A Dichotomy Theorem for Constraint Satisfaction Problems on a 3-Element Set

ANDREI A. BULATOV

Simon Fraser University, Burnaby, Canada

Abstract. The Constraint Satisfaction Problem (CSP) provides a common framework for many combinatorial problems. The general CSP is known to be NP-complete; however, certain restrictions on a possible form of constraints may affect the complexity and lead to tractable problem classes. There is, therefore, a fundamental research direction, aiming to separate those subclasses of the CSP that are tractable and those which remain NP-complete.

Schaefer gave an exhaustive solution of this problem for the CSP on a 2-element domain. In this article, we generalise this result to a classification of the complexity of the CSP on a 3-element domain. The main result states that every subproblem of the CSP is either tractable or NP-complete, and the criterion separating them is that conjectured in Bulatov et al. [2005] and Bulatov and Jeavons [2001b]. We also characterize those subproblems for which standard constraint propagation techniques provide a decision procedure. Finally, we exhibit a polynomial time algorithm which, for a given set of allowed constraints, outputs if this set gives rise to a tractable problem class. To obtain the main result and the algorithm, we extensively use the algebraic technique for the CSP developed in Jeavons [1998b], Bulatov et al. [2005], and Bulatov and Jeavons [2001b].

Categories and Subject Descriptors: F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes—*Reducibility and completeness*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Logic and constraint programming*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*Combinatorial algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Constraint satisfaction problem, homomorphism problem, complexity, dichotomy theorem

1. *Introduction*

In the Constraint Satisfaction Problem (CSP) [Montanari 1974], we aim to find an assignment to a set of variables subject specified constraints. This problem can also be reformulated either as (1) the problem of deciding the existence of a homomorphism between two finite relational structures **A** and **B** [Feder and Vardi 1998; Kolaitis and Vardi 2000b; Kolaitis 2003] (this form we frequently

---

Author's address: School of Computing Science, Simon Fraser University, Burnaby, Canada, e-mail: abulatov@cs.sfu.ca.

use throughout the article along with the standard form), or as (2) the problem of deciding whether there is a model of a conjunctive formula, or as (3) the problem of deciding whether the evaluation $Q(D)$ of a conjunctive query Q on a database D is non-empty; see Feder and Vardi [1998] and Kolaitis and Vardi [2000a].

It is remarkable that many combinatorial problems appearing in computer science and artificial intelligence can be expressed as particular subclasses of the CSP. The standard examples include the propositional satisfiability problem, in which the variables must be assigned Boolean values, graph colorability, scheduling problems, systems of linear equations and many others. An advantage of considering a common framework for all of these diverse problems is that it makes it possible to obtain generic structural results concerning the computational complexity of constraint satisfaction problems that can be applied in many different areas such as database theory [Kolaitis and Vardi 2000a; Vardi 2000], temporal and spatial reasoning [Schwalb and Vila 1998], machine vision [Montanari 1974], belief maintenance [Dechter and Dechter 1996], technical design [Nadel and Lin 1991], natural language comprehension [Allen 1994], programming language analysis [Nadel 1995], etc.

The general CSP is NP-complete; however, certain restrictions on the allowed form of the constraints involved may ensure tractability. We call a problem *tractable* if it is solvable in polynomial time. Therefore, one of the main approaches in study of the CSP is to identify tractable subclasses of the general CSP obtained in this way [Schaefer 1978; Feder and Vardi 1998; Jeavons 1998b; Gottlob et al. 2000; Creignou et al. 2001]. Developments in this direction provide efficient algorithms solving some particular problems, which fall in one of the known tractable subclasses, or assist in speeding up of general superpolynomial algorithms [Dechter and Pearl 1988; Kumar 1992; Dechter and Meiri 1994].

To formalize the idea of restricting the allowed constraints, we use the notion of a *constraint language* [Jeavons 1998a], which is simply a set of possible relations that can be used to specify constraints in a problem. Equivalently, a class of CSPs in the form of homomorphism problems for relational structures can be defined by specifying the allowed target structures [Feder and Vardi 1998; Kolaitis and Vardi 2000b; Kolaitis 2003]. The ultimate goal of this research direction is to find the precise 'border' between tractable and intractable constraint languages or tractable and intractable relational structures. This goal was achieved by Schaefer [1978] in the important case of Boolean constraints, that is with a 2-element set of possible values. He has characterised tractable constraint languages and proved that the rest are NP-complete. Schaefer's result is known as Dichotomy Theorem for Boolean constraints. Remarkably, all constraint languages known so far are also either tractable or NP-complete. Dichotomy theorems are of particular interest in study of the CSP, because, on the one hand, they determine the precise complexity of constraint languages, and on the other hand, the a priori existence of a dichotomy result cannot be taken for granted. Ladner [1975] showed that if P$\neq$NP, then there are problems in NP that are neither NP-complete nor in P. Feder and Vardi [1998] conjectured that dichotomy holds for all classes of the CSP defined by constraint languages. For more dichotomy results for Boolean CSPs and a brief survey of dichotomy results for other cases, the reader is referred to Creignou et al. [2001].

The analogous problem for the CSP in which the variables can be assigned more than 2 values remains open since 1978, in spite of intensive efforts. For instance, Feder and Vardi [1998] used database technique and group theory to identify some large tractable families of constraints; Jeavons and co-authors have characterized many tractable and NP-complete constraint languages using invariance properties of constraints [Jeavons et al. 1997, 1998a; Jeavons 1998b]; in Bulatov et al. [2005], a possible form of a dichotomy result for the CSP on finite domains was conjectured; in Bulatov et al. [2001], a dichotomy result was proved for a certain type of constraint languages on a 3-element domain.

In this article, we generalize the results of Schaefer [1978] and Bulatov et al. [2001] and prove the dichotomy conjecture from Bulatov et al. [2005] for the constraint satisfaction problem on a 3-element domain. In particular, we completely characterize tractable constraint languages in this case and prove that the rest are NP-complete. The main result will be precisely formulated at the end of Section 2.

