Algebra Univers. 75 (2016) 331–350
DOI 10.1007/s00012-016-0382-3
Published online March 21, 2016

Algebra Universalis

Loosely-abelian algebras

Jacek Krzaczkowski

Abstract. Abelianity has two different meanings in universal algebra. On the one hand, the term "abelian" is used interchangeably with "commutative" whilst on the other, an algebra is said to be abelian if for every term $t(x, \overline{y})$ and for all elements a, b, $\overline{c}$, $\overline{d}$ we have the following implication: $t(a, \overline{c}) = t(a, \overline{d}) \Rightarrow t(b, \overline{c}) = t(b, \overline{d})$. These two definitions are equivalent for groups but not generally. We will introduce the class of loosely-abelian algebras which for finite algebras is a generalization of both kinds of abelianity mentioned above. We will prove some basic properties of loosely-abelian algebras and using the introduced concept, we will characterize the subreducts of finite semilattices. Furthermore, we will present an algorithm which solves equations over loosely-abelian algebras.

1. Introduction

Groups are one of the best known and studied algebraic structures. One of the reasons is that group theory has many applications in numerous areas of mathematics and other sciences, e.g., public key cryptography, algebraic geometry, combinatorics, physics and chemistry. One of the most important classes of groups is commutative groups, which generalize the arithmetic of integer addition. Commutative groups are also called abelian groups.

There are two different generalizations of the commutative group concept. The best known and intuitive generalization of commutative groups are commutative semigroups, sometimes called abelian semigroups. The second generalization are abelian algebras, defined by a term condition, which are a generalization transferred from the concept of commutative groups into general algebra. Abelianity in such a sense plays a crucial role in tame congruence theory [6] and commutator theory [2]. A group is commutative iff it is abelian (i.e., it fulfills the term condition). If we consider generalizations of abelian groups, the weakness of abelianity, defined by the term condition, is the fact that commutativity and abelianity are not the same for semigroups. There are commutative semigroups which are not abelian and abelian semigroups which are not commutative (for characterization of abelian semigroups see [14]). We would like to have a language independent generalization of abelian groups that is also a generalization of commutative semigroups and abelian algebras.

Presented by K. Kearnes.
Received March 17, 2014; accepted in final form February 19, 2015.
2010 *Mathematics Subject Classification*: Primary: 08A62; Secondary: 08A50, 20K99, 06A12.
Key words and phrases: commutativity, abelianity, finite algebra, solving equations, semilattice.

In this paper, we will introduce the concepts of k-loosely-abelian and loosely-abelian algebras. Speaking informally, an algebra $\mathbf{A}$ is k-loosely-abelian iff for all term operations t of $\mathbf{A}$ we can divide the arguments of t into at most k classes such that the arguments contained in any one class behave identically. We say that the algebra is loosely-abelian if it is k-loosely-abelian for some k. Finite loosely-abelian algebras are the common generalization of both finite commutative semigroups and finite abelian algebras. Furthermore, for finite monoids, we have that a monoid is commutative iff it is loosely-abelian. We will also prove some other properties of loosely-abelian algebras.

An interesting question is which algebras are loosely-abelian. In the final three sections, we try to characterize 1-loosely-abelian algebras. It turns out that under one natural additional assumption, 1-loosely-abelian finite algebras and subreducts of finite semilattices are essentially the same. An open problem is to characterize, in a similar way, k-loosely-abelian algebras for $k > 1$ and loosely-abelian algebras in general.

Loosely-abelian algebras are interesting from the algebraic point of view but their idea is derived from studies into the computational complexity of solving equations over finite algebras. An attempt to generalize the algorithms for solving equations over abelian algebras and semiaffine algebras led to the definition of the property, which later became the definition of loosely-abelian algebras. We will show a polynomial time algorithm that solves equations over finite loosely-abelian algebras. The technique used in this algorithm can be used to solve other problems connected with equations over loosely-abelian algebras.

This paper is organized as follows. Section 1 presents an introduction to the subject. Section 2 contains some algebraic definitions. In Section 3, we define the problem of solving equations and present some known results. In Section 4, we introduce loosely-abelian algebras and prove some of their basic properties. In Section 5, we show a polynomial time algorithm that solves equations over finite loosely-abelian algebras. Sections 6 and 7 contain lemmas which we need in Section 8. Finally, in Section 8, we characterize a large class of finite 1-loosely-abelian algebras and discuss what other finite 1-loosely-abelian algebras look like.

2. Definitions and notations

We use standard algebraic definitions, which the reader can find in, e.g., [1]. We only consider algebras with at least two elements. Note that the set of fundamental operations of an algebra may not be finite.

If t is a term (polynomial), in which only the (distinct) variables from $\{x_1, \ldots, x_n\}$ appear, then $t^{\mathbf{A}}(x_1, \ldots, x_n)$ describes the corresponding n-ary term (polynomial) operation. We denote the set of term (polynomial) operations of $\mathbf{A} = (A, F)$ by $\mathrm{Clo}(\mathbf{A})$ ($\mathrm{Pol}(\mathbf{A})$). Note that $\mathrm{Clo}(\mathbf{A})$ and $\mathrm{Pol}(\mathbf{A})$ are

the *clones of operations* on A, i.e., sets of operations on A, closed under composition, and containing the projection operations $\pi_i^n(x_1, \ldots, x_n) = x_i$. We denote the set of variables occurring in the term (polynomial) t by Var(t). We define $\mathrm{Term}(\mathbf{A}) \subseteq \mathrm{Clo}(\mathbf{A})$ as the set of term operations $t^{\mathbf{A}}(x_1, \ldots, x_n)$ such that $\mathrm{Var(t)} = \{x_1, \ldots, x_n\}$ (the arity of $t^{\mathbf{A}}$ is equal to the number of variables occurring in term t). In a similar way we define $\mathrm{Polyn}(\mathbf{A}) \subseteq \mathrm{Pol}(\mathbf{A})$ as the set of polynomial operations of arity equal to the number of variables occurring in the corresponding polynomial. We write $\mathrm{Term_k}(\mathbf{A})$ ($\mathrm{Polyn_k}(\mathbf{A})$) to denote the set of k-ary operations from $\mathrm{Term}(\mathbf{A})$ ($\mathrm{Polyn}(\mathbf{A})$).

We say that algebra $\mathbf{B}$ is a *reduct* of algebra $\mathbf{A}$ iff $\mathbf{A}$ and $\mathbf{B}$ have the same universe and $\mathrm{Clo}(\mathbf{B}) \subseteq \mathrm{Clo}(\mathbf{A})$. Subalgebras of reducts of algebra $\mathbf{A}$ are called *subreducts* of $\mathbf{A}$.

We say that an equation between terms over $\mathbf{A}$ in the form

$$t_1(x_1, \ldots, x_n) = t_2(y_1, \ldots, y_m),$$

where $\{x_1, \ldots, x_n\} \cup \{y_1, \ldots, y_m\} = X$, is satisfiable iff there exists a function $s\colon X \to A$, such that $t_1^{\mathbf{A}}(s(x_1), \ldots, s(x_n)) = t_2^{\mathbf{A}}(s(y_1), \ldots, s(y_m))$. Such a function s is called a *solution*. Note that the intersection of sets $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$ may not be empty. We may similarly define the solution for equations between polynomials.

We will now define a few important classes of algebras which we will generalize in the next section. Firstly, we will define the abelian algebras.

Definition 2.1. An algebra $\mathbf{A} = (A, F)$ is called abelian if the following holds: for every $n+1$-ary term f of $\mathbf{A}$ and every $u, v, x_1, \ldots, x_n, y_1, \ldots, y_n \in A$ we have that:

$$f(u, x_1, \ldots, x_n) = f(u, y_1, \ldots, y_n) \Leftrightarrow f(v, x_1, \ldots, x_n) = f(v, y_1, \ldots, y_n).$$

Note that a group is abelian iff it is commutative. For this purpose abelianity is considered a generalization of the concept of commutative groups.

Affine algebras are another generalization of abelian groups. They can be defined in the following way.

Definition 2.2. An algebra $\mathbf{A} = (A, F)$ is called an *affine algebra* iff there is an abelian group $\mathbf{G} = (A, +)$ such that

- the operation $m(x, y, z) = x - y + z$ is a term operation of $\mathbf{A}$,
- every polynomial of $\mathbf{A}$ may be expressed in the form: $\sum_{i=1}^{n} e_i(x_i) + a$, where e_i is an endomorphism of $\mathbf{G}$ and $a \in A$.

