# Solving equations over small unary algebras

Przemyslaw Broniek

*Computational Logic and Applications, CLA '05* DMTCS proc. **AF**, 2006, 49–60

# *Solving equations over small unary algebras*

Przemyslaw Broniek

*broniek@ii.uj.edu.pl, Algorithmics Research Group, Jagiellonian University, Krakow, Poland*

We consider the problem of solving a system of polynomial equations over fixed algebra $\mathbf{A}$ which we call MPolSat($\mathbf{A}$). We restrict ourselves to unary algebras and give a partial characterization of complexity of MPolSat($\mathbf{A}$). We isolate a preorder $\mathbf{P}(\mathbf{A})$ to show that when $\mathbf{A}$ has at most 3 elements then MPolSat($\mathbf{A}$) is in P when width of $\mathbf{P}(\mathbf{A})$ is at most 2 and is NP-complete otherwise. We show also that if P $\neq$ NP then the class of unary algebras solvable in polynomial time is not closed under homomorphic images.

**Keywords:** algebra, SAT, computational complexity, dichotomy

## 1 Introduction

We consider the problem of solving a system of polynomial equations over fixed algebra $\mathbf{A}$ which we call MPolSat($\mathbf{A}$). The complexity of this problem has been studied and solved for several classes of algebras but it is still open in general. A study of this problem can be found in [5]. However they consider algebras with sufficiently many non-unary operations. Namely the existence of the so called Taylor term is needed to apply the results of [5]. Their characterization of algebras $\mathbf{A}$ with polynomial time algorithms for MPolSat($\mathbf{A}$) may be applied only if the variety HSP($\mathbf{A}$) generated by $\mathbf{A}$ omits type $\mathbf{1}$ in the sense of Tame Congruence Theory of Hobby and McKenzie [4]. This paper starts to fill this gap and eventually incorporate type $\mathbf{1}$. To do this one needs to understand first the algebras where all operations are unary.

On the other hand the case $|A| = 3$ is covered by the transformation of MPolSat to Constraint Satisfaction Problem (CSP) described in [5] and then by applying a deep result of Bulatov [1]. Since this approach theoretically works in the general setting it produces a characterization that is messy and hard to follow. Therefore we decided to use our own approach for unary algebras. The class of unary algebras has been studied in slightly different context in [3] and is also interesting for us because of the phenomena that solving one equation over unary algebras is very simple, but the complexity of checking whether systems of equations are solvable (decision version of solving equations) could happen to be NP-complete.

We fully characterize the complexity of MPolSat($\mathbf{A}$) when $\mathbf{A}$ has at most 3 elements. We isolate a preorder $\mathbf{P}(\mathbf{A})$ associated with an algebra and our condition for complexity considers only width of $\mathbf{P}(\mathbf{A})$ thus our approach is more compact. This approach, omitting Bulatov's [1] characterization, seems to be more transparent for unary algebras. In particular we believe that our approach via width can be extended for solving equations over unary algebras of arbitrary size.

We also construct a special class of unary algebras to show that if P $\neq$ NP then the class of unary algebras solvable in polynomial time is not closed under homomorphic images.

# 2 Basic definitions

An *algebra* $\mathbf{A}$ is an ordered pair $(A, F)$ where $A$ is a non-empty set called *carrier* and $F$ is a family of finitary operations $f : A^n \mapsto A$. For more detailed description of algebraic terminology the reader is referred to [2]. We call an algebra $\mathbf{A}$ *unary* if all of its operations in $F$ are unary.

**Example 2.1** $\mathbf{A} = (\{0, 1, 2\}, f, g, h)$ *with the following operations is unary.*

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ |
|---|---|---|---|
| *0* | *1* | *0* | *0* |
| *1* | *0* | *1* | *0* |
| *2* | *0* | *0* | *1* |

The *term* over $\mathbf{A}$ is any proper expression build with operations from $F$ over some set of variables. The polynomial over $\mathbf{A}$ is a term in which some variables are replaced by constants from $A$. For a unary algebra $\mathbf{A}$ by $F^*$ we denote the *term-monoid* of $\mathbf{A}$, i.e., all (unary) terms of $\mathbf{A}$ with composition.