Another achievement of this article is that we characterize those constraint satisfaction problems on a 3-element domain, which are solvable by "local algorithms." Many of the standard algorithms for the CSP use constraint propagation, which on each pass solves the problem restricted to a limited number of variables or constraints, finds the solutions of such a restricted problem, and then use this information to tighten the constraints and so to reduce the search space. We say that a problem is solvable by a *local algorithm* if there exists such a heuristic that provides a decision procedure, that is, if, looking at the resulted transformed problem, we are able to conclude whether the original problem has a solution. An exact definition of local algorithms will be given in Section 4.1. There are close connections between solvability by local algorithms, expressibility of the problem in terms of Datalog and properties of certain abstract games [Kolaitis 2003]. Such connections are especially clear and clean when a constraint satisfaction problem is represented in the form of the homomorphism problem.

The dichotomy problem for a domain containing more than 2 elements, even for a 3-element domain, turns out to be much harder than that for the 2-element case. Besides the obvious reason that Boolean CSPs closely relates to various problems from propositional logic, and therefore are much better investigated, there is another deep reason. As showed in Jeavons et al. [1997, 1998b] and Jeavons [1998b], when studying the complexity of constraint languages, we may restrict ourselves to a certain class of languages, so called *relational clones*. There are only countably many relational clones on a 2-element set, and all of them are known [Post 1941]. However, the class of relational clones on a set with 3 or more elements contains continuum many elements, and is believed to be incomprehensible.

Another problem tackled here is referred to, in Creignou et al. [2001], as the *meta-problem*: given a constraint language, decide whether it gives rise to a tractable problem class. Making use of the dichotomy theorem obtained, we provide an efficient algorithm solving the meta-problem for the CSP on a 3-element domain. Furthermore, following Kolaitis and Vardi [2000a], we consider the *uniform* constraint satisfaction problem, in which the constraint language is treated as a part of the input. We show that the uniform problem for the class of tractable constraint languages is also solvable in polynomial time.

The technique used in this article relies upon the idea developed in Jeavons [1998b], Bulatov et al. [2005], and Bulatov and Jeavons [2001b] (and also mentioned in Feder and Vardi [1998] as a possible direction for future research) that algebraic invariance properties of constraints can be used for studying the complexity of the corresponding constraint satisfaction problems. These invariance properties are usually expressed in the form of *polymorphisms* of a constraint language. The main advantage of this technique is that it allows us to employ structural results from universal algebra. The algebraic approach has proved to be very fruitful in identifying tractable classes of the CSP [Jeavons et al. 1998a; Bulatov and Jeavons 2000; Bulatov 2002]. A usual result in the mentioned papers and in other previous papers using this approach exploits some particular properties of polymorphisms to design an algorithm solving constraint problems of the corresponding type. However, there are infinitely many different possible polymorphisms even on small sets; thus, it is hardly possible that this way will lead us to a complete complexity classification of CSPs. Here, we use a more advanced technique, first suggested in Bulatov et al. [2005] and Bulatov and Jeavons [2001b], that uses structural elements of finite universal algebras such as subalgebras and congruences. The advantage of this method is clear: small algebras (correspondingly, small domains) may have only few distinct structural elements. This makes it possible for small domains to perform a comprehensive case analysis; and this is exactly what is done in this paper in the case of a 3-element domain. Of course, this method does not work for bigger domains, and more sophisticated techniques are required. We strongly believe that the synthesis between complexity theory and universal algebra, which we describe here, is likely to lead to new results in both fields.

The article is organized as follows: In Section 2, we recall basic definitions and results about the CSP and algebraic approach in the study of the CSP. In Section 3, we introduce 10 properties such that every constraint language on a 3-element set that gives rise to a tractable problem satisfies one of them. Then, in Section 4, we recall notions related to local algorithms and multisorted constraint satisfaction problems, provide polynomial-time algorithms for each of the 10 properties and prove a criterion characterizing problems solvable by a local algorithm. The metaproblem is solved in Section 5; we exhibit a polynomial time algorithm that allows one to recognize if a constraint language gives rise to a tractable problem. The main technical theorem is proved in Section 6; finally, in Section 7, we show how the uniform problem can be solved, that is we present a polynomial time algorithm that, given a constraint language, decides which algorithm from Section 4 can solve the corresponding constraint problem. We frequently use, especially in Section 6, some basic algebraic notions, results and methods. Although, in order to make the paper self-contained, most of them are defined in the article; a keen reader can find a detailed introduction into the corresponding algebraic areas in Burris and Sankappanavar [1981] and McKenzie et al. [1987].

2. *Algebraic Structure of CSP Classes*

2.1. CONSTRAINT SATISFACTION PROBLEM. The set of all n-tuples with components from a set A is denoted A^n. The ith component of a tuple $\mathbf{a}$ will be denoted by $\mathbf{a}[i]$. Any subset of A^n is called an n-ary *relation* on A; and a *constraint language* on A is an arbitrary set of finitary relations on A. The Constraint Satisfaction Problem has been introduced by Montanari [1974]. Here is a standard formulation of it.

Definition 1 (*Constraint Satisfaction Problem*). The *constraint satisfaction problem* (*CSP*) over a constraint language Γ, denoted CSP(Γ), is defined to be the decision problem with instance $(V; A; \mathcal{C})$, where

—V is a set of *variables*;

—A is a set of *values* (sometimes called a *domain*); and

—$\mathcal{C}$ is a set of *constraints*, $\{C_1, \dots, C_q\}$,
in which the constraint $C_i \in \mathcal{C}$ is a pair $\langle s_i, R_i \rangle$ with s_i is a tuple of variables of length m_i, called the *constraint scope*, and $R_i \in \Gamma$ an m_i-ary relation on A, called the *constraint relation*.

The question is whether there exists a *solution* to $(V; A; C)$, that is, a function from V to A, such that, for each constraint in $\mathcal{C}$, the image of the constraint scope is a member of the constraint relation.

The size of a problem instance is taken to be the length of a string containing all constraint scopes and all tuples of all constraint relations from the instance.

It will sometimes be convenient for us to abuse terminology and to call an instance of the constraint satisfaction problem a constraint satisfaction problem. It was observed in Feder and Vardi [1998] that the Constraint Satisfaction Problem can be equivalently reformulated in the form of the Homomorphism Problem. To do this, we need more definitions.