Note that every affine algebra is abelian.

An obvious generalization of abelian groups are commutative semigroups. Unfortunately, not all commutative semigroups are abelian and not all abelian semigroups are commutative.

Example 2.3. The semilattice $(2, \wedge)$ is a commutative semigroup, but it is not abelian. On the other hand, every set with a two-argument projection as an operation forms an abelian semigroup which is not commutative.

Semiaffine algebras are a common generalization of affine algebras and commutative semigroups.

Definition 2.4. An algebra $\mathbf{A} = (A, F)$ is called a *semiaffine algebra* iff there is a commutative semigroup $\mathbf{S} = (A, +)$, such that any basic operation of $\mathbf{A}$ may be expressed in one of the following two forms:

$$\sum_{i=1}^{n} e_i(x_i) + a \qquad \text{or} \qquad \sum_{i=1}^{n} e_i(x_i),$$

where $\{e_1, \ldots, e_n\}$ are endomorphisms of $\mathbf{S}$ and $a \in A$. Any operations which can be expressed in one of the above two forms are called *semiaffine operations*.

In Section 4, we will introduce the concept of loosely-abelian algebras which, in the case of finite algebras, seems to be an interesting generalization of abelian groups containing both semiaffine and abelian algebras.

3. Solving equations over finite algebras

The most commonly considered problem connected with solving equations over finite algebras is the following.

Definition 3.1. For a finite algebra $\mathbf{A}$, the *polynomial satisfiability problem* (POL-SAT($\mathbf{A}$)) is the decision problem with

Instance: A pair of polynomials (s, t) with the tables of the fundamental operations of $\mathbf{A}$ corresponding to all function symbols occurring in s and t.

Question: Does there exist a substitution of variables from A, such that the values of functions $s^{\mathbf{A}}$ and $t^{\mathbf{A}}$ are the same?

In a similar way, we can define the systems of equations of the polynomial satisfiability problem (SysPol($\mathbb{A}$)) and the satisfiability of the term equations problem (TERM-SAT($\mathbf{A}$)).

Such definitions, originally stated in [4], are different from those use by other authors. The difference is that our definitions allow algebras with an infinite number of basic operations, and for this reason the tables of operations used in the equations are part of the instance (in this way we guarantee that the problems are contained in NP). For algebras with a finite number of basic operations these definitions are equivalent to the former ones (in the sense that there are polynomial-time reductions from the new problem to the corresponding old problem and from the former problem to the new one).

There are many results for the computational complexity for solving equations or systems of equations over well-known structures, e.g., groups [3], rings [7], monoids [11] and lattices [13].

In [12], it is shown that the problem SysPol($\mathbf{A}$) for any algebra $\mathbf{A}$ is polynomially equivalent to $CSP(\Gamma)$ for some Γ (Constraint Satisfaction Problem

for a relational structure Γ). One of the important results of an algebraic approach to *CSP* is the proof that the computational complexity of $CSP(\Gamma)$ for the finite relational structure Γ with a finite number of relations depends only on the relational clone of Γ. Therefore, it seems natural to question whether problems connected with solving equations over fixed algebra $\mathbf{A}$ depend only on the clone of the term operation of $\mathbf{A}$. As a result, from [12], we have that the computational complexity of SysPol($\mathbf{A}$) for $\mathbf{A}$ of a finite type depends only on Clo($\mathbf{A}$). On the other hand, we have the following example.

Example 3.2. Consider the smallest non-nilpotent, solvable group $\mathbf{S_3} = (S_3, \circ)$. Let $s(x, y, z, w) = x \circ [[[x, y], z], w]^{-1}$, where $[x, y] = x^{-1} \circ y^{-1} \circ x \circ y$. Obviously, $\mathrm{Clo}(\mathrm{S}_3, \circ)) = \mathrm{Clo}(\mathrm{S}_3, \circ, \mathrm{s})$.

POL-SAT$(\mathrm{S}_3, \circ)$ is in P ([8]) but POL-SAT$(\mathrm{S}_3, \mathrm{s}, \circ)$ is NP-complete (P.M. Idziak's result published in [5]).

Szabó and Horváth in [10] showed that group A_4 is another example of an algebra for which the complexity of POL-SAT does not only depend on the term clone. Moreover, they proved that for every non-nilpotent solvable finite group $(G, \circ)$, we may choose operations $f_1, \ldots, f_k \in \mathrm{Clo}(\mathrm{G}, \circ)$ such that POL-SAT$(\mathrm{G}, \circ, \mathrm{f}_1, \ldots, \mathrm{f}_\mathrm{k})$ is NP-complete [9].

However, there are some big classes of algebras, e.g two-element algebras [4] and preprimal algebras [5], for which the computational complexity of TERM-SAT and POL-SAT depends only on Clo($\mathbf{A}$). In Section 5, we will prove that loosely-abelian algebras are another such class of algebras.

4. Loosely-abelian algebras

In this section, we introduce the concept of loosely-abelian algebras, which is a generalization of concepts of abelianity and commutativity in finite algebras.

To define loosely-abelian algebras, we will need the following definition.

Definition 4.1. For a k-ary operation p on set A, we define a relation $\rho_p \subset \{1, \ldots, k\}^2$ in the following way:

$(i, j) \in \rho_p$ iff

$$\forall_{a_1,\ldots,a_k \in A}\ p(a_1, \ldots, a_i, \ldots, a_j, \ldots, a_k) = p(a_1, \ldots, a_j, \ldots, a_i, \ldots, a_k).$$

Note that for any operation p, the relation ρ_p is an equivalence relation on indexes of arguments of p. For a term $t(x_1, \ldots, x_k)$ over algebra $\mathbf{A}$, we define $\rho_{\mathrm{var(t)}} = \{(x_i, x_j) \colon (i, j) \in \rho_{t^A}\}$. Obviously, $\rho_{\mathrm{var(t)}}$ is an equivalence relation on $\{x_1, \ldots, x_k\}$.

We can now define loosely-abelian algebras.

Definition 4.2. Let $\mathbf{A}$ be an algebra. We say that $\mathbf{A}$ is *k-loosely-abelian* if for every operation $f \in \mathrm{Term}(\mathbf{A})$, the relation ρ_f has at most k equivalence classes. We say that $\mathbf{A}$ is *loosely-abelian* if it is k-loosely-abelian for some k.

Note that if in the above definition we used polynomial operations of an algebra instead of term operations, we would obtain the equivalent definition of loosely-abelian algebras.

Loosely-abelian algebras have some useful properties.

Fact 4.3. Let $\mathbf{A}$ and $\mathbf{B}$ be algebras of the same type. If $\mathbf{A}$ is k-loosely-abelian and $\mathbf{B}$ is l-loosely-abelian, then

- $\mathbf{A}$ is (k+1)-loosely-abelian,
- homomorphic images of $\mathbf{A}$ are k-loosely-abelian,
- subalgebras of $\mathbf{A}$ are k-loosely-abelian,
- $\mathbf{A} \times \mathbf{B}$ is $k \cdot l$-loosely-abelian,
- $\mathbf{A} \times \cdots \times \mathbf{A}$ is k-loosely-abelian,
- subreducts of $\mathbf{A}$ are k-loosely-abelian.

The following corollary is the straightforward consequence of Fact 4.3.

Corollary 4.4. *Let $\mathbf{A}$ be a k-loosely-abelian algebra. Then every algebra in the variety generated by $\mathbf{A}$ is k-loosely-abelian.*

It turns out that in the finite case, loosely-abelian algebras may be considered as a generalization of both commutativity and abelianity concepts.

Theorem 4.5. *Let $\mathbf{A}$ be a semiaffine algebra over a finite semigroup. Then $\mathbf{A}$ is loosely-abelian.*

Proof. From the assumptions of the current lemma and the definition of semiaffine algebras, we have that there exists a finite commutative semigroup $\mathbf{S}$ such that every term operation p of $\mathbf{A}$ may be expressed in one of the following two forms:

$$\sum_{i=1}^{n} e_i(x_i) + a \qquad \text{or} \qquad \sum_{i=1}^{n} e_i(x_i),$$

where every variable occurs at most once, and $\{e_1, \ldots, e_n\}$ are the endomorphisms of $\mathbf{S}$.

There are only a finite number of endomorphisms of the finite semigroup $\mathbf{S}$. Let k be the number of endomorphisms of $\mathbf{S}$. It is clear that if $e_i = e_j$, then $(i, j) \in \rho_p$ and consequently, we have that the number of equivalence classes of the relation ρ_p is bounded by k. Note that k depends on algebra $\mathbf{A}$ only. This implies that $\mathbf{A}$ is k-loosely-abelian and, as a result, it is loosely-abelian. □

The corollary for this theorem is the fact that every commutative semigroup is loosely-abelian.