**Example 2.2** *For the algebra from the Example 2.1 the term-monoid contains* 9 *elements:*

| $x$ | $id$ | $f(x)$ | $g(x)$ | $h(x)$ | ***0*** | ***1*** | $i(x)$ | $j(x)$ | $k(x)$ |
|---|---|---|---|---|---|---|---|---|---|
| *0* | *0* | *1* | *0* | *0* | *0* | *1* | *0* | *1* | *1* |
| *1* | *1* | *0* | *1* | *0* | *0* | *1* | *1* | *0* | *1* |
| *2* | *2* | *0* | *0* | *1* | *0* | *1* | *1* | *1* | *0* |

By $Ker(f) = \{(x, y) \in A^2 : f(x) = f(y)\}$ we denote the *kernel* of $f$.

**Definition 2.3** *For a unary algebra* $\mathbf{A} = (A, F)$ *we define a* preorder $\leqslant$ *on* $F^*$ *by putting* $f \leqslant g$ *if and only if* $Ker(f) \subseteq Ker(g)$. *We also put* $\mathbf{P}(\mathbf{A}) = (F^*, \leqslant)$.

**Example 2.4** *In the algebra* $\mathbf{A} = (\{0, 1, 2\}, f, g, h)$*, with the operations:*

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ |
|---|---|---|---|
| *0* | *0* | *0* | *0* |
| *1* | *0* | *1* | *0* |
| *2* | *0* | *0* | *1* |

*we have:* $g \leqslant f, h \leqslant f$ *and* $g, h$ *are incomparable.*

By the width of an ordered set (or more general preordered set) $\mathbf{P}$ we mean the largest number of pairwise incomparable elements of $\mathbf{P}$.

In the following problems we assume that the algebra $\mathbf{A}$ is *never* part of the input.

**Definition 2.5** Problem POLSAT($\mathbf{A}$):
*Input:* Two polynomials $t$ and $s$ over the algebra $\mathbf{A}$
*Question:* Does the equation $t = s$ have a solution?

**Definition 2.6** Problem MPOLSAT($\mathbf{A}$):
*Input:* A finite set of equations $S = \{t_i = s_i | i = 1, \ldots, n\}$, where $t_i, s_i$ are polynomials over the algebra $\mathbf{A}$.
*Question:* Does the system $S$ of equations have a solution?

Note that replacing any composition of basic operations by a single operation symbol from $F^*$ reduces the size of input. Thus the problems MPOLSAT$(A, F)$ and MPOLSAT$(A, F^*)$ are polynomially equivalent. We also consider the following auxiliary problem which allows us to formulate our algorithms in a more transparent way:

**Definition 2.7** Problem CMPOLSAT$(\mathbf{A})$:
*Input:* A set of variables $V$, a finite set of equations $S = \{t_i = s_i | i = 1, \ldots, n\}$, where $t_i, s_i$ are polynomials over the algebra $\mathbf{A}$, and a constraint function $C : V \mapsto 2^A$
*Question:* Does $S$ have a solution $J : V \mapsto A$ such that $J(v) \in C(v)$ for each $v \in V$?

Of course, if for any variable $v$ the set $C(v)$ is empty then there is no solution. On the other hand for any $v$ if $C(v) = A$ then there is in fact no constraint at all on $v$. CMPOLSAT$(\mathbf{A})$ is a generalization of MPOLSAT$(\mathbf{A})$. An instance $I = (V, S, C)$ of the problem of CMPOLSAT$(\mathbf{A})$ with $C(v) \equiv A$ is equivalent to the instance $I' = S$ of MPOLSAT$(\mathbf{A})$. It is worth noting that there exists an algebra $\mathbf{A}'$ for which the problem CMPOLSAT$(\mathbf{A}')$ is NP-complete while MPOLSAT$(\mathbf{A}')$ is in P, which we show in Proposition 5.2.

We say that two instances $I_1, I_2$ of CMPOLSAT$(\mathbf{A})$ are equivalent if $I_1$ has a solution if and only if $I_2$ does. The $size(I)$ of an instance $I = (V, S, C)$ of CMPOLSAT$(\mathbf{A})$ is a pair (number of variables in $V$, number of equations in $S$).

# 3 Unary algebras

It is known, see e.g. [5], that there are finite algebras $\mathbf{A}$ for which MPOLSAT$(\mathbf{A})$ is NP-complete. Our first lemma shows that it remains NP-complete even for some unary algebras (even with 3 elements).