A *vocabulary* is a finite set of relational symbols $R_1, \dots, R_n$ each of which has a fixed arity. A *relational structure* over the vocabulary $R_1, \dots, R_n$ is a tuple $\mathcal{H} = (H; R_1^{\mathcal{H}}, \dots, R_n^{\mathcal{H}})$ such that H is a nonempty set, called the *universe* of $\mathcal{H}$, and each $R_i^{\mathcal{H}}$ is a relation on H having the same arity as the symbol R_i. Let $\mathcal{G}, \mathcal{H}$ be relational structures over the same vocabulary $R_1, \dots, R_n$. A *homomorphism* from $\mathcal{G}$ to $\mathcal{H}$ is a mapping $\varphi: G \to H$ from the universe of $\mathcal{G}$ (the *source structure*) to the universe H of $\mathcal{H}$ (the *target structure*) such that, for every relation $R^{\mathcal{G}}$ of $\mathcal{G}$ and every tuple $(\mathbf{a}[1], \dots, \mathbf{a}[m]) \in R^{\mathcal{G}}$, we have $(\varphi(\mathbf{a}[1]), \dots, \varphi(\mathbf{a}[m])) \in R^{\mathcal{H}}$.

Let $\mathfrak{H}$ be a class of relational structures over a vocabulary $R_1, \dots, R_n$. In the *uniform homomorphism problem associated with* $\mathfrak{H}$ ($\mathsf{HOM}(\mathfrak{H})$), the question is, given a structure $\mathcal{H} \in \mathfrak{H}$ over a vocabulary $R_1, \dots, R_n$ and a structure $\mathcal{G}$ over the same vocabulary, whether there exists a homomorphism from $\mathcal{G}$ to $\mathcal{H}$. If $\mathfrak{H}$ consists of a single structure $\mathcal{H}$, then we write $\mathsf{HOM}(\mathcal{H})$ instead of $\mathsf{HOM}(\{\mathcal{H}\})$. We refer to such a problem as a *nonuniform homomorphism problem*, because the inputs are just source structures.

The intuition behind the mentioned equivalence between the constraint satisfaction problem and the homomorphism problem is that the source structure in the latter represents the variables and the constraint scopes, while the target structure represents the domain of values and the constraint relations. Moreover, the homomorphisms between the structures are precisely the solutions to the constraint satisfaction problem.

If a constraint language $\Gamma = \{R_1, \dots, R_n\}$ on a set H is finite, then $\mathrm{CSP}(\Gamma)$ can be identified with the nonuniform problem $\mathsf{HOM}(\{\mathcal{H}\})$, where $\mathcal{H} = (H; R_1, \dots, R_n)$. If Γ is infinite, then $\mathrm{CSP}(\Gamma)$ is equivalent to the uniform problem $\mathsf{HOM}(\mathfrak{H})$, where $\mathfrak{H}$ is the class of structures $\mathcal{H} = (A; R_1^A, \dots, R_n^A)$, $R_1^A, \dots, R_n^A \in \Gamma$. Nonuniform problems are widely studied, see, for example, Feder and Vardi [1998], Jeavons et al. [1998b], Jeavons [1998b], Kolaitis and Vardi [2000a, 2000b], and Kolaitis [2003]. On the one hand, since the technique we use in this article makes it more natural to deal with infinite constraint languages, we state our main results within the framework of the constraint satisfaction problem and, correspondingly, uniform homomorphism problems. On the other hand, many of the results can be stated in a stronger form for nonuniform homomorphism problems. We therefore use this formalism as well.

We shall be concerned with distinguishing between those constraint languages (relational structures), which give rise to *tractable* problems, that is, problems for which there exists a polynomial-time solution algorithm, and those which do not.

A relational structure $\mathcal{H}$ is said to be *tractable* if $\mathsf{HOM}(\mathcal{H})$ is tractable. It is said to be *NP-complete* if $\mathsf{HOM}(\mathcal{H})$ is NP-complete. A class of finite structures $\mathfrak{H}$ is said to be *tractable* if every $\mathcal{H} \in \mathfrak{H}$ is tractable. It is said to be *NP-complete* if there is NP-complete $\mathcal{H} \in \mathfrak{H}$.

Analogously, a constraint language Γ is said to be *tractable*, if CSP(Γ') is tractable for each finite subset $\Gamma' \subseteq \Gamma$. It is said to be *NP-complete*, if CSP(Γ') is NP-complete for some finite subset $\Gamma' \subseteq \Gamma$.

Schaefer [1978] has classified two-element structures or, equivalently, Boolean constraint languages with respect to the complexity. This result is known as Schaefer's Dichotomy theorem.

PROPOSITION 1 (SCHAEFER'S DICHOTOMY THEOREM [SCHAEFER 1978]). *A Boolean constraint language Γ is tractable if and only if one of the following six conditions holds:*

(1) *every R in Γ contains* $(0, \ldots, 0)$.
(2) *every R in Γ contains* $(1, \ldots, 1)$.
(3) *every R in Γ is definable by a CNF formula in which each clause has at most one negated variable.*
(4) *every R in Γ is definable by a CNF formula in which each clause has at most one unnegated variable.*
(5) *every R in Γ is definable by a CNF formula in which each clause has at most two literals.*
(6) *every R in Γ is the solution space of a system of linear equations over* GF(2).

Otherwise, Γ is NP-complete.

More examples of both tractable and NP-complete relational structures will appear later in this article and can also be found in Jeavons et al. [1997], Feder and Vardi [1998], Bulatov et al. [2005], and Dalmau [2000].

The following classification problem for larger structures is still open and seems to be very interesting and hard [Feder and Vardi 1998].

Problem 1 (*Classification Problem*). Characterize tractable relational structures [constraint languages].

From a more practical perspective, it is also important to find algorithms to solve constraint satisfaction problems whenever it is tractable, and to find an efficient algorithm to recognize tractable cases. So we pose the following:

Problem 2 (*Algorithmic Problem*). Find efficient (polynomial time) algorithms for $\mathsf{HOM}(\mathcal{H})$ [for CSP(Γ)], where $\mathcal{H}$ [Γ] is tractable.

Problem 3 (*Meta-Problem*). Find an efficient (polynomial time) algorithm, if any, that decides whether a relational structure [a constraint language] is tractable or not.

It follows from Proposition 1 that every Boolean constraint language [respectively, 2-element relational structure] is either tractable or NP-complete; and so there is no language of intermediate complexity. Moreover, in all known cases, problems of the form $\mathsf{HOM}(\mathcal{H})$ or CSP(Γ) are either tractable or NP-complete. Feder and Vardi [1998] conjectured that this is the case for all problems of the forms $\mathsf{HOM}(\mathcal{H})$ and CSP(Γ). A criterion distinguishing tractable problems has been suggested in Bulatov et al. [2005]. Therefore, the following problem is very plausible.