Theorem 4.6. *Let $\mathbf{A} = (A, F)$ be a finite abelian algebra. Then $\mathbf{A}$ is loosely-abelian.*

Proof. Let $|A| = n$, $a, a_2, \ldots, a_k, b_2, \ldots, b_k \in A$ and let $p \in \mathrm{Term}(\mathbf{A})$ be a k-ary operation. From the definition of abelian algebras, we have that the unary operations $t_{a_2 \ldots, a_k}(x) = p(x, a_2, \ldots, a_k)$ and $t_{b_2 \ldots, b_k}(x) = p(x, b_2, \ldots, b_k)$ are

the same iff $t_{a_2\ldots,a_k}(a) = t_{b_2\ldots,b_k}(a)$. This means that $t_{a_2\ldots,a_k}(x)$ is determined by two values: $p(a, a_2, \ldots, a_k)$ and x. Hence, we obtain that $p(x_1, \ldots, x_k) = f_1(p(a, x_2, \ldots, x_k), x_1)$ for some operation $f_1 \colon A^2 \to A$.

We can prove that $p(x_1, \ldots, x_k) = f_1(f_2(p(a, a, x_3, \ldots, x_k), x_2), x_1)$ in a similar way, and finally that

$$p(x_1, \ldots, x_k) = f_1(f_2(\ldots f_k(p(a, \ldots, a), x_k) \ldots, x_2), x_1). \tag{4.1}$$

Note that in the construction of expression (4.1), the order of the arguments under consideration is not important. Furthermore, the operations $\{f_i\}_{i=0}^{k-1}$ obtained as a result of this construction do not depend on the order in which the arguments of p were considered. Therefore, for any permutation $\sigma \colon k \to k$, we have that:

$$p(x_1, \ldots, x_k) = f_{\sigma(1)}(f_{\sigma(2)}(\ldots f_{\sigma(k)}(p(a, \ldots, a), x_{\sigma(k)}) \ldots, x_{\sigma(2)}), x_{\sigma(1)}).$$

Note that if $f_i \equiv f_j$, then $(i, j) \in \rho_p$. Moreover, there are at most $m = n^{n^2}$ different binary operations on set A. Thus, the relation ρ_p has at most m equivalence classes. Notice that m does not depend on operation p. From the above facts, we have that $\mathbf{A}$ is m-loosely-abelian and consequently is loosely-abelian. □

The next theorem gives us characterizations of loosely-abelian semigroups and monoids.

Theorem 4.7. *Let $\mathbf{S} = (S, \cdot)$ be a finite semigroup; then the following hold:*

(1) *$\mathbf{S}$ is loosely-abelian iff there exist k, m such that*

$$x_1 \cdot \ldots \cdot x_k \cdot x_{k+1} \cdot x_{k+2} \cdot x_{k+3} \cdot \ldots \cdot x_{k+m+2} = x_1 \cdot \ldots \cdot x_k \cdot x_{k+2} \cdot x_{k+1} \cdot x_{k+3} \cdot \ldots \cdot x_{k+m+2}. \tag{4.2}$$

(2) *If $\mathbf{S}$ is a monoid, then $\mathbf{S}$ is loosely-abelian iff it is commutative.*

Proof. We will prove the first point and the second one will turn out to be a simple consequence of the first one.

If expression 4.2 holds, then we can express any term operation $t(x_1, \ldots, x_w)$ of $\mathbf{S}$ with $(w > k + m)$ in the following form:

$$t(x_1, \ldots, x_w) = x'_1 \cdot \ldots \cdot x'_k \cdot (x_{1,1})^1 \cdot \ldots \cdot (x_{1,n_1})^1 \cdot (x_{2,1})^2 \cdot \ldots \cdot (x_{2,n_2})^2 \cdot \ldots \cdot (x_{|A|,1})^{|A|} \cdot \ldots \cdot (x_{|A|,n_{|A|}})^{|A|} \cdot x''_1 \cdot \ldots \cdot x''_m,$$

where $x'_i, x_{o,p}, x''_j \in \{x_1, \ldots, x_w\}$ and x_{i_1,j_1}, x_{i_2,j_2} are different if $i_1 \neq i_2$ or $j_1 \neq j_2$. Hence, $\mathbf{S}$ is $(k + m + |A|)$-loosely-abelian.

Now let us assume that semigroup $\mathbf{S}$ is l-loosely-abelian. Then for term $t(x_1, \ldots, x_{l+1}) = x_1 \cdot x_2 \cdot \ldots \cdot x_{l+1}$, there exist i and j, such that $(x_i, x_j) \in \rho_{\mathrm{var(t)}}$. This implies that

$$x_1 \cdot \ldots \cdot x_{i-1} \cdot x_i \cdot x_{i+1} \cdot \ldots \cdot x_{j-1} \cdot x_j \cdot x_{j+1} \cdot \ldots \cdot x_{l+1} = x_1 \cdot \ldots \cdot x_{i-1} \cdot x_j \cdot x_{i+1} \cdot \ldots \cdot x_{j-1} \cdot x_i \cdot x_{j+1} \cdot \ldots \cdot x_{l+1}.$$

If $i+1=j$, then expression 4.2 is true for $k=i-1$ and $m=l-j+1$. Assume that $j-i>1$. Set $X_{a,b}=x_a\cdot x_{a+1}\cdot\ldots\cdot x_b$. We now have

$$\begin{aligned}&X_{1,i-1}\cdot x_i\cdot X_{i+1,j-2}\cdot x_{j-1}\cdot x_j\cdot x_{j+1}\cdot X_{j+2,l+2}\\ &= X_{1,i-1}\cdot x_j\cdot X_{i+1,j-2}\cdot x_{j-1}\cdot x_i\cdot x_{j+1}\cdot X_{j+2,l+2},\end{aligned}$$

and from the associativity of "$\cdot$":

$$\begin{aligned}&X_{1,i-1}\cdot x_j\cdot X_{i+1,j-2}\cdot x_{j-1}\cdot x_i\cdot x_{j+1}\cdot X_{j+2,l+2}\\ &= X_{1,i-1}\cdot x_j\cdot X_{i+1,j-2}\cdot (x_{j-1}\cdot x_i)\cdot x_{j+1}\cdot X_{j+2,l+2}\\ &= X_{1,i-1}\cdot x_{j+1}\cdot X_{i+1,j-2}\cdot (x_{j-1}\cdot x_i)\cdot x_j\cdot X_{j+2,l+2}.\end{aligned}$$

Finally,

$$\begin{aligned}&X_{1,i-1}\cdot x_{j+1}\cdot X_{i+1,j-2}\cdot (x_{j-1}\cdot x_i)\cdot x_j\cdot X_{j+2,l+2}\\ &= X_{1,i-1}\cdot x_{j+1}\cdot X_{i+1,j-2}\cdot x_{j-1}\cdot x_i\cdot x_j\cdot X_{j+2,l+2}\\ &= X_{1,i-1}\cdot x_i\cdot X_{i+1,j-2}\cdot x_{j-1}\cdot x_{j+1}\cdot x_j\cdot X_{j+2,l+2}.\end{aligned}$$

This implies that expression 4.2 holds for $k=j-1$ and $m=l-j+1$.

We will now prove the second point. If $\mathbf{S}$ is commutative, then from Theorem 4.5, $\mathbf{S}$ is loosely-abelian. On the other hand, if $\mathbf{S}$ is loosely-abelian, then expression 4.2 holds and if we substitute in it the variables $x_1,\ldots,x_k$ and $x_{k+3},\ldots,x_{k+m+2}$ with a neutral element of $\mathbf{S}$, we obtain a new expression $x_{k+1}\cdot x_{k+2}=x_{k+2}\cdot x_{k+1}$. Thus, $\mathbf{S}$ is commutative. □

Lattices are one of the examples of algebras that are not loosely-abelian.

Example 4.8. Let $\mathbf{A}=(A,\vee,\wedge)$ be a lattice. Then $\mathbf{A}$ is not loosely-abelian.