**Lemma 3.1** *There are finite unary algebras for which* MPOLSAT$(\mathbf{A})$ *is* NP*-complete.*

**Proof:** A similar argument can be found in [3]. It is easy to check that MPOLSAT$(\mathbf{A})$ belongs to NP, because we can simply check whether given assignment to variables satisfies all the equations.

Let $\mathbf{A} = (\{0, 1, 2\}, f, g, h)$ be an algebra with the following operations:

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ |
|---|---|---|---|
| 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 2 | 0 | 0 | 1 |

Next we need the following NP-complete version of SAT problem:

POSITIVE 1-IN-3-SAT is a problem taking on its input a formula $F = C_1 \wedge \cdots \wedge C_n$, in which each clause $C_i$ is of the form $(x \vee y \vee z)$, where $x, y, z$ are (non-negated) variables, and answering the question if there is a boolean valuation such that in each clause *exactly one* variable takes value 1. For example for a formula $(x \vee y \vee z) \wedge (x \vee t \vee v) \wedge (v \vee t \vee z)$ the positive answer can be witnessed by $(0 \vee \mathbf{1} \vee 0) \wedge (0 \vee 0 \vee \mathbf{1}) \wedge (\mathbf{1} \vee 0 \vee 0)$.

Next we reduce POSITIVE 1-IN-3-SAT to systems of equations of our algebra. A formula $F = C_1 \wedge \cdots \wedge C_n$ is transformed into $3n$ equations as follows:

$$C_i = x \vee y \vee z \quad \rightsquigarrow \quad \begin{cases} f(v_i) = x \\ g(v_i) = y \\ h(v_i) = z \end{cases},$$

where $v_1, \dots, v_n$ are new variables not occurring in $F$.

It is easy to check that a solution of the equations exists if and only if the formula $F$ is satisfiable according to POSITIVE 1-IN-3-SAT rules. If $v_i$ takes the value 0 (1 or 2) then only $x$ ($y$ or $z$ respectively) takes boolean value 1 to make $C_i$ true. □

From the next proposition we know how CMPOLSAT($\mathbf{A}$) is connected with our main problem MPOLSAT($\mathbf{A}$):

**Proposition 3.2** *For each algebra $\mathbf{A} = (A, F)$ there exists algebra $\mathbf{A}'$ such that* CMPOLSAT($\mathbf{A}$) *is polynomially reducible to* MPOLSAT($\mathbf{A}'$).

**Proof:** If $|A| = 1$ the thesis is trivial because all the constraints are trivial. Otherwise we construct $\mathbf{A}'$ by adding one operation $f_X$ for each $X \subset A$. Let $0, 1$ will be two different elements of $A$. We put:

$$f_X(v) = \begin{cases} 1, & v \in X \\ 0, & \text{otherwise} \end{cases}$$

Because $\mathbf{A}$ is not a part of the input we need not to worry about exponentially (from $|A|$) many operations added. Now, given an instance $(V, S, C)$ of CMPOLSAT($\mathbf{A}$) we reduce it to MPOLSAT($\mathbf{A}'$) by adding to equations in the set $S$ one additional equation of the form $f_{C(v)}(v) = 1$ for each variable $v \in V$. The solution of the new set of equations will satisfy constraints $C$. □

In contrast to Lemma 3.1 solving one equation over unary algebra is always easy:

**Proposition 3.3** *If $\mathbf{A}$ is unary then* POLSAT($\mathbf{A}$) *is in* P.

**Proof:** There are at most two variables involved into each equation, since equations over unary algebra are of the form:

$$f_1(\dots f_n(x)) = g_1(\dots g_m(y)),$$

where $f_i, g_i$ are basic operations of $\mathbf{A}$ and $x, y$ are variables. We can simply check all the possibilities in $O(n|A|^2)$ time. □

# 4 Small algebras with MPOLSAT in P

First we introduce an algorithm that simplifies instances of MPOLSAT by changing them to instances of CMPOLSAT. Next we state our characterization theorem for at most three element algebras.

## 4.1 Simplification algorithm

We construct a simplification algorithm $SA$ which takes an instance $I$ as its input and produces instance $SA(I)$ with the properties:

1. $SA(I)$ is equivalent to $I$,
2. $size(SA(I)) \leqslant size(I)$, where $\leqslant$ is the lexicographic order on $\mathbf{N} \times \mathbf{N}$,
3. The only equations in $SA(I)$ are of the form $p(x) = r(y)$, where $p, r$ are polynomials over $\mathbf{A}$ and $x, y$ are different variables. Both $p$ and $r$ are not permutations and $|p(C(x)) \cap r(C(y))| > 1$,

4. $|C(x)| > 1$ for each variable occurring in $SA(I)$.

