems

A short informal summary of these results is completed in the following table, where 'DL-like' stays for being a subdirect product of algebras polynomially equivalent to 2-element lattices.

	tractable	open	intractable
CEqv	supernilpotent Aichinger & Mudrinski [1]	nil but not supernil	non nilpotent Thm 2.15
CSat	supernil×DL-like Thm 2.11 (2)	nil but not supernil	non (nil×DL-like) Thm 2.11 (1)
MCSat	affine×DL-like Cor 2.14 (2)	–	otherwise Cor 2.14 (1)
SCSat	affine Gaussian elimination	_	otherwise Larose, Zádori [30]

An obvious open question is the following:

Problem 2. *Determine the computational complexity of* CEqv *and* CSat *for nilpotent, but not supernilpotent finite algebras from congruence modular varieties.*

Two polynomial time algorithms presented in this paper i.e. in the proofs of Theorem 3.7 and Theorem 3.8 are based on a similar idea. We prove 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). Moreover our proofs show that set S depends only on the number of variables occurring in the equation but not on the structure/syntax of the equation. Now, to decide the existence of solution the algorithms evaluate the polynomials (i.e compute the values on output gates) on tuples from this small set S. It seems however that in the nilpotent but not supernilpotent setting there is no chance for a polynomial time algorithm for CSat or CEqv based on this kind of ideas. In fact our results contained in [25] confirm this claim. We proved that if $P \neq NP$ then for some nilpotent but not supernilpotent algebras $\mathbf{A}$ there is no polynomial time algorithm which solves CSat($\mathbf{A}$) by reducing to the search space S depending only on the number of variables.

On the other hand we provide in [25] polynomial time algorithms for CEqv$(\mathbf{B}_{pq})$ and CSat$(\mathbf{B}_{p2})$, where B_{pq} are algebras defined in Example 2.13, and we believe that CSat and CEqv for all finite nilpotent algebras (of finite type) can be shown to be in P.

Another question that arises naturally is the role of quotient algebras in the proofs of NP-completeness of considered problems. Note that the result of B. Larose and L. Zádori [30] for SCSat makes no use of quotient algebras. This is because a quotient of an affine algebra is affine itself.

Example 2.10 shows that in general it is not enough to establish NP-completeness for a quotient algebra to conclude the hardness for the original one. However it may suffice in some more restricted settings like for example congruence modularity. In concrete algebraic structures where basic operations are described explicitly it might be much easier. In fact in structures described in Examples 2.4, 2.5 and 2.6, passing to quotients is hidden in the hardness proofs and (implicitly) replaced by an involved control over congruences in groups, rings or lattices, respectively.

Problem 3. *Is it true that NP-completeness of* CSat *for some quotient of a finite algebra* $\mathbf{A}$ *from a congruence modular variety implies NP-completeness of* CSat *for* $\mathbf{A}$ *itself.*

Even if the answer to Problem 3 would be negative the next one remains open.

Problem 4. *Do the characterizations of Theorems 2.11 (1), 2.15 and Corollary 2.14 (1) remain true without passing to quotient algebras.*

Note here that when restricting to equations of the form

$$\mathbf{t}(x_1, \ldots, x_n) = c$$

where $\mathbf{t}$ is a polynomial but c is a constant, the satisfiability in the quotient $\mathbf{A}/\theta$ reduces to the satisfiability of at least one of the equations in the following disjunction

$$\mathbf{t}(x_1, \ldots, x_n) = c_1 \quad \vee \ \ldots \ \vee \quad \mathbf{t}(x_1, \ldots, x_n) = c_s,$$

where $\{c_1, \ldots, c_s\}$ is the equivalence class of c modulo θ. This Cook style reduction gives the hope to attack the following problem.

Problem 5. *Characterize finite algebras* $\mathbf{A}$ *for which determining the existence of a solution to the equations of the form* $\mathbf{t}(\overline{x}) = c$ *can be done in polynomial time.*

In view of Problem 1, the natural conjecture about dichotomy for CSat is not so evident. However there is a slightly bigger hope for such dichotomy after:

- restricting CSat to the equations of the form $\mathbf{t}(\overline{x}) = c$, and
- relaxing many-to-one reductions to Cook reductions.

Problem 6. *Prove the dichotomy in the above settings.*

References

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

[2] Libor Barto and Marcin Kozik. 2014. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM (JACM)* 61, 1 (2014), 3.

[3] Przemysław Broniek. 2015. *Computational Complexity of Solving Equation Systems.* Springer.

[4] Andrei A. Bulatov. 2000. On the number of finite Mal'tsev algebras. *Contributions to general algebra* 13 (2000), 41–54.

[5] Andrei A. Bulatov. 2017. A Dichotomy Theorem for Nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, Vol. 00. 319–330. https://doi.org/10.1109/FOCS.2017.37

[6] Stanley Burris and John Lawrence. 1993. The equivalence problem for finite rings. *Journal of Symbolic Computation* 15, 1 (1993), 67–71.