Proof. Let $0,1\in A$ be such that $0\leq 1$ and $0\neq 1$. Consider a family $\{f_i\}_{i=1}^{\infty}$ of term operations over $\mathbf{A}$ defined by

$$f_i(x_1^1,x_2^1,x_1^2,x_2^2,\ldots,x_1^i,x_2^i)=(x_1^1\wedge x_2^1)\vee(x_1^2\wedge x_2^2)\vee\cdots\vee(x_1^i\wedge x_2^i),$$

and the family of assignments $s_{k,l}^i\colon\{x_1^1,x_2^1,x_1^2,x_2^2,\ldots,x_1^i,x_2^i\}\to\{0,1\}$ such that

$$s_{k,l}^i(x_h^j)=\begin{cases}1 & k=j\wedge h=1,\\ 1 & l=j\wedge h=2,\\ 0 & \text{otherwise.}\end{cases}$$

It is easy to see that $f_i(s_{k,l}^i(x_1^1),s_{k,l}^i(x_2^1),\ldots,s_{k,l}^i(x_1^i),s_{k,l}^i(x_2^i))$ is equal to 1 iff $k=l$ and equal to 0 otherwise. This implies that $(x_h^j,x_o^p)\in\rho_{ver(f_i)}$ iff $j=p$. Hence, for every i, relation $\rho_{\mathrm{var(f_i)}}$ has i equivalence classes and consequently $\mathbf{A}$ is not loosely-abelian. □

The following corollary is an obvious consequence of Example 4.8 and the fact that homomorphic images and subreducts of loosely-abelian algebras are loosely-abelian.

Corollary 4.9. *Let* $\mathbf{A}$ *be a finite loosely-abelian algebra. Then* $\mathbf{A}$ *omits types* 3 *and* 4*.*

In the above corollary, we use the terms "type 3" and "type 4" as it is used in Tame Congruence Theory [6].

5. TERM-SAT for loosely-abelian algebras

As mentioned in the introduction, the idea of loosely-abelian algebras is derived from the studies on computational complexity of solving equations over finite algebras. The definition of loosely-abelian algebras is the result of looking for a common characteristic for algorithms solving equations over abelian and semiaffine algebras. In this section, we show a polynomial time algorithm solving TERM-SAT for loosely-abelian algebras that is a generalization of both algorithms mentioned above. To show that the presented algorithm is computable in polynomial time we will need two lemmas.

Lemma 5.1. *Let* p *be an operation on a finite set* A*. There exists a polynomial time algorithm that computes equivalence classes of* ρ_p *for a given table of operation* p*.*

Proof. For a given table of a k-ary operation p, we may compute equivalence classes of ρ_p considering each argument of p and assigning equivalence classes of ρ_p to each of them. If, when considering the i-th argument of p, it turns out not to have an assigned equivalence class of ρ_p, we create a new equivalence class containing i and compare the behavior of the i-th argument with the behavior of the remaining arguments without an equivalence class. If the j-th argument behaves in the same way as the i-th argument, we add j to class $[i]_{\rho_p}$. Note that to compare the behavior of the i-th and j-th arguments, it is sufficient to compare the values of the operation p for the pairs of tuples from the set

$$\{((a_1,\dots,a_i,\dots,a_j,\dots,a_k),(a_1,\dots,a_j,\dots,a_i,\dots,a_k)) \in A^k \times A^k\},$$

which can be done in polynomial time in the size of the table of p. Finally, note that to compute all equivalence classes of ρ, it is sufficient to compare the behavior of $O(k^2)$ pairs of arguments. This observation completes the proof. □

The proof of the following lemma is obvious and we will omit it.

Lemma 5.2. *Let* $\mathbf{A} = (A, F)$ *be a finite algebra. There exists a polynomial time algorithm that for a given polynomial* $p(x_1,\dots,x_k)$ *over* $\mathbf{A}$*,* $a_1,\dots,a_k \in A$*, and tables of fundamental operations of* $\mathbf{A}$ *occurring in* p*, computes the value of* $p^{\mathbf{A}}(a_1,\dots,a_k)$ *in polynomial time.*

Next we prove the main theorem in this section. To do it, we need two definitions. Let $\{x_1,\dots,x_k\}$ be a set of variables, A an ordered set, and

$R \subseteq \{x_1, \dots, x_k\}^2$ an equivalence relation. Then function $s\colon \{x_1, \dots, x_k\} \to A$ is called *monotonic on the classes of* R iff for all pairs $(x_i, x_j) \in R$, if $i \leq j$, then $s(x_i) \leq s(x_j)$. For a given $s\colon \{x_1, \dots, x_k\} \to A$, we denote by s_{x_i,x_j} the function $s_{x_i,x_j}\colon \{x_1, \dots, x_k\} \to A$ such that $s_{x_i,x_j}(x_i) = s(x_j)$, $s_{x_i,x_j}(x_j) = s(x_i)$, and $s_{x_i,x_j}(x) = s(x)$ for $x \notin \{x_i, x_j\}$.

Theorem 5.3. *Let* $\mathbf{A}$ *be a finite loosely-abelian algebra. There is a polynomial time algorithm solving the problem* TERM-SAT($\mathbf{A}$).

Proof. Let $\mathbf{A} = (A, F)$ be an m-loosely-abelian algebra. Assume that the given equation is in the form

$$t_1(x_1, \dots, x_k) = t_2(y_1, \dots, y_l). \tag{5.1}$$

Let $X = \{x_1, \dots, x_k\}$, $Y = \{y_1, \dots, y_l\}$, and $Z = X \cup Y$. Assume that $Z = \{z_1, \dots, z_b\}$ and fix a linear order $\leq_A$ on A.

Claim. *There are polynomial time algorithms computing* ρ_{t_1} *and* ρ_{t_2}.

From Lemma 5.1, we can compute the relation ρ_p for every fundamental operation p of $\mathbf{A}$ occurring in t_1 and t_2. Obviously, if we have ρ_p, it is easy to compute $\rho_{\mathrm{var(t)}}$ for the term $t(x_1, \dots, x_k) = p(x_1, \dots, x_k)$. We will prove that for the given terms $w(x_1, \dots, x_k)$ and $h(y, x_1, \dots, x_k)$ and the relations $\rho_{\mathrm{var(w)}}$, $\rho_{\mathrm{var(h)}}$, we can compute the relation $\rho_{\mathrm{var(g)}}$ for

$$g(x_1, \dots, x_k) = h(w(x_1, \dots, x_k), x_1, \dots, x_k)$$

in polynomial time. Let ρ denote $\rho_{\mathrm{var(h)}} \cap \{x_1, \dots, x_k\}^2$. Notice that $\rho \cap \rho_{\mathrm{var(w)}} \subset \rho_{\mathrm{var(g)}}$, and to obtain $\rho_{\mathrm{var(g)}}$ it is sufficient to compare only the behavior of the representatives of the equivalence classes of $\rho \cap \rho_{\mathrm{var(w)}}$. Note that $\rho \cap \rho_{\mathrm{var(w)}}$ has at most m^2 equivalence classes. Thus, we only need $O(m \cdot m^2) = O(1)$ comparisons of the variables behavior to obtain $\rho_{\mathrm{var(g)}}$ (we compute one equivalence class of $\rho_{\mathrm{var(g)}}$ using $O(m^2)$ comparisons of behavior of the representative of one class of $\rho \cap \rho_{\mathrm{var(w)}}$ with the behavior of other class representatives, and $\rho_{\mathrm{var(g)}}$ has at most m equivalence classes). Notice that to determine if two variables x_i and x_j are in the relation $\rho_{\mathrm{var(g)}}$, it is enough to compare values of $g(s(x_1), \dots, s(x_k))$ and $g(s_{x_i,x_j}(x_1), \dots, s_{x_i,x_j}(x_k))$ for assignments s monotonic on the classes of $\rho \cap \rho_{\mathrm{var(w)}}$. There are at most $O(((k + |A| - 1)^{|A|-1})^{m^2}) = O((k^{|A|-1})^{m^2}) = O(k^{(|A|-1)m^2})$ such assignments of variables $x_1, \dots, x_k$, and there are polynomially many in the size of g. Hence, we can compute $\rho_{\mathrm{var(g)}}$ in polynomial time.

Notice that t_1 and t_2 consist of linearly many (in the size of polynomials) function symbols. Therefore, we can compute $\rho_{\mathrm{var(t_1)}}$ and $\rho_{\mathrm{var(t_2)}}$ by a linear number of iterations of the algorithm described in the previous paragraph. Hence, $\rho_{\mathrm{var(t_1)}}$ and $\rho_{\mathrm{var(t_2)}}$ can be computed in polynomial time in the size of the input. This completes the proof of the claim.