The algorithm $SA$ removes some equations, according to inference rules presented below, as long as possible. We will see that $SA$ works in a polynomial time. Since $SA(I)$ is equivalent to $I$ we get the answer for $I$ by considering $SA(I)$, which will be done in Section 4.2. Actually one can transform any solution of $SA(I)$ to a solution of $I$.

Let $I = (V, S, C)$ be an instance of CMPOLSAT with $V$ - set of variables, $S$ - set of equations, $C$ - constraint function. By $\perp$ we denote that a given instance has no solution. For a set of equations $S$, a variable $x$ and a term $t$ by $S[x/t]$ we denote set of equations $S$ in which every occurrence of a variable $x$ has been replaced by $t$.

**Simplification Algorithm SA(I):**

FAIL:
$(V, S, C) \to \perp$ if $v$ is a variable and $C(v) = \emptyset$

CST1:
$(V, S \cup \{c_1 = c_2\}, C) \to (V, S, C)$ if $c_1$ and $c_2$ are the same constants

CST2:
$(V, S \cup \{c_1 = c_2\}, C) \to \perp$ if $c_1$ and $c_2$ are different constants

VAR1:
$(V, S \cup \{x = c\}, C) \to \perp$ if $c$ is a constant, $x$ is a variable and $c \notin C(x)$

VAR2:
$(V, S \cup \{x = c\}, C) \to (V \setminus \{x\}, S[x/c], C)$ if $c$ is a constant, $x$ is a variable and $c \in C(x)$

VARS1:
$(V, S \cup \{x = y\}, C) \to (V, S, C)$ if $x$ and $y$ are the same variables

VARS2:
$(V, S \cup \{x = y\}, C) \to (V \setminus \{x\}, S[x/y], C')$ if $x$ and $y$ are different variables;
$C' := C$ except $C'(y) := C(x) \cap C(y)$

VARPOL1:
$(V, S \cup \{p(x) = c\}, C) \to (V, S, C')$ if $x$ is a variable;
$C' := C$ except $C'(x) := C(x) \cap p^{-1}(c)$

VARPOL2:
$(V, S \cup \{p(x) = x\}, C) \to (V, S, C')$ if $x$ is a variable;
$C' := C$ except $C'(x) := C(x) \cap \{a \in A : p(a) = a\}$

VARPOL3:
$(V, S \cup \{p(y) = x\}, C) \to$
$\to (V \setminus \{x\}, S[x/p(y)], C')$ if $x, y$ are different variables;
$C' := C$ except $C'(y) := C(y) \cap p^{-1}(C(x))$

POL1:
$(V, S \cup \{p(x) = r(x)\}, C) \to (V, S, C')$ if $x$ is a variable;
$C' := C$ except $C'(x) := C(x) \cap \{a \in A : p(a) = r(a)\}$

Applying all of the above rules as many times as possible all equations are of the form $p(x) = r(y)$ where $p, r$ are polynomials over $\mathbf{A}$ and $x, y$ are different variables. But we can still remove equations by the following rules:

POL2:
$(V, S \cup \{p(x) = r(y)\}, C) \to \bot$ if $x, y$ are different variables and $p(C(x)) \cap r(C(y)) = \emptyset$

POL3:
$(V, S \cup \{p(x) = r(y)\}, C) \to (V, S \cup \{p(x) = c, r(y) = c\}, C)$ if $x, y$ are different variables and $p(C(x)) \cap r(C(y)) = \{c\}$

The equations $p(x) = c$ and $r(y) = c$ will be removed by the rule VARPOL1, so that the instance eventually will be decreased.

POL4:
$(V, S \cup \{p(x) = r(y)\}, C) \to$ if $x, y$ are different variables and $p$ is a permutation of the algebra;
$\to (V \setminus \{x\}, S[x/h(y)], C')$ $h := p^{k-1}r$ $(\star)$, $C' := C$ except $C'(y) := C(y) \cap h^{-1}(C(x))$

$(\star)$ It is easy to see that there exists $k$ such that $p^k(x) = x$ thus $p^k(x) = p^{k-1}(r(y))$ which is equivalent to $x = p^{k-1}(r(y))$, thus we act like in the VARPOL3 rule.