Problem 4 (*Dichotomy Problem*). Is it true that every relational structure [constraint language] is either tractable or NP-complete?

2.2. Algebraic Structure of Problem Classes. The algebraic approach to the CSP has been developed in Jeavons et al. [1997], Jeavons [1998b], Bulatov et al. [2005], and Bulatov and Jeavons [2001b, 2003]. However, since we shall intensively use algebraic concepts, we find it useful to give a short overview of this approach here. First, we remind the notion of a polymorphism of a relational structure and Jeavons' theorem that links polymorphisms and the complexity of CSPs, and provide some examples. The next step is to introduce the notion of the universal algebra dual to a relational structure and to translate the properties of polymorphism of the structure to properties of the corresponding algebra.

Schaefer's successful study of Boolean constraint satisfaction problem heavily uses the technique based on representation of Boolean relations by propositional formulas. Such a representation does not exist for larger domains. Instead, we shall use algebraic properties of relations. In the algebraic definitions below, we mainly follow Burris and Sankappanavar [1981] and McKenzie et al. [1987].

Every relational structure $\mathcal{H}$ has a collection of associated operations on the same universe. The unary operations associated to the structure are widely used: they are the *endomorphisms* of $\mathcal{H}$ that is homomorphisms of the structure into itself. We shall use operations of arbitrary arity. An n-ary *polymorphism* of a relational structure $\mathcal{H}$ is a homomorphism from the n-th Cartesian power of $\mathcal{H}$, that is $\mathcal{H} \times \cdots \times \mathcal{H}$, into $\mathcal{H}$. The set of all polymorphisms of $\mathcal{H}$ is denoted by $\mathsf{Pol}\ \mathcal{H}$.

On a lower level, polymorphisms can be defined as operations preserving relations. An n-ary operation f *preserves* an m-ary relation R (or f is a *polymorphism* of R, or R is *invariant* under f) if, for any $(a_{11}, \dots, a_{m1}), \dots, (a_{1n}, \dots, a_{mn}) \in R$, the tuple $(f(a_{11}, \dots, a_{1n}), \dots, f(a_{m1}, \dots, a_{mn}))$ belongs to R.

For a constraint language Γ, the set of operations which are polymorphisms of every relation from Γ is denoted $\mathsf{Pol}\ \Gamma$. The set of all relations invariant under every operation from a set F is denoted $\mathsf{Inv}\ F$.

To illustrate the notions introduced, we consider a simple example.

Example 1. Let $A = \{0, 1\}$. The operation $x \oplus y \oplus z$, where $\oplus$ denotes addition modulo 2, is a polymorphism of the relation

$$R = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Moreover, every polymorphism of R is an operation of the form $x_1 \oplus \cdots \oplus x_n$ or $x_1 \oplus \cdots \oplus x_n \oplus 1$, n odd, on A. The invariants of $x \oplus y \oplus z$ are the relations definable by a system of linear equations on A. For instance, the relation R is defined by the equation $x \oplus y \oplus z \oplus t = 1$.

Operators Pol and Inv form so-called *Galois connection*, see for example, Pöschel and Kalužnin [1979] and Denecke and Wismath [2002]. In particular, they satisfy the conditions

$$\mathsf{Pol}\ \mathsf{Inv}\ \mathsf{Pol}\ \Gamma = \mathsf{Pol}\ \Gamma, \qquad \mathsf{Inv}\ \mathsf{Pol}\ \mathsf{Inv}\ F = \mathsf{Inv}\ F.$$

A connection between the complexity of a relational structure and its polymorphisms is provided by the following theorem.

PROPOSITION 2 (THEOREM 3.4 OF JEAVONS [1998B]). *Let $\mathcal{H}, \mathcal{G}$ be finite relational structures with the same universe. If $\mathsf{Pol}\ \mathcal{H} \subseteq \mathsf{Pol}\ \mathcal{G}$, then $\mathsf{HOM}(\mathcal{G})$ is polynomial time reducible to $\mathsf{HOM}(\mathcal{H})$.*

COROLLARY 1.

(1) *If $\mathcal{H}_1, \mathcal{H}_2$ are relational structures and $\mathsf{Pol}\ \mathcal{H}_1 = \mathsf{Pol}\ \mathcal{H}_2$, then $\mathcal{H}_1$ is tractable [NP-complete] if and only if so is $\mathcal{H}_2$.*
(2) *If Γ_1, Γ_2 are constraint languages over a finite set A and $\mathsf{Pol}\ \Gamma_1 = \mathsf{Pol}\ \Gamma_2$, then Γ_1 is tractable [NP-complete] if and only if so is Γ_2.*

It is well known (see Pöschel and Kalužnin [1979] and Denecke and Wismath [2002]) that, for any finite $\Delta \subseteq \mathsf{Inv}\ \mathsf{Pol}\ \Gamma$, there is a finite $\Delta' \subseteq \Gamma$ such that $\Delta \subseteq \mathsf{Inv}\ \mathsf{Pol}\ \Delta'$. Therefore, by Proposition 2, we obtain

COROLLARY 2. *A constraint language Γ over a finite set is tractable [NP-complete] if and only if $\mathsf{Inv}\ \mathsf{Pol}\ \Gamma$ is tractable [NP-complete].*

Informally speaking, Corollary 1 says that the complexity of a constraint language or a relational structure is determined by the set of its polymorphisms. Therefore, it seems to be a good strategy to identify which types of polymorphisms correspond to tractable and NP-complete problems.

Jeavons et al. [1997, 1998a] have identified certain types of operations, which give rise to tractable problem classes.

Let A be a finite set. An operation f on A is called

—a *projection* if there is $i \in \{1, \dots, n\}$ such that $f(x_1, \dots, x_n) = x_i$ for any $x_1, \dots, x_n \in A$;

—*essentially unary* if $f(x_1, \dots, x_n) = g(x_i)$ for some bijective unary operation g and any $x_1, \dots, x_n \in A$;

—a *constant* operation if there is $c \in A$ such that $f(x_1, \dots, x_n) = c$ for any $x_1, \dots, x_n \in A$;

—a *semilattice* operation,[1] if it is binary and satisfies the following three conditions:
(a) $f(x, f(y, z)) = f(f(x, y), z)$ (Associativity),
(b) $f(x, y) = f(y, x)$ (Commutativity),
(c) $f(x, x) = x$ (Idempotency), for any $x, y, z \in A$;

[1] Note that in some earlier papers [Jeavons et al. 1998a; Jeavons 1998b], the term, *ACI operation*, is used for this type of operations.