[7] Stanley Burris and John Lawrence. 2005. Results on the equivalence problem for finite groups. *Algebra Universalis* 52, 4 (2005), 495–500.

[8] Tomás Feder, Florent Madelaine, and Iain A Stewart. 2004. Dichotomies for classes of homomorphism problems involving unary functions. *Theoretical Computer Science* 314, 1-2 (2004), 1–43.

[9] Tomás Feder and Moshe Y. Vardi. 1998. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.* 28, 1 (1998), 57–104. http://dx.doi.org/10.1137/S0097539794266766

[10] Ralph Freese and Ralph McKenzie. 1987. *Commutator theory for congruence modular varieties.* London Mathematical Society Lecture Note Series, Vol. 125. Cambridge University Press, Cambridge. iv+227 pages.

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

[12] Tomasz A. Gorazd and Jacek Krzaczkowski. 2010. Term equation Satisfiability over Finite Algebras. *IJAC* 20, 8 (2010), 1001–1020. http://dx.doi.org/10.1142/S021819671000600X

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

[14] Heinz Peter Gumm. 1983. *Geometrical methods in congruence modular algebras.* Vol. 286. American Mathematical Soc.

[15] Joachim Hagemann and Christian Herrmann. 1979. A concrete ideal multiplication for algebraic systems and its relation to congruence distributivity. *Archiv der Mathematik* 32, 1 (1979), 234–245.

[16] David Hobby and Ralph McKenzie. 1988. *The structure of finite algebras.* Contemporary Mathematics, Vol. 76. American Mathematical Society, Providence, RI. xii+203 pages.

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

[18] Gábor Horváth. 2015. The complexity of the equivalence and equation solvability problems over meta-Abelian groups. *Journal of Algebra* 433 (2015), 208–230.

[19] Gábor Horváth, László Mérai, Csaba Szabó, and John Lawrence. 2007. The complexity of the equivalence problem for nonsolvable groups. *Bulletin of the London Mathematical Society* 39, 3 (2007), 433–438.

[20] Gábor Horváth and Csaba Szabó. 2006. The Complexity of Checking Identities over Finite Groups. *IJAC* 16, 5 (2006), 931–940. http://dx.doi.org/10.1142/S0218196706003256

[21] Gábor Horváth and Csaba Szabó. 2011. The extended equivalence and equation solvability problems for groups. *Discrete Mathematics & Theoretical Computer Science* Vol. 13 no. 4 (Jan. 2011). http://dmtcs.episciences.org/536

[22] Gábor Horváth and Csaba Szabó. 2012. Equivalence and equation solvability problems for the alternating group A4. *Journal of Pure and Applied Algebra* 216, 10 (2012), 2170–2176.

[23] Harry B Hunt III and Richard Edwin Stearns. 1990. The complexity of equivalence for commutative rings. *Journal of Symbolic Computation* 10, 5 (1990), 411–436.

[24] Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. 2010. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.* 39, 7 (2010), 3023–3037.

[25] Paweł M. Idziak, Piotr Kawałek, and Jacek Krzaczkowski. 2018. Expressive power, satisfiability and equivalence of circuits over nilpotent algebras. (2018).

[26] Keith A. Kearnes. 1999. Congruence modular varieties with small free spectra. *Algebra Universalis* 42, 3 (01 Oct 1999), 165–181. https://doi.org/10.1007/s000120050132

[27] Ondřej Klíma. 2009. Complexity issues of checking identities in finite monoids. *Semigroup Forum* 79, 3 (22 Aug 2009), 435. https://doi.org/10.1007/s00233-009-9180-y

[28] Ondřej Klíma, Pascal Tesson, and Denis Thérien. 2007. Dichotomies in the complexity of solving systems of equations over finite semigroups. *Theory of Computing Systems* 40, 3 (2007), 263–297.

[29] Michael Kompatscher. 2017. The equation solvability problem over nilpotent Mal'cev algebras. (Oct. 2017). arXiv:1710.03083

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

[31] Yuri V. Matijasevič. 1970. The Diophantineness of enumerable sets. *Dokl. Akad. Nauk SSSR* 191 (1970), 279–282.

[32] Ralph McKenzie, George F McNulty, and Walter Taylor. 1987. *Algebras, lattices, varieties, Volume I.* Wadsworth & Brooks/Cole Advanced Books & Software Monterey, California.

[33] Thomas J. Schaefer. 1978. The complexity of satisfiability problems. In *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978)*. ACM, New York, 216–226.

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

[35] Jonathan DH Smith. 1976. Mal'cev varieties. (1976).

[36] László Zádori. 2011. On solvability of systems of polynomial equations. *Algebra universalis* 65, 3 (2011), 277–283.

[37] Dmitriy Zhuk. 2017. A Proof of CSP Dichotomy Conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, Vol. 00. 331–342. https://doi.org/10.1109/FOCS.2017.38