The simplification algorithm works in $O(n^2)$ time, where $n$ is size of the input. Each of the rules transforms instance to a smaller one, so that the number of inference steps is at most linear. Each step can be done in linear time.

## 4.2 Characterization theorem

**Lemma 4.1** *For a unary algebra* $\mathbf{A} = (A, F)$ *with every* $f \in F$ *being a constant or a permutation* $\text{MPOLSAT}(\mathbf{A}) \in \mathsf{P}$.

**Proof:** We apply our simplification algorithm $SA$. Because all the operations in $F$ and thus in $F^*$ are constants or permutations $SA$ finishes with empty set of equations (then the solution exists) or breaks with result $\bot$ meaning that there is no solution. □

For our main proof we need two following lemmas:

**Lemma 4.2** *For a unary algebra* $\mathbf{A}$ *with 3 elements the problem* $\text{MPOLSAT}(\mathbf{A})$ *is* $\mathsf{NP}$*-complete if* $width(\mathbf{P}(\mathbf{A})) = 3$.

**Proof:**
To witness width 3 in the preorder $\mathbf{P}(\mathbf{A})$ the algebra $\mathbf{A}$ must have 3 polynomials $f_0, f_1, f_2$ with pairwise incomparable kernels. Without loss of generality we may assume that $f_0, f_1, f_2$ act as follows $(a_i \neq b_i)$:

| $x$ | $f_0(x)$ |
|---|---|
| 0 | $b_0$ |
| 1 | $a_0$ |
| 2 | $a_0$ |

| $x$ | $f_1(x)$ |
|---|---|
| 0 | $a_1$ |
| 1 | $b_1$ |
| 2 | $a_1$ |

| $x$ | $f_2(x)$ |
|---|---|
| 0 | $a_2$ |
| 1 | $a_2$ |
| 2 | $b_2$ |

We are going to show that there are $f, g, h \in F^*$ and $\bot \neq \top$ in $\mathbf{A}$ such that:

P1

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ |
|---|---|---|---|
| 0 | $\top$ | $\bot$ | $\bot$ |
| 1 | $\bot$ | $\top$ | $\bot$ |
| 2 | $\bot$ | $\bot$ | $\top$ |

P2

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ |
|---|---|---|---|
| 0 | $\bot$ | $\bot$ | $\bot$ |
| 1 | $\top$ | $\top$ | $\bot$ |
| 2 | $\top$ | $\bot$ | $\top$ |

- Case 1: There exists $i \in \{0,1,2\}$ such that $i \neq b_i$.

  Without loss of generality we can assume that $i = 0$ and $b_0 = 1$. Then for $f_3 := f_1 f_0 \in F^*$ we have:

| $x$ | $f_0(x)$ | $f_1(x)$ | $f_2(x)$ | $f_3(x)$ |
|---|---|---|---|---|
| 0 | 1 | $a_1$ | $a_2$ | $b_1$ |
| 1 | $a_0$ | $b_1$ | $a_2$ | $a_1$ |
| 2 | $a_0$ | $a_1$ | $b_2$ | $a_1$ |

  – Subcase 1.1: $\{a_1, b_1\} \neq \{0,1\}$. Then $f_2, f_2 f_1, f_2 f_3$ satisfy:

| $x$ | $f_2(x)$ | $f_2(f_1((x))$ | $f_2(f_3((x))$ |
|---|---|---|---|
| 0 | $a_2$ | $a_2$ | $b_2$ |
| 1 | $a_2$ | $b_2$ | $a_2$ |
| 2 | $b_2$ | $a_2$ | $a_2$ |

*or*

| $x$ | $f_2(x)$ | $f_2(f_1((x))$ | $f_2(f_3((x))$ |
|---|---|---|---|
| 0 | $a_2$ | $b_2$ | $a_2$ |
| 1 | $a_2$ | $a_2$ | $b_2$ |
| 2 | $b_2$ | $b_2$ | $b_2$ |

  i.e., $P1$ or $P2$ (with 0 and 2 interchanged).