We can now define the relation

$$\varrho = (\rho_{\mathrm{var(t_1)}} \cap (X \setminus Y)^2) \cup (\rho_{\mathrm{var(t_2)}} \cap (Y \setminus X)^2) \cup (\rho_{\mathrm{var(t_1)}} \cap \rho_{\mathrm{var(t_2)}}).$$

Notice that ϱ is an equivalence relation on the set Z with at most $m \cdot m + 2 \cdot m$ equivalence classes. Moreover, observe that if $(z_i, z_j) \in \varrho$ and

$$(a_1, \ldots, a_i, \ldots, a_j, \ldots, a_b) \in A^b$$

is a solution of Equation (5.1), then $(a_1, \ldots, a_j, \ldots, a_i, \ldots, a_b) \in A^b$ is another solution of this equation.

Therefore, to determine if Equation (5.1) has a solution, it is enough to check if any assignments monotonic on classes of ϱ is a solution. This implies that it is sufficient to check $O(((b+|A|-1)^{|A|-1})^{m^2+2m}) = O(b^{(|A|-1)\cdot(m^2+2m)})$ assignments and obviously it may be done in polynomial time in the size of the input. □

From the proof of Theorem 5.3, we have that loosely-abelian algebras are a class of algebras for which TERM-SAT is in P for the whole clones.

6. Polynomial operations over 1-loosely-abelian algebras

In the next two sections, we will introduce tools needed to prove the main theorem of Section 8. In this section, we will prove three lemmas that together characterize the behavior of polynomial operations over 1-loosely-abelian algebras.

Concepts of unseparative and separative algebras will play an important role in the last three sections.

Definition 6.1. Let $\mathbf{A} = (A, F)$ be an algebra. We say that $b, c \in A$ are *indistinguishable* in $\mathbf{A}$ if for every $f \in \mathrm{Term}(\mathbf{A})$ of arity n (for $n > 1$), every $i \in \{1, 2, \ldots, n\}$ and $a_1, \ldots, a_n \in A$,

$$f(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n) = f(a_1, \ldots, a_{i-1}, c, a_{i+1}, \ldots, a_n).$$

We say that algebra $\mathbf{A} = (A, F)$ is *unseparative* if there exist $a, b \in A$ that are indistinguishable in $\mathbf{A}$. An algebra is *separative* if it is not unseparative.

The following Lemma turns out to be very useful.

Lemma 6.2. *Let $\mathbf{A} = (A, F)$ be a 1-loosely-abelian algebra. Then every $f \in \mathrm{Polyn}(\mathbf{A})$ of arity at least 3 satisfies*

$$f(x, y, y, z_4, \ldots, z_k) = f(x, x, y, z_4, \ldots, z_k).$$

Proof. Let $g(x, y, z_4, \ldots, z_k) = f(x, y, y, z_4, \ldots, z_k)$. Obviously, $g \in \mathrm{Polyn}(\mathbf{A})$ and from the fact that $\mathbf{A}$ is 1-loosely-abelian, we have that

$$\begin{aligned} f(x, y, y, z_4, \ldots, z_k) &= g(x, y, z_4, \ldots, z_k) = g(y, x, z_4, \ldots, z_k) \\ &= f(y, x, x, z_4, \ldots, z_k) = f(x, x, y, z_4, \ldots, z_k). \end{aligned}$$ □

Lemma 6.3 will be used a few times in the next two sections.

Lemma 6.3. *Let* $\mathbf{A} = (A, F)$ *be a finite separative* 1*-loosely-abelian algebra and* $f \in \mathrm{Polyn}(\mathbf{A})$*. Then*

$$f(f(x_1, x_2, \ldots, x_k), \ldots, f(x_1, x_2, \ldots, x_k)) = f(x_1, x_2, \ldots, x_k).$$

Proof. Before proceeding with the main part of the proof, we will prove two claims.

Claim 1. *Let* $p \in \mathrm{Polyn}_1(\mathbf{A})$ *and* $a, b, c \in A$*. If* $p(a) = b$*,* $p(b) = c$*, and* $p(c) = c$*, then* $b = c$

Suppose, contrary to our claim, that $b \neq c$. The fact that $\mathbf{A}$ is separative implies there exist $g_0 \in \mathrm{Term_k}(\mathbf{A})$ (for $k > 1$) and $d_2, \ldots, d_k \in A$ such that $g_0(b, d_2, \ldots, d_k) \neq g_0(c, d_2, \ldots, d_k)$. Then for the same reasons, there exist $h \in \mathrm{Term_l}(\mathbf{A}))$ (for $l > 1$) and $d_2', \ldots, d_l' \in A$, such that

$$h(g_0(b, d_2, \ldots, d_k), d_2', \ldots, d_l') \neq h(g_0(c, d_2, \ldots, d_k), d_2', \ldots, d_l').$$

Let $g_1(x_1, \ldots, x_{k+l-1}) = h(g_0(x_1, \ldots, x_k), x_{k+1}, \ldots, x_{k+l-1})$. Observe that g_1 distinguishes b and c, and has arity bigger than g_0.

In this way, we can define the sequence $(g_i)_{i=0}^{\infty}$ of operations contained in $\mathrm{Term}(\mathbf{A})$ so that for all $i \geq 0$, operation g_i distinguishes b and c and the arity of g_{i+1} is greater than that of g_i. Clearly, there is j such that arity of g_j is $t > |A| + 1$.

Assume that $e_2, \ldots, e_t \in A$ such that $g_j(b, e_2, \ldots, e_t) \neq g_j(c, e_2, \ldots, e_t)$. Since $t > |A| + 1$, at least two of the values $e_2, \ldots, e_t$ are equal. Assume, without loss of generality, that $e_2 = e_3$. From Lemma 6.2, we have that $g_j(b, e_2, e_3, e_4 \ldots, e_k) = g_j(b, e_2, e_2, e_4 \ldots, e_k) = g_j(b, b, e_3, \ldots, e_k)$.

Now using the fact that $\mathbf{A}$ is 1-loosely-abelian we obtain that

$$\begin{aligned} &g_j(b, e_2, \ldots, e_k) = g_j(b, b, e_3, \ldots, e_k) = g_j(p(a), p(a), e_3, \ldots, e_k) \\ &= g_j(p(p(a)), a, e_3, \ldots, e_k) = g_j(c, a, e_3, \ldots, e_k) = g_j(p(p(c)), a, e_3, \ldots, e_k) \\ &= g_j(p(p(a)), c, e_3, \ldots, e_k) = g_j(c, c, e_3, \ldots, e_k). \end{aligned}$$

By Lemma 6.2, $g_j(b, e_2, \ldots, e_k) = g_j(c, c, e_3, \ldots, e_k) = g_j(c, e_2, e_3, \ldots, e_k)$, which is a contradiction because $g_j(b, e_2, \ldots, e_t) \neq g_j(c, e_2, \ldots, e_t)$. This completes the proof of the claim.

The following second claim is needed in the main part of the proof of the lemma.

Claim 2. *Let* $p \in \mathrm{Polyn}_1(\mathbf{A})$ *and* $a \in A$*. If* $p^k(a) = a$*, then* $p(a) = a$*.*

To prove this claim, we only need to show that a and $p(a)$ are indistinguishable. This may be shown by a similar method to the proof of Claim 1. If $a \neq p(a)$, then we can construct the sequence $(g_i)_{i=0}^{\infty}$ of operations distinguishing a and $p(a)$ such that for all $i \geq 0$, the arity of g_{i+1} is greater than that of g_i. This implies that there is j such that l the arity of g_j is greater

than $|A| + k$. Then from Lemma 6.2, for all $b_2, \dots, b_l, b'_{k+1}, \dots, b'_l$ such that $\{b_2, \dots, b_l\} = \{b'_{k+1}, \dots, b'_l\}$, we have that

$$g_j(a, b_2, \dots, b_l) = g_j(\underbrace{a, \dots, a}_{k}, b'_{k+1}, \dots, b'_l) = g_j(\underbrace{p^k(a), a, \dots, a}_{k}, b'_{k+1}, \dots, b'_l)$$
$$= g_j(\underbrace{p(a), \dots, p(a)}_{k}, b'_{k+1}, \dots, b'_l) = g_j(p(a), b_2, \dots, b_l).$$

This contradicts the fact that g_j distinguishes a and $p(a)$ and completes the proof of Claim 2.