—a *binary conservative commutative* operation if $f(x, y) = f(y, x)$ and $f(x, y) \in \{x, y\}$ for any $x, y \in A$;

—a *majority* operation if it is ternary, and $f(x, x, y) = f(x, y, x) = f(y, x, x) = x$ for any $x, y \in A$;

—*affine* if $f(x, y, z) = x - y + z$ for any $x, y, z \in A$, where $+, -$ are the operations of an additive Abelian group;

—*Mal'tsev* if $f(x, y, y) = f(y, y, x) = x$ for any $x, y \in A$; notice that every affine operation is Mal'tsev.

The results given in the following proposition have been obtained in Jeavons et al. [1997, 1998a], Bulatov and Jeavons [2000], Bulatov [2002].

PROPOSITION 3. *If a relational structure* [*a constraint language*] *has a polymorphism which is a constant, semilattice, binary conservative commutative, Mal'tsev, or majority operation, then the structure* [*the constraint language*] *is tractable.*

PROPOSITION 4 [JEAVONS ET AL. 1997, 1998A]. *If every polymorphism of a relational structure* [*a constraint language*] *is an essentially unary operation, then it is NP-complete.*

The 2-element relational structures from Schaefer's six classes (see Proposition 1) have a constant polymorphism 0 or 1 in cases (1), (2); a semilattice polymorphism $\vee$ or $\wedge$ in cases (3), (4); the majority polymorphism $(x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$ in case (5); and the affine polymorphism $x \oplus y \oplus z$ in case (6).

PROPOSITION 5 [SCHAEFER 1978; BULATOV ET AL. 2005]. *A 2-element relational structure $\mathcal{H}$ is tractable if and only if* $\mathsf{Pol}\ \mathcal{H}$ *contains a constant, or semilattice, or majority, or affine operation. Otherwise, $\mathcal{H}$ is NP-complete.*

Relational structures are defined to be sets endowed with some family of relations. Sets endowed with some family of operations have been intensively studied and is the main object to study in universal algebra. An *algebra* is a pair $\mathbb{A} = (A, F)$ such that A is a nonempty set and F is a family of finitary operations on A. The set A is called the *universe* of $\mathbb{A}$, the operations from F are called *basic*. An algebra with a finite universe is referred to as a *finite algebra*.

Algebras naturally appear as objects *dual* to relational structures. Let $\mathcal{H}$ be a relational structure with the universe A. The algebra $(A, \mathsf{Pol}\ \mathcal{H})$ is called the *algebra associated with* $\mathcal{H}$, and is denoted $\mathbb{A}_{\mathcal{H}}$.

Similarly, given a constraint language Γ on A, the algebra $(A, \mathsf{Pol}\ \Gamma)$ is called the *algebra associated with* Γ and is denoted by $\mathbb{A}_\Gamma$.

The reverse connection is ambiguous as many relational structures and constraint languages may give rise to the same algebra. However, for any finite algebra $\mathbb{A} = (A; F)$, there is the greatest constraint language associated with $\mathbb{A}$, namely, the language $\mathsf{Inv}\ F$. Therefore, the problem class CSP($\mathsf{Inv}\ F$) can be associated with $\mathbb{A}$. We will denote it by CSP($\mathbb{A}$). The algebra $\mathbb{A}$ is said to be *tractable* if $\mathsf{Inv}\ F$ is tractable. It is said to be *NP-complete* if $\mathsf{Inv}\ F$ is NP-complete.

The language $\mathsf{Inv}\ F$ is always infinite, which means that $\mathbb{A}$ cannot be assigned a single relational structure. Instead, it can be associated to the class of structures with the universe A and relations from $\mathsf{Inv}\ F$.

Clearly, an arbitrary algebra $\mathbb{A} = (A; F)$ is not necessarily of the form $\mathbb{A}_\Gamma$ for any constraint language Γ. Therefore, there exist polymorphisms of $\mathsf{Inv}\ F$ which are not in F. Such "implicit" operations are very important in our study. For a finite algebra $\mathbb{A} = (A; F)$, an operation from $\mathsf{Pol}\ \mathsf{Inv}\ F$ is said to be a *term* operation of $\mathbb{A}$. If Γ is a constraint language, then term operations of $\mathbb{A}_\Gamma$ are the polymorphisms of Γ. Every term operation of $\mathbb{A}$ can be obtained from operations of F by means of superpositions.

We illustrate the definitions above by the following example.

Example 2. Let $\mathbb{A} = (\{0, 1\}; \{f\})$ where $f(x, y, z) = x \oplus y \oplus z$. Then, for any odd n, the operation $x_1 \oplus x_2 \oplus \cdots \oplus x_n$ is a term operation of $\mathbb{A}$ because

$$x_1 \oplus x_2 \oplus \cdots \oplus x_n = \underbrace{f(f(\ldots f(f}_{\frac{n-1}{2}\text{ times}}(x_1, x_2, x_3), x_4, x_5), \ldots), x_{n-1}, x_n).$$

As is easily seen, all results on polymorphisms of a relational structure including Proposition 3 and Proposition 4 can be converted in algebraic form. An algebra is said to be a *G-set* if every its term operation is essentially unary.

PROPOSITION 6 [JEAVONS ET AL. 1997, 1998A]. *If a finite algebra has a term operation that is constant, semilattice, binary conservative commutative, Mal'tsev, or majority, then it is tractable.*

PROPOSITION 7 [JEAVONS ET AL. 1997, 1998A]. *A finite G-set with at least* 2 *elements is NP-complete.*

By combining those two results, and the classical result of E.Post that describes all sets of the form $\mathsf{Pol}\ \Gamma$ for constraint languages Γ on a 2-element set [Post 1941], the algebraic version of Schaefer's theorem can be derived [Bulatov et al. 2005].