  – Subcase 1.2: $\{a_1, b_1\} = \{0,1\}$. Since we are not going to use $f_0$ any more without loss of generality we can assume that $a_1 = 0, b_1 = 1$. Observe that if $\{a_2, b_2\} = \{0,1\}$ then $f_1, f_2, f_3$ satisfy either $P1$ or $P2$. Let $\{a_2, b_2\} \neq \{0,1\}$ and put:

$$f_4 = \begin{cases} f_1 f_2 & \text{if } b_2 = 2 \text{ and } a_2 = 1 \\ f_3 f_2 & \text{if } b_2 = 2 \text{ and } a_2 = 0 \end{cases} \qquad f_5 = \begin{cases} f_1 f_2 & \text{if } a_2 = 2 \text{ and } b_2 = 1 \\ f_3 f_2 & \text{if } a_2 = 2 \text{ and } b_2 = 0 \end{cases}$$

  to get:

| $x$ | $f_1(x)$ | $f_3(x)$ | $f_2(x)$ | $f_4(x)$ | $f_5(x)$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | $a_2$ | 1 | 0 |
| 1 | 1 | 0 | $a_2$ | 1 | 0 |
| 2 | 0 | 0 | $b_2$ | 0 | 1 |

  So that $f_1, f_3, f_4$ satisfy $P2$ or $f_1, f_3, f_5$ satisfy $P1$.

- Case 2: For each $i \in \{0,1,2\}$ we have $i = b_i$.

  Without loss of generality we can assume that $a_0 = 1$, so that:

| $x$ | $f_0(x)$ | $f_1(x)$ | $f_2(x)$ |
|---|---|---|---|
| 0 | 0 | $a_1$ | $a_2$ |
| 1 | 1 | 1 | $a_2$ |
| 2 | 1 | $a_1$ | 2 |

If $a_2 = 0$ then replacing $f_2$ by $f_0 f_2$ we fall into case 1. If $a_2 = 1$ and $a_1 = 2$ then the polynomials $f_1, f_2, f_1 f_0$ put us into $P2$ situation. Finally, if $a_2 = 1$ and $a_1 = 0$ the polynomials $f_0, f_1, f_1 f_2$ again put us into $P2$ situation.

Now we know that $\mathbf{A}$ has 3 polynomials $f, g, h$ satisfying either $P1$ or $P2$.

Being in situation $P1$ we use the reduction of POSITIVE 1-IN-3-SAT presented in Lemma 3.1 to conclude that MPOLSAT$(\mathbf{A}) \in$ NP-complete. The reduction of POSITIVE 1-IN-3-SAT in situation $P2$ is only a bit harder. A formula $F = C_1 \wedge \cdots \wedge C_n$ is transformed into equations as follows. For each variable $x$ occurring in $F$ we need two variables $v_x$ and $x'$ and 3 equations:

$$x \rightsquigarrow \quad \begin{cases} f(v_x) = \top \\ g(v_x) = x \\ h(v_x) = x' \end{cases}$$

Next, for each clause $C_i$ we need a variable $v_i$ and then $C_i$ is transformed into 3 equations as follows:

$$C_i = x \vee y \vee z \quad \rightsquigarrow \quad \begin{cases} f(v_i) = x' \\ g(v_i) = y \\ h(v_i) = z \end{cases}$$

The equations for variable $x$ force $x'$ to simulate the negation of $x$. Indeed, because of the equation $f(v_x) = \top$ the variable $v_x$ cannot be valuated to 0. If $v_x = 1$ then $x = \top$ and $x' = \bot$, while for $v_x = 2$ we have $x = \bot$ and $x' = \top$. The equations for the clause $C_i$ work as in the proof of Lemma 3.1. Indeed, if $v_i = 0$ then $x', y, z$ take value $\bot$ and thus $x$ take value $\top$. If $v_i$ takes value 1 or 2 again exactly one of the variables $x, y, z$ takes value $\top$. □

**Lemma 4.3** *For a unary algebra* $\mathbf{A}$ *with 3 elements the problem* MPOLSAT$(\mathbf{A})$ *is in* P *(in fact it is* $O(n^2)$*) if* $width(\mathbf{P}(\mathbf{A})) \leqslant 2$.

**Proof:**

Given a system $S$ of equations we put $C(x) = A$ for all variables $x$ and apply our simplification algorithm $SA$ described in Subsection 4.1. As a result we get an equivalent instance $SA(I)$ satisfying:

1. All equations are of the form $f(x) = g(y)$, where $x \neq y$, $f(A) = g(A)$ and $|f(A)| = 2$.
2. $|C(x)| > 1$ for each $x$.