We are now ready to present the main part of the proof. First, we will consider the case where f is a unary function. If $f(x) = x$, then assertion of the current lemma is obvious. Assume that there exist $a_0, a_1 \in A$ such that $f(a_0) = a_1$ and $a_0 \neq a_1$. Let $a_i = f^i(a_0)$. From the fact that A is finite, there exist i and j, such that $i < j$ and $a_i = a_j$. Hence, from Claim 2, $f(a_i) = a_i$. From multiple use of Claim 1, we obtain that $f(a_0) = a_i$, and consequently $f(f(a_0)) = f(a_0)$.

Now assume that the arity of f is $k > 1$. Let $a_1, \dots, a_k \in A$ and $f'(x) = f(x, a_2, \dots, a_k)$. To shorten the notation, we will denote $a_i, a_{i+1}, \dots, a_k$ by $\overline{A_{i,k}}$. From the definition of 1-loosely-abelian algebras, we have that

$$\begin{aligned}
&f(f(\overline{A_{1,k}}), f(\overline{A_{1,k}}), f(\overline{A_{1,k}}), \dots, f(\overline{A_{1,k}})) \\
&= f(f(f(\overline{A_{1,k}}), \overline{A_{2,k}}), a_1, f(\overline{A_{1,k}}), \dots, f(\overline{A_{1,k}})) \\
&= f(f(f(f(\overline{A_{1,k}}), \overline{A_{2,k}}), \overline{A_{2,k}}), a_1, a_1, f(\overline{A_{1,k}}), \dots, f(\overline{A_{1,k}})) \\
&\vdots \\
&= f(f(f(\cdots f(f(\overline{A_{1,k}}), \overline{A_{2,k}}), \dots), \overline{A_{2,k}}), a_1, \dots, a_1).
\end{aligned}$$

Now, from Lemma 6.2, we obtain

$$\begin{aligned}
&f(f(f(\cdots f(f(\overline{A_{1,k}}), \overline{A_{2,k}}), \dots), \overline{A_{2,k}}), a_1, \dots, a_1) \\
&= f(f(f(\cdots f(f(\overline{A_{1,k}}), \overline{A_{2,k}}), \dots), \overline{A_{2,k}}), \overline{A_{2,k}}).
\end{aligned}$$

Finally, using that we have proven the lemma for unary functions, we have

$$\begin{aligned}
&f(f(f(\cdots f(f(\overline{A_{1,k}}), \overline{A_{2,k}}), \dots), \overline{A_{2,k}}), \overline{A_{2,k}}) \\
&= f'(\cdots f'(f'(a_1)) \dots) = f'(a_1) = f(a_1, \dots, a_k).
\end{aligned}$$

□

The last lemma in this section describes the relationships between different unary polynomials over the same 1-loosely-abelian algebra.

Lemma 6.4. *Let* $\mathbf{A} = (A, F)$ *be a separative* 1*-loosely-abelian algebra and* $f, g \in \mathrm{Polyn}_1(\mathbf{A})$*. Then,* $f(g(x)) = g(f(x))$*.*

Proof. To obtain a contradiction, suppose that $f(g(a)) \neq g(f(a))$ for some $a \in A$. Then there exists $h \in \mathrm{Clo}(\mathbf{A})$ of arity $k \geq 2$ such that

$$h(g(f(a), a_2, \dots, a_k) \neq h(f(g(a)), a_2, \dots, a_k)$$

for some $a_2, \ldots, a_k \in A$. On the other hand, from properties of 1-loosely-abelian algebras, we have the following:

$$\begin{aligned} &h(g(f(a), a_2, a_3, \ldots, a_k) = h(g(a_2), f(a), a_3, \ldots, a_k) \\ &= h(a, f(g(a_2)), a_3, \ldots, a_k) = h(a_2, f(g(a)), a_3, \ldots, a_k) \\ &= h(f(g(a)), a_2, a_3, \ldots, a_k). \end{aligned}$$

This contradicts that h distinguishes $f(g(a))$ and $g(f(a))$. □

7. Order on elements of a 1-loosely-abelian algebra

To show that any finite separative 1-loosely-abelian algebra $\mathbf{A} = (A, F)$ is a subreduct of a finite semilattice with constant operations, we will construct a semilattice $(A^+, \leq_{A^+})$ such that $A \subseteq A^+$ and prove that $\mathbf{A}$ is a subreduct of $(A^+, \{\sup_{\leq_{A^+}}\} \cup \mathrm{const}_{A^+})$, where const_{A^+} is a set of constant operations on A^+. This section covers the first step of the construction. We define the order $\leq_A$ on the set A and prove some simple properties of $\leq_A$.

Definition 7.1. Let $\mathbf{A}$ be a separative 1-loosely-abelian algebra. Define

$$\leq_A = \{(a, b) \in A^2 \mid a = b \text{ or } \exists_{f \in \mathrm{Polyn}_1(\mathbf{A})} f(a) = b\}.$$

Firstly, we have to show that $\leq_A$ is an order on the set A.

Lemma 7.2. *Let $\mathbf{A} = (A, F)$ be a finite separative 1-loosely-abelian algebra. The relation $\leq_A$ is an order on the set A.*

Proof. We only need to show that $\leq_A$ is reflexive, antisymmetric and transitive.

(1) Reflexivity of $\leq_A$ is obvious.

(2) Antisymmetry: We need to show that if $a \leq_A b$ and $b \leq a$, then $a = b$.

Assume that $a \leq_A b$, $b \leq_A a$, and $a \neq b$. From the definition of $\leq_A$, we have that there exist $f_1, f_2 \in \mathrm{Polyn}_1(\mathbf{A})$ such that $f_1(a) = b$ and $f_2(b) = a$. From Lemmas 6.3 and 6.4, it follows that $a = f_2(f_1(a)) = f_2(f_1(f_1(a))) = f_1(f_2(f_1(a))) = f_1(a) = b$.

Transitivity: We will now show that if $a \leq_A b$ and $b \leq_A c$, then $a \leq_A c$. Assume that $a \leq_A b$, $b \leq_A c$, and a, b, c are pairwise different. If $a = b$, $b = c$ or $a = c$, then the transitivity for a, b and c is obvious.

From the definition of $\leq_A$, we have that there exist $f_1, f_2 \in \mathrm{Polyn}_1(\mathbf{A})$ such that $f_1(a) = b$ and $f_2(b) = c$. Consider $h(x) = f_2(f_1(x))$. Obviously, $h \in \mathrm{Polyn}_1(\mathbf{A})$ and $h(a) = f_2(f_1(a)) = c$. This, from the definition of $\leq_A$, implies that $a \leq_A c$. □

We will now prove a few lemmas that will help us embed a given separative 1-loosely-abelian algebra into some semilattice.

Lemma 7.3. *Let $\mathbf{A} = (A, F)$ be a finite separative 1-loosely-abelian algebra, $f \in \mathrm{Polyn}(\mathbf{A})$ and $a, b, c_2, \ldots, c_k \in A$. If $f(a, c_2, \ldots, c_k) = b$, then $f(d_1, \ldots, d_k) = b$ for every $\{d_1, \ldots, d_k\} \subseteq \{a, b\}$ such that $b \in \{d_1, \ldots, d_k\}$.*

Proof. Consider $h(z) = f(a, z, c_3, \dots, c_k)$. From the facts that $h \in \mathrm{Polyn}_1(\mathbf{A})$ and $h(c_2) = b$ from Lemma 6.3, we have $b = h(c_2) = h(h(c_2)) = h(b)$, and consequently $f(a, b, c_3, \dots, c_k) = b$. In a similar way, we can show that

$$f(a, b, b, c_4, \dots, c_k) = f(a, b, b, b, c_5, \dots, c_k) \\ = \cdots = f(a, b, \dots, b) = f(b, \dots, b) = b.$$

From Lemma 6.2 and the fact that $\mathbf{A}$ is 1-loosely-abelian, we have that $f(d_1, \dots, d_k) = b$ for $b \in \{d_1, \dots, d_k\} \subseteq \{a, b\}$, which completes the proof. □

Lemma 7.4. *Let $\mathbf{A} = (A, F)$ be a finite separative 1-loosely-abelian algebra, $f \in \mathrm{Polyn}(\mathbf{A})$, and $a, b \in A$. If $a \leq_A b$ and $f(a, \dots, a) = a$, then for every $\{d_1, \dots, d_k\} \subseteq \{a, b\}$ such that $b \in \{d_1, \dots, d_k\}$, it holds that $f(d_1, \dots, d_k) = b$.*

Proof. If $a = b$, then the assertion of the current Lemma is obvious. Assume that $a \leq_A b$, $f(a, \dots, a) = a$, and $a \neq b$.