PROPOSITION 8 [SCHAEFER 1978; BULATOV ET AL. 2005]. *A* 2*-element relational structure $\mathcal{H}$ is tractable if and only if $\mathbb{A}_\mathcal{H}$ is not a G-set. Otherwise $\mathcal{H}$ is NP-complete.*

2.3. ALGEBRAIC CONSTRUCTIONS AND THE COMPLEXITY OF CONSTRAINT LANGUAGES. The main advantage of using algebras instead of constraint languages and even polymorphisms is that an algebra $\mathbb{A}$ usually has a collection of related algebras, whose complexity is closely connected to the complexity of $\mathbb{A}$, and thus they can be used to study the complexity of $\mathbb{A}$. This is our principal method in this article. In this section, we define the required related algebras and state the required connection between their complexity. However, sometimes this connection does not hold for an arbitrary algebra, therefore, we first introduce some transformations of relational structures and constraint languages that preserve the complexity of the associated problems.

An *induced substructure* of a structure $\mathcal{H}$ with universe H over a vocabulary $R_1, \ldots, R_n$ is a structure $\mathcal{G}$ with universe $G \subseteq H$ and relations $R_i^\mathcal{G}$ such that if R_i is, say, m-ary, then $R_i^\mathcal{G} = R_i^\mathcal{H} \cap G^m$. A *core* of $\mathcal{H}$ is an induced substructure $\mathcal{H}'$ such that there is a homomorphism φ from $\mathcal{H}$ to $\mathcal{H}'$ but there is no homomorphism to a proper substructure of $\mathcal{H}'$. It is well known that φ can be assumed satisfying the condition $\varphi^2 = \varphi$, and that any two cores of a structure are isomorphic. Therefore, we can define the core, $\mathsf{core}\,\mathcal{H}$, of a relational structure. A structure $\mathcal{H}$ is said to be a *core* if $\mathsf{core}\,\mathcal{H} = \mathcal{H}$. We use $\mathcal{H}^+$ to denote the structure $(H; \{R_1^\mathcal{H}, \ldots, R_n^\mathcal{H}\} \cup \{R_a^\mathcal{H} \mid a \in H\})$ where $R_a^\mathcal{H} = \{(a)\}$.

PROPOSITION 9 [BULATOV ET AL. 2005]. *Let $\mathcal{H}$ be a relational structure.*

—*$\mathcal{H}$ is tractable* [*NP-complete*] *if and only if* $\mathsf{core}\,\mathcal{H}$ *is tractable* [*NP-complete*].
—*If $\mathcal{H}$ is a core then $\mathcal{H}$ is tractable* [*NP-complete*] *if and only if $\mathcal{H}^+$ is tractable* [*NP-complete*].

Similar transformations of constraint languages can be defined as follows: Let Γ be a constraint language on A. A *core* of Γ is the language $g(\Gamma) = \{g(R) \mid R \in \Gamma\}$ where $g(R) = \{(g(a_1), \dots, g(a_n)) \mid (a_1, \dots, a_n) \in R\}$, and g is a unary polymorphism of Γ such that $g(g(x)) = g(x)$ and the range of g is minimal under inclusion. If $g_1(\Gamma), g_2(\Gamma)$ are cores, then $g_1(\Gamma) = g_1(g_2(\Gamma))$ and $g_2(\Gamma) = g_2(g_1(\Gamma))$. Therefore, any two cores of Γ are in some sense isomorphic, and we can define the core, $\mathsf{core}\,\Gamma$, of a language Γ. We use Γ^+ to denote the language $\Gamma \cup \{\{(a)\} \mid a \in A\}$, and we use Γ^{id} to denote the language $(\mathsf{core}\,\Gamma)^+$.

COROLLARY 3 [BULATOV ET AL. 2005]. *Let Γ be a constraint language on A. Then Γ is tractable* [*NP-complete*] *if and only if Γ^{id} is tractable* [*NP-complete*].

An operation f on a set A is said to be *idempotent* if $f(x, \dots, x) = x$ for any $x \in A$. For any constraint language Γ, the algebra $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ is idempotent, that is, all its basic operations are idempotent.

Due to Proposition 2 and Corollary 3, the study of the complexity of constraint languages is completely reduced to the study of properties of idempotent algebras.

Let us consider the following example.

Example 3. Let $A = \{0, 1, 2\}$ and

$$S = \begin{pmatrix} 0&1&1&1&1&0&0&0&0&2&2&2&2&0&0&0&0&1&1&1&0&2&2&2 \\ 1&0&1&1&0&1&0&0&2&0&2&2&0&2&0&0&1&0&2&2&2&0&1&1 \\ 1&1&0&1&0&0&1&0&2&2&0&2&0&0&2&0&2&2&0&1&1&1&0&2 \\ 1&1&1&0&0&0&0&1&2&2&2&0&0&0&0&2&1&1&1&0&2&2&2&0 \end{pmatrix}.$$

The constraint language $\Gamma = \{S\}$ has two polymorphisms g and h satisfying the conditions above: g maps 0 to 0 and 1, 2 to 1, and h maps 0 to 0 and 1, 2 to 2. As is easily seen, $g(S) = R$, where R is the relation from Example 1. Therefore, $\Gamma^{\mathrm{id}} = \{R\} \cup \{\{(0)\}, \{(1)\}\}$, and the corresponding idempotent algebra is $(\{0, 1\}; F)$ where the members of F are operations of the form $x_1 \oplus \cdots \oplus x_n$, n odd.

We are now making the last step towards the dichotomy conjecture for arbitrary constraint languages and the dichotomy theorem for constraint languages on a 3-element set. We need several standard algebraic constructions.

Let $\mathbb{A} = (A, F)$ be an algebra, and B a subset of A such that, for any $f \in F$ (n-ary), and for any $b_1, \dots, b_n \in B$, we have $f(b_1, \dots, b_n) \in B$. Then the algebra $\mathbb{B} = (B, F|_B)$, where $F|_B$ consists of restrictions of operations from F to B, is called a *subalgebra* of $\mathbb{A}$. The universe of a subalgebra of $\mathbb{A}$ is called a *subuniverse* of $\mathbb{A}$. A subalgebra $\mathbb{B}$ (a subuniverse B) is said to be *proper* if $\mathbb{B} \neq \mathbb{A}$ ($B \neq A$).