Now we present the algorithm solving such simplified instance. We do a reduction into 2-SAT which is known to be polynomial (in fact $O(n^2)$). For each pair $(v, f) \in V \times F^*$ we need a SAT variable $X_{v,f}$. Since $|f(A)| = 2$, there is exactly one $c_f \in A$ with $|f^{-1}(f(c_f))| = 1$ and then our intended interpretation can be described by:

$X_{v,f}$ is valuated by $T \Leftrightarrow v$ is valuated by $c_f$

| $v$ | $f(v)$ | $X_{v,f}$ |
|---|---|---|
| 0 | $a$ | $F$ |
| 1 | $a$ | $F$ |
| 2 | $b$ | $T$ |

$(\star)$

The equation $f(v) = g(u)$ is transformed into two 2-SAT clauses:

$$\text{or} \quad \begin{array}{ll} X_{v,f} \Leftrightarrow X_{u,g}, & \text{if } f(c_f) = g(c_g), \\ X_{v,f} \Leftrightarrow \neg X_{u,g}, & \text{otherwise.} \end{array} \qquad (\star\star)$$

For polynomials $f, g \in F^*$ and a variable $v$ with $C(v) = A$ we add a clause which code the interaction between SAT variables $X_{v,f}$ and $X_{v,g}$:

$$\text{or} \quad \begin{array}{ll} X_{v,f} \Rightarrow X_{v,g}, & \text{if } c_f = c_g, \\ X_{v,f} \Rightarrow \neg X_{v,g}, & \text{if } c_f \neq c_g. \end{array} \qquad (\star\star\star)$$

If there is a constraint on a variable $v$ with $C(v) \neq A$ then from (2) we know that $|C(v)| = 2$. For each such variable $v$ and $f, g \in F^*$ we add clauses:

- $\neg X_{v,f}$, whenever $c_f \notin C(v)$,
- $X_{v,f} \Leftrightarrow X_{v,g}$, if $c_f = c_g$ and $\{c_f, c_g\} \in C(v)$, $\qquad (\star\star\star')$
- $X_{v,f} \Leftrightarrow \neg X_{v,g}$, if $c_f \neq c_g$ and $\{c_f, c_g\} \in C(v)$.

It is easy to check that any solution of the system of equations $SA(I)$ can be transformed into an assignment for the variables $X_{v,f}$ satisfying all the clauses defined above. This can be simply done by $(\star)$.

Conversely, for a boolean valuation satisfying all clauses (defined above) we define a valuation of $v$ in $\mathbf{A}$ as follows:

- Case 1: $X_{v,f} = T$ for some $f \in F^*$. We evaluate $v$ by $c_f$. The clauses of the form $(\star\star\star)$ or $(\star\star\star')$ guarantee that it does not conflict with $X_{v,g}$ for $g \neq f$.
- Case 2: $X_{v,f} = F$ for all $f \in F^*$. Since $width(\mathbf{P}(\mathbf{A})) \leqslant 2$, we get $a \in A$ with $a \neq c_f$ for all $f \in F^*$. We evaluate $v$ by $a$. If there is a constraint on $v$ then from $(\star\star\star')$ we know that evaluating $y$ by one out of two elements of $C(v)$ will satisfy all $X_{v,f}$.

Now our condition $(\star\star)$ ensures us that this valuation is a solution for the instance $SA(I)$. □

Now we are ready to state the main theorem:

**Theorem 4.4** *For a unary algebra* $\mathbf{A}$ *with at most 3 elements, we have* MPOLSAT($\mathbf{A}$) *is in* P *(in fact it is* $O(n^2)$*) if* $width(\mathbf{P}(\mathbf{A})) \leqslant 2$ *holds, otherwise it is* NP*-complete.*

**Proof:** Note that $width(\mathbf{P}(\mathbf{A})) = 1$ for any two element algebra. On the other hand on the 2-element set all four unary operations are constants or permutations. Lemma 4.1 gives us that MPOLSAT($\mathbf{A}$) $\in$P. For $|A| = 3$ we directly apply Lemma 4.2 and 4.3. □

# 5 Other properties

We start with the following generalization of Lemma 4.1:

**Theorem 5.1** *Let $\mathbf{A} = (A, F)$ be a unary algebra in which there is $x_0 \in A$ such that for all $f \in F$:*

1. $f(x_0) = x_0$
2. $|f^{-1}(x)| \leqslant 1$ *for all* $x \neq x_0$.