From the definition of $\leq_A$ and Lemma 7.3, there exists $h \in \mathrm{Polyn}_1(\mathbf{A})$ such that $h(a) = h(b) = b$. Let us set $f'(x) = f(x, a, \dots, a)$. Notice that $b = h(a) = h(f(a, \dots, a)) = h(f'(a))$. Now, from Lemma 6.4, we see that

$$b = h(f'(a)) = f'(h(a)) = f(h(a), a, \dots, a) = f(b, a, \dots, a)$$

Thus, from the definition of 1-loosely-abelian algebras and Lemma 7.3, the proof is complete. □

The next lemma gives us a connection between the separative 1-loosely-abelian algebras, relation $\leq_A$ defined in Definition 7.1, and semilattices.

Lemma 7.5. *Let $\mathbf{A} = (A, F)$ be a finite separative 1-loosely-abelian algebra, $f \in \mathrm{Polyn}(\mathbf{A})$, and $a_1, \dots, a_k \in A$. If there exists $i \in \{1, \dots, k\}$ such that $f(a_i, \dots, a_i) = a_i$, then $f(a_1, \dots, a_k) = \sup_{\leq_A}\{a_1, \dots, a_k\}$.*

Proof. Let $f(a_1, \dots, a_k) = a$. Moreover, let us assume without loss of generality, that $f(a_1, \dots, a_1) = a_1$. Note that for all $i \in \{1, \dots, k\}$, $a_i \leq_A a$. We will prove that $a = \sup_{\leq_A}\{a_1, \dots, a_k\}$.

Let $b \in A$ such that for all $i \in \{1, \dots, k\}$, there is $a_i \leq_A b$. We need only to show that $a \leq_A b$.

From the definition of $\leq_A$ and Lemma 7.3, for all $i \in \{1, \dots, k\}$, either there exists h_i such that $h_i(a_i) = h_i(b) = b$ or $b = a_i$.

If $b = a_i$ for all $i \in \{1, \dots, k\}$, then from the assumptions of the lemma, we have that $a = f(a_1, \dots, a_k) = f(a_1, \dots, a_1) = a_1 = b$ and so $a \leq_A b$.

Let us assume, without loss of generality, that there exists $j \in \{1, \dots, k\}$ such that $a_j \neq b$. Then there exists h_j such that $h_j(a_j) = h_j(b) = b$. Therefore, for all $i \in \{i, \dots, k\}$, we have that there exists h_i fulfilling $h_i(a_i) = h_i(b) = b$. If $a_i \neq b$, such an h_i exists from Lemma 7.3 and the fact that $a_i \leq_A b$. If

$a_i = b$, we get $h_i = h_j$. Hence,

$$\begin{aligned} f(a,b,\dots,b) &= f(f(a_1,\dots,a_k),h_2(b),\dots,h_k(b)) \\ &= f(f(a_1,b,\dots,b),h_2(a_2),\dots,h_k(a_k)) = f(f(a_1,b,\dots,b),b\dots,b) \\ &= f(f(a_1,h_1(b),b,\dots,b),b\dots,b) = f(f(h_1(a_1),b,\dots,b),b,\dots,b) \\ &= f(f(b,\dots,b),b,\dots,b). \end{aligned}$$

From Lemma 7.4, it follows that $f(b,\dots,b) = b$, and consequently

$$f(a,b,\dots,b) = f(f(b,\dots,b),b,\dots,b) = f(b,\dots,b) = b.$$

This implies that $a \leq_A b$. From an arbitrary choice of b, we have that $a = f(a_1,\dots,a_k) = \sup_{\leq_A}\{a_1,\dots,a_k\}$, and the lemma follows. □

8. 1-loosely-abelian algebras and semilattices

In this section, we will prove that every finite separative 1-loosely-abelian algebra is a subreduct of a semilattice with constants. We will then discuss the various finite unseparative 1-loosely-abelian algebra cases.

Lemma 8.1. *Let $\mathbf{A} = (A, F)$ be a finite separative 1-loosely-abelian algebra. Then there exists the set A' and an order $\leq_{A'}$ such that $A \subseteq A'$, $\leq_{A'} \subset A' \times A'$, and for every operation $f \in F$, one of the following holds:*

- $\forall_{a_1,\dots,a_k \in A}\ f(a_1,\dots,a_k) = \sup_{\leq_{A'}}\{a_1,\dots,a_k\}$,
- $\exists_{b\in A'}\forall_{a_1,\dots,a_k\in A}\ f(a_1,\dots,a_k) = \sup_{\leq_{A'}}\{a_1,\dots,a_k,b\}$.

Proof. To extend A, we will need the following notation:

$$\operatorname{const}(f) = \{a \in A \mid f(a,\dots,a) = a\}.$$

Notice that if $f(x_1,\dots,x_k) = \sup\{x_1,\dots,x_k,b\}$, then b is the only minimal element of $\operatorname{const}(f)$. Hence, to be able to express any fundamental operation of $\mathbf{A}$ as a supremum, we have to add, for every operation $f \in F$ such that $\operatorname{const}(f) \subsetneq A$ and the number of minimal elements of $\operatorname{const}(f)$ greater than 1, some extra element to A. We will now introduce some new notions and define the set A':

$$\begin{aligned} \operatorname{minim}(f) &= \{a \in A \mid a \text{ is minimal in the set } \operatorname{const}(f)\}, \\ \operatorname{minim}(\mathbf{A}) &= \{B \subset A \mid \exists_{f\in F} B = \operatorname{minim}(f) \text{ and } 1 < |B| \leq |\operatorname{const}(f)| < |A|\}. \end{aligned}$$

Finally, $A' = A \cup \operatorname{minim}(\mathbf{A})$.

We extend $\leq_A$ to A' in the following way:

$$\begin{aligned} \leq_{A'} = \leq_A \cup \big\{(B,B) \in \operatorname{minim}(\mathbf{A}) \times \operatorname{minim}(\mathbf{A})\big\} \\ \cup \big\{(B,c) \in \operatorname{minim}(\mathbf{A}) \times A \;\big|\; \exists_{b\in B} b \leq_A c\big\}. \end{aligned}$$

It is easy to see that $\leq_{A'}$ is an order on the set A' and that $\leq_{A'}$ restricted to A is equal $\leq_A$.

Later in the the proof, we will need the following claim.

Claim. *Let $g \in \mathrm{Pol}(\mathbf{A})$, $a, c \in A$, and $a \leq_{A'} c$. If one of the following holds*

- $\exists_{b \in \mathrm{minim}(g)}\ b \leq_{A'} c$,
- $1 < |\operatorname{minim}(g)| \leq |\operatorname{const}(g)| < |A|$ *and* $\operatorname{minim}(g) \leq_{A'} c$,

then $g(a, \dots, a) \leq_{A'} c$.

Firstly, we will prove the case when $b \leq_{A'} c$. Note that Lemma 7.4 and the fact that $g(b, \dots, b) = b$ give $g(c, \dots, c) = c$. From the definition of $\leq_A$, we have that there exists $f_1 \in \mathrm{Polyn}_1(\mathbf{A})$ such that $f_1(a) = c$, and from Lemma 7.3, that $f_1(c) = c$. Let $f_2(x) = g(x, \dots, x)$. From Lemma 6.4, we have that

$$c = f_2(f_1(a)) = f_1(f_2(a)) = f_1(g(a, \dots, a)),$$

and hence $g(a, \dots, a) \leq_{A'} c$.

Now assume that $1 < |\operatorname{minim}(g)| \leq |\operatorname{const}(g)| < |A|$ and $\operatorname{minim}(g) \leq_{A'} c$. From the fact that $\operatorname{minim}(g) \leq_{A'} c$, there exists $d \in \operatorname{minim}(g)$ such that $d \leq_{A'} c$. From the previous paragraph, we have $g(a, \dots, a) \leq_{A'} c$. This completes the proof of the claim.

Let $a_1, \dots, a_k \in A$ and $f \in F$. To prove the lemma, it is enough to prove the following:

- if $\operatorname{const}(f) = A$, then $f(a_1, \dots, a_k) = \sup_{\leq_{A'}}\{a_1, \dots, a_k\}$;
- if $\operatorname{minim}(f) = \{b\}$, then $f(a_1, \dots, a_k) = \sup_{\leq_{A'}}\{a_1, \dots, a_k, b\}$;
- if $\operatorname{minim}(f) = B$ and $1 < |B| \leq |\operatorname{const}(g)| < |A|$, then $f(a_1, \dots, a_k) = \sup_{\leq_{A'}}\{a_1, \dots, a_k, B\}$.