Let $\mathbb{A} = (A, F)$ be an algebra. An equivalence relation $\theta \in \mathsf{Inv}\ F$ is said to be a *congruence* of $\mathbb{A}$. The θ-class containing $a \in A$ is denoted a^θ, the set $A/_\theta = \{a^\theta \mid a \in A\}$ is said to be the *quotient-set*, and the algebra $\mathbb{A}/_\theta = (A/_\theta; F^\theta)$, $F^\theta = \{f^\theta \mid f \in F\}$ where $f^\theta(a_1^\theta, \dots, a_n^\theta) = (f(a_1, \dots, a_n))^\theta$, is said to be the *quotient-algebra*.

PROPOSITION 10 [BULATOV ET AL. 2005]. *Let Γ be a tractable constraint language on A, a set $B \subseteq A$ a subuniverse of $\mathbb{A}_\Gamma$, and θ a congruence of $\mathbb{A}_\Gamma$. Then*

(1) *the subalgebra $\mathbb{B} = (B; (\mathsf{Pol}\ \Gamma)|_B)$ of $\mathbb{A}_\Gamma$ is tractable;*

(2) *$\mathbb{A}_\Gamma/_\theta$ and the set $\Gamma^\theta = \{R^\theta \mid R \in \Gamma\}$, where $R^\theta = \{(a_1^\theta, \ldots, a_n^\theta) \mid (a_1, \ldots, a_n) \in R\}$, are tractable.*

Hence, every subalgebra and every quotient-algebra of a tractable algebra is tractable. Furthermore, every *factor* of a tractable algebra, that is, a quotient-algebra of a subalgebra, is tractable and cannot be a G-set. Thus, every tractable algebra $\mathbb{A}$ satisfies the condition

$$\text{none of the factors of } \mathbb{A} \text{ is a } G\text{-set.} \qquad (\text{NO-G-SET})$$

Moreover, all known examples of NP-complete subclasses of the CSP have a G-set behind. We therefore arrive to the following conjecture.

CONJECTURE [BULATOV ET AL. 2005]. A constraint language Γ on a finite set A is tractable if and only if $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ satisfies the condition (NO-G-SET). Otherwise, it is NP-complete.

2.4. OUR RESULTS. By Proposition 8, the conjecture from the previous section holds for constraint languages on a 2-element set. The main result of this paper is just that the conjecture holds for constraint languages on a 3-element set.

THEOREM 1. *A constraint language Γ on a 3-element set is tractable if and only if the algebra $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ satisfies the condition* (NO-G-SET). *Otherwise, Γ is NP-complete.*

We also show that every nonuniform homomorphism problem for a 3-element structure can be uniformized.

THEOREM 2. *Let A be a 3-element set, and $\mathfrak{T}$ the class of all tractable structures with the universe A. The uniform problem $\mathsf{HOM}(\mathfrak{T})$ is solvable in polynomial time.*

The NP-completeness part of Theorem 1 follows from the results of Section 2.3, while the tractability part follows from Theorem 3 and the results of Section 4.3. Theorem 2 is proved in Section 5.

3. *Tractable Constraint Languages on a 3-Element Set*

In this section, we define 10 properties such that every 3-element tractable algebra satisfies one of them, and state the main technical theorem. In Section 4, for each of the 10 properties, we construct an algorithm using this property to solve constraint satisfaction problems. The theorem then will be proved in in Section 6 by an elaborate case analysis.

From now on, we consider problems of the form CSP($\mathbb{A}$), where $\mathbb{A}$ is a 3-element algebra. The necessity of the condition (NO-G-SET) for the tractability of a finite algebra follows from Propositions 7 and 10. To show that the condition is sufficient, we have to provide, for any 3-element algebra $\mathbb{A}$ satisfying (NO-G-SET), an algorithm that solves the problem CSP($\mathbb{A}$) in polynomial time. We split the proof into two parts. The first part is "algebraic"; we show that, for any 3-element algebra $\mathbb{A} = (A; F)$ satisfying (NO-G-SET), the relations from $\mathsf{Inv}\ F$ satisfy one of certain 10 properties. The second part is "algorithmic"; for each of those 10 properties, we present a polynomial-time algorithm that solves the constraint satisfaction problem rising from a set of relations satisfying this property.

Throughout this section, $\mathbb{A}$ satisfies the condition (NO-G-SET). By $\underline{n}$, for a natural number n, we will denote the set $\{1, \dots, n\}$. Let R be an n-ary relation, and $I = \{i_1, \dots, i_k\} \subseteq \underline{n}$; then, R_I denotes the k-ary relation $\{\mathbf{a}_I \mid \mathbf{a} \in R\}$ where $\mathbf{a}_I = (\mathbf{a}[i_1], \dots, \mathbf{a}[i_k])$. If $I = \{i\}$ is a singleton, then we use R_i instead of $R_{\{i\}}$. It will often be convenient for us to consider relations whose coordinate positions are indexed by elements of a certain arbitrary set, not necessarily by natural numbers. For example, the coordinate positions of constraint relations will be supposed to be indexed by variables. In the definition below, we use the following notation. For an (n-ary) relation $R \in \mathsf{Inv}\ F$ and a 2-element subuniverse B of $\mathbb{A}$, we set $U = \{i \in \underline{n} \mid R_{\{i\}} = A\}$; $W = \{i \in \underline{n} \mid B \subseteq R_{\{i\}}\}$. We denote by $\theta_B(R)$ the equivalence relation on W

$$\{(i, j) \mid \text{for any } \mathbf{a} \in R, \text{ either } \mathbf{a}[i], \mathbf{a}[j] \in B \text{ or } \mathbf{a}[i], \mathbf{a}[j] \notin B\},$$

and let $W_1, \dots, W_k$ be the classes of this equivalence relation. Recall that the *graph* of a mapping $f: A \to B$ is the binary relation $\{(a, f(a)) \mid a \in A\}$. The relation R is said to be *irreducible* if, for every pair i, j of coordinate positions, the projection $R_{\{i,j\}}$ is not the graph of a bijective mapping.

Definition 1 (*The Properties*). The algebra $\mathbb{A}$

(1) satisfies the *partial zero property* if there exist a set of its subuniverses Z and $z_B \in B$ for each $B \in Z$, such that (a) $A \in Z$; (b) for any relation $R \in \mathsf{Inv}\ F$, and any $\mathbf{a} \in R$, there is $\mathbf{b} \in R$ with

$$\mathbf{b}[i] = \begin{cases} z_B, & \text{if } R_{\{i\}} = B \in Z, \\ \mathbf{a}[i] & \text{otherwise;} \end{cases}$$

(2) satisfies the *splitting* property if any (n-ary) relation $R \in \mathsf{Inv}\ F$ can be represented in the form $R_U \times R_{\underline{n}-U}$, and $R_U = A^{|U|}$;

(3) satisfies the $(a - b)$-*replacement property*, $a \in A - B$ and $b \in B$, if, for any (n-ary) $R \in \mathsf{Inv}\ F$, and any $\mathbf{a} \in R$, there is $\mathbf{b} \in R$ with

$$\mathbf{b}[i] = \begin{cases} b, & \text{if } \mathbf{a}[i] = a \text{ and } R_{\{i\}} = A, \\ \mathbf{a}[i] & \text{otherwise;} \end{cases}$$

(4) is called B-*rectangular*, if for any relation $R \in \mathsf{Inv}\ F$,
$R_W \cap B^{|W|} = (R_{W_1} \cap B^{|W_1|}) \times \dots \times (R_{W_k} \cap B^{|W_k|})$,
and, for any $\mathbf{a} \in R$ such that $\mathbf{a}[i] \in B$ whenever $i \in W$, there is $\mathbf{b} \in R$ with

$$\mathbf{b}[i] = \begin{cases} \mathbf{a}[i], & \text{if } i \in W \text{ or } |R_{\{i\}}| = 1, \\ c & \text{otherwise, with } \{c\} = B \cap R_{\{i\}}; \end{cases}$$

(5) is said to be B-*semirectangular* if the equivalence relation η with classes B and $A - B = \{c\}$ is a congruence of $\mathbb{A}$ and, for any (n-ary) relation $R \in \mathsf{Inv}\ F$, any tuple $\mathbf{b} \in R$ and any $\mathbf{a}_j \in R_{W_j} \cap B^{|W_j|}$, $j \in \underline{k}$, R contains the tuple $\mathbf{a}$ with

$$\mathbf{a}[i] = \begin{cases} \mathbf{b}[i], & \text{if } B \not\subseteq R_{\{i\}}, \\ c, & \text{if } B \subseteq R_{\{i\}} \text{ and } \mathbf{b}[i] = c, \\ \mathbf{a}_j[i], & \text{if } i \in W_j \text{ and } \mathbf{b}[i] \in B; \end{cases}$$

(6) satisfies the B-*semisplitting* property if, for any irreducible (n-ary) relation $R \in \mathsf{Inv}\ F$, we have (i) $(R_U \cap B^{|U|}) \times R_{\underline{n}-U} \subseteq R$; and (ii) for any $i, j \in U$ and any $(a_i, a_j) \in R_{\{i,j\}} \cap B^2$, there is a tuple $\mathbf{a} \in R_U \cap B^{|U|}$ such that $\mathbf{a}[i] = a_i, \mathbf{a}[j] = a_j$,

(7) satisfies the B-*extendibility property* if, for any (n-ary) relation $R \in \mathsf{Inv}\ F$,
   (a) for any $k \in W$ and any $a \in B$, there is $\mathbf{a} \in R$ with $\mathbf{a}[i] \in B$ for all $i \in W$, and $\mathbf{a}[k] = a$;
   (b) for any $k, l \in W$ and any $(a, b) \in R_{\{k,l\}}$, there is $\mathbf{a} \in R$ with $\mathbf{a}[i] \in B$ for all $i \in W$, and $\mathbf{a}[k] = a$, $\mathbf{a}[l] = b$;
   (c) for any $\mathbf{a} \in B^{|W|}$ such that $(\mathbf{a}[i], \mathbf{a}[j]) \in R_{\{i,j\}}$, for any $i, j \in W$, there is $\mathbf{b} \in R$ such that

$$\mathbf{b}[i] = \begin{cases} \mathbf{a}[i], & \text{if } i \in W, \\ d, & \text{if } |R_{\{i\}}| = 2 \text{ and } \{d\} = R_{\{i\}} \cap B, \\ d, & \text{if } \{d\} = R_{\{i\}}. \end{cases}$$

An algebra $\mathbb{A}' = (A; F')$ is said to be a *reduct* of an algebra $\mathbb{A}$ if F' is a subset of the set of term operations of $\mathbb{A}$.

THEOREM 3. *If an idempotent* 3-*element algebra* $\mathbb{A} = (A; F)$ *satisfies the condition* (NO-G-SET), *then there a reduct* $\mathbb{A}' = (A; F')$ *of* $\mathbb{A}$ *satisfies* (NO-G-SET) *and one of the following conditions holds:*

(1) $\mathbb{A}'$ *satisfies the partial zero property;*
(2) $\mathbb{A}'$ *satisfies the splitting property;*
(3) $\mathbb{A}'$ *satisfies the* $(a - b)$-*replacement property for a* 2-*element subuniverse* B *and* $a \in A - B$, $b \in B$;
(4) $\mathbb{A}'$ *is* B-*rectangular for a* 2-*element subuniverse* B;
(5) $\mathbb{A}'$ *satisfies the* B-*semirectangular property for a* 2-*element subuniverse* B;
(6) $\mathbb{A}'$ *satisfies the* B-*semisplitting property for a* 2-*element subuniverse* B, *and* $\mathbb{B}$ *has a majority term operation;*
(7) $\mathbb{A}'$ *satisfies the* B-*extendibility property for a* 2-*element subuniverse* $B \subseteq A$;
(8) $\mathbb{A}'$ *has a majority term operation;*
(9) $\mathbb{A}'$ *has a conservative commutative term operation;*
(10) $\mathbb{A}'$ *has a Mal'tsev term operation.*

In fact, in some cases, to get one of the 10 properties satisfied, we need to add a unary relation (a subuniverse) to the constraint language corresponding to $\mathbb{A}$. This is why we take a reduct of the algebra, F' is the set of term operations preserving the extra unary relation.

Since the members of F' are term operations of $\mathbb{A}$, we have $\mathsf{Inv}\ F \subseteq \mathsf{Inv}\ F'$. Hence the tractability of $\mathbb{A}'$ implies the tractability of $\mathbb{A}$.

Theorem 3 is proved in Section 6.

4. *Algorithms*

It turns out that we need only three types of algorithms: the first one is based on finding partial solutions, the second one reduces CSP($\mathbb{A}$) to the case of a 2-element domain, and the third one resembles algorithms from linear algebra. In Section 4