*Then* MPOLSAT($\mathbf{A}$) $\in$P.

**Proof:** Given an instance of MPOLSAT($\mathbf{A}$) we apply our simplification algorithm to get $I = (V, S, C)$. Following the algorithm $SA$ one can check that the condition $x_0 \in C(v)$ is kept invariant for all variables in $S$. For example, if $SA$ sets $C(v) := C(v) \cap h^{-1}(c)$ and $c \neq x_0$ then $C(v)$ has at most one element and this variable is reduced by $SA$. If $c = x_0$ then obviously $x_0 \in h^{-1}(x_0)$.

Now we evaluate all variables in $S$ by $x_0$ and note that by (1) all equations in $S$ are satisfied. □

One consequence of Theorem 5.1 blocks a natural generalization of our characterizing Theorem 4.4:

**Proposition 5.2** *For each $n$ there exists $\mathbf{A}_n$ such that $width(\mathbf{P}(\mathbf{A}_n)) = n$ and* MPOLSAT($\mathbf{A}_n$) *is polynomial.*

**Proof:** Put $\mathbf{A}_n = (\{0, \ldots, n\}, f_1, \ldots, f_n)$, where:

| $x$ | $f_1(x)$ | $f_2(x)$ | $\ldots$ | $f_n(x)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | $\ldots$ | 0 |
| 1 | 1 | 0 | $\ldots$ | 0 |
| 2 | 0 | 1 | $\ldots$ | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| $n$ | 0 | 0 | $\ldots$ | 1 |

The algebra $\mathbf{A}_n$ satisfies (1) and (2) of Theorem 5.1 and therefore MPOLSAT($\mathbf{A}_n$) $\in$P. On the other hand one easily check that $width(\mathbf{P}(\mathbf{A}_n)) = n$. □

Note however, that if we allow constraints, e.g of the form $C(x) = \{1, 2, 3\}$ then the problem will be NP-complete.

**Theorem 5.3** *If* P $\neq$ NP *then the class* $\{\mathbf{A} : \text{MPOLSAT}(\mathbf{A}) \in \text{P}\}$ *of unary algebras is not closed under homomorphic images.*

**Proof:** Note that the 7 element algebra $\mathbf{H}$ given by:

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ | $p(x)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 2 |
| 6 | 0 | 0 | 1 | 3 |

satisfies (1) and (2) of Theorem 5.1, so that MPOLSAT($\mathbf{H}$) $\in$P.

On the other hand congruence $\Theta$ with the only nontrivial block being $\{1, 2, 3\}$ gives rise to the quotient algebra $\mathbf{H}' = \mathbf{H}/\Theta$:

| $x$ | $f(x)$ | $g(x)$ | $h(x)$ | $p(x)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 |

Following the argument in the proof of Lemma 3.1 with additional twist given by equations of the form $p(v) = 1$ for each variable $v$, we conclude that MPOLSAT($\mathbf{H}'$) $\in$NP-complete. □

Theorem 5.3 shows that MPOLSAT cannot be reduced to smaller structures as it can happen that after such reduction it appears to be harder. This situation stays in a great contrast to CSP, where actually such reduction is often done, see [1].

# 6 Conclusion

From Proposition 5.2 we know that there is no uniform bound for width of an preorder that will guarantee that MPOLSAT is in P. Although we can still have hope to find connections between width (or other properties of preorder) and complexity. For example a sublinear function of $|A|$ can possibly limit width of preorders for algebras with tractable systems of equations.

# References

[1] A. Bulatov. A dichotomy theorem for constraints on a three-element set. *43rd IEEE Symposium on Foundations of Computer Science (FOCS'02), Vancouver, Canada*, 2002.

[2] S. N. Burris and H. P. Sankappanavar. *A course in universal algebra*. Graduate Texts in Mathematics. Springer-Verlag, 1981.

[3] T. Feder, F. Madelaine, and I. A. Stewart. Dichotomies for classes of homomorphism problems involving unary functions. *Theor. Comput. Sci.*, 314:1–43, 2004.

[4] D. Hobby and R. McKenzie. *The Structure of Finite Algebras*. Contemporary Mathematics v. 76, American Mathematical Society, 1988.

[5] B. Larose and L. Zádori. Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. *Internat. J. Algebra Comput.*, to appear.