Note that the first case is trivial if the arity of f is 1; otherwise, it is an easy consequence of Lemma 7.5. We will consider the second and third case simultaneously. Let $d = \operatorname{minim}(f)$ if $1 < |\operatorname{minim}(f)| \leq |\operatorname{const}(f)| < |A|$, and $d = b$ if $\operatorname{minim}(f) = \{b\}$. Notice that $f(a, \dots, a) \in \operatorname{const}(f)$ for $a \in A$, and consequently, from the definition of $\leq_{A'}$, we have $d \leq_{A'} f(a, \dots, a)$ for $a \in A$. Moreover, by the above claim for all $c \in A$ such that $d \leq_{A'} c$ and $a \leq_{A'} c$, we have $f(a, \dots, a) \leq_{A'} d$. Hence, $f(a, \dots, a) = \sup_{\leq_{A'}}\{a, d\}$ for all $a \in A$ (this completes the proof if f is a unary function).

Therefore, from Lemma 6.3, Lemma 7.5, and the fact that $\mathbf{A}$ is 1-loosely-abelian, it follows that

$$\begin{aligned} f(a_1, \dots, a_k) &= f(f(a_1, \dots, a_k), \dots, f(a_1, \dots, a_k)) \\ &= f(f(a_1, \dots, a_1), f(a_2, \dots, a_2), \dots, f(a_k, \dots, a_k)) \\ &= \sup_{\leq_{A'}}\{\sup_{\leq_{A'}}\{d, a_1\}, \dots, \sup_{\leq_{A'}}\{d, a_k\}\} = \sup_{\leq_{A'}}\{d, a_1, \dots, a_k\}. \end{aligned}$$

This completes the proof of the lemma. □

We will now show that every order can be embedded in some join-semilattice in such a way that all supremums existing with respect to the order have the same value in the join-semilattice.

Lemma 8.2. *Let $(O, \leq)$ be a finite ordered set. Then there exists a join-semilattice $(S, \leq')$ such that $O \subseteq S$ and for all $A \subseteq O$, if there exists $\sup_{\leq} A$, then $\sup_{\leq'} A = \sup_{\leq} A$.*

Proof. Let S be the join-semilattice of upward closed subsets of O ordered by reverse inclusion. Embed O into S by the map taking an element to the upset it generates. It is easy to see that such the embeding preserves all existing supremums. □

The main theorem of this section is a straightforward consequence of Lemmas 8.1 and 8.2.

Theorem 8.3. *Let $\mathbf{A}$ be a finite separative algebra. The following are equivalent:*

- $\mathbf{A}$ *is a* 1*-loosely-abelian algebra.*
- $\mathbf{A}$ *is a subreduct of a semilattice with constant operations.*

Proof. Let $\mathbf{A} = (A, F)$ be 1-loosely-abelian. From Lemma 8.1, we have that there exists a set $A' \supseteq A$ and an order on A' such that every operation $f \in F$ can be expressed as a supremum with respect to this order. Now, Lemma 8.2 allows us to extend A' to a semilattice A'' that preserves all supremums existing in A'. This preservation shows how the operations from F can be extended to A''. On the other hand, every semilattice is 1-loosely-abelian. Therefore, every subreduct of a semilattice and every subreduct of a semilattice with constant operations is 1-loosely-abelian. □

Notice that there are finite separative 1-loosely-abelian algebras that are not polynomially equivalent to any semilattice.

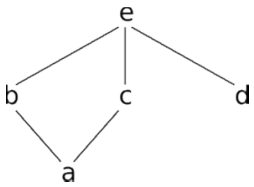

FIGURE 1. Hasse diagram of the semilattice $\mathbf{A}$ from Example 8.4

Example 8.4. Let $\mathbf{A} = (\{a, b, c, d, e\}, \sup_{\leq})$, where $\leq$ is an order whose Hasse diagram is in Figure 1.

Then $\mathbf{B} = (\{b, c, d, e\}, \sup_{\leq}, f)$, where $f(x, y) = \sup_{\leq}\{a, x, y\}$, is a separative 1-loosely-abelian algebra that is not polynomially equivalent to any semilattice.

We characterized in Theorem 8.3 separative 1-loosely-abelian algebras. If a 1-loosely-abelian algebra $\mathbf{A}$ is unseparative, there are many possibilities. Firstly, $\mathbf{A}$ may be a subreduct of some semilattice with constant operations.

Example 8.5. Let $\mathbf{A} = (\{0,1,2\}, f)$ and $f(x,y) = \sup_{\leq}\{1,x,y\}$, where $\leq$ is the usual order on numbers. Then

- $\mathbf{A}$ is a subreduct of a semilattice with constants,
- $\mathbf{A}$ is 1-loosely-abelian, and
- $\mathbf{A}$ is unseparative (0 and 1 are indistinguishable).

The second important case, in which every two elements are indistinguishable, is the class of unary algebras.

Example 8.6. Let $\mathbf{A}$ be a unary algebra. Then $\mathbf{A}$ is 1-loosely-abelian.

Finally, as the following example illustrates, there are many more complicated cases.

Example 8.7. Let $\mathbf{A} = (\{0,1,2\}, f_1, f_2)$, $f_1(x,y) = \sup_{\leq}\{1,x,y\}$, where $\leq$ is the usual order on numbers and f_2 is given by the table below.

x	0	1	2
$f_2(x)$	1	0	2

Then

- $\mathbf{A}$ is 1-loosely-abelian and unseparative,
- $\mathbf{A}$ is not a subreduct of any semilattice, and
- $\mathbf{A}/\theta$, where $\theta = \{(0,0),(1,1),(2,2),(0,1),(1,0)\}$, is a separative 1-loosely-abelian algebra (so it is a subreduct of a semilattice).

Acknowledgment. I would like to thank Michał Stronkowski for the conversation held in Novi Sad, which inspired me to define loosely-abelian algebras.

References

[1] Burris, S., Lawrence, J.: The equivalence problem for finite rings. J. Symbolic Comput. **15**, 67–71 (1993)
[2] Freese, R., McKenzie, R.: Commutator Theory for Congruence Modular Varieties. London Math. Soc. Lecture Note Ser., vol. 125. Cambridge University Press, Cambridge (1987)
[3] Goldmann, M., Russel, A.: The complexity of solving equations over finite groups. Inform. and Comput. **178**, 253–262 (2002)
[4] Gorazd, T., Krzaczkowski, J.: Term equation satisfiability over finite algebras. Internat. J. Algebra Comput. **20**, 1001–1020 (2010)
[5] Gorazd, T., Krzaczkowski, J.: The complexity of problems connected with two-element algebras. Rep. Math. Logic **26**, 91–108 (2011)
[6] Hobby, D., McKenzie, R.: The Structure of Finite Algebras. Contemporary Mathematics, vol. 76. American Mathematical Society, Providence (1988)
[7] Horváth, G.: The complexity of the equivalence and equation solvability problems over nilpotent rings and groups. Algebra Universalis **66**, 391–403 (2011)
[8] Horváth, G., Szabó, C.: The complexity of checking identities over finite groups. Internat. J. Algebra Comput. **16**, 931–939 (2006)
[9] Horváth, G., Szabó, C.: The extended equivalence and equation solvability problems for groups. Discrete Math. Theor. Comput. Sci. **13**, 3–32 (2011)
[10] Horváth, G., Szabó, C.: Equivalence and equation solvability problems for the group A_4. J. Pure Appl. Algebra **216**, 2170–2176 (2012)

[11] Klíma, O., Tesson, P., Thérien, D.: Dichotomies in the complexity of solving systems of equations over finite semigroups. Theory Comput. Syst. **40**, 263–297 (2007)
[12] Larose, B., Zádori, L.: Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. Internat. J. Algebra Comput. **16**, 563–581 (2006)
[13] Schwarz, B.: The complexity of satisfiability problems over finite lattices. In: Proceedings of the 21st Annual Symposium on Theoretical Aspects of Computer Science (Montpellier 2004). Lecture Notes in Comput. Sci., vol. 2996, pp. 31–43. Springer, Berlin (2004)
[14] Warne, R.J.: Semigroups obeying the term condition. Algebra Universalis **31**, 113–123 (1994)

Jacek Krzaczkowski

Institute of Computer Science,, Maria Curie-Skłodowska University in Lublin, pl. M. Curie-Skłodowskiej 1, 20-031, Lublin
e-mail: krzacz@umcs.lublin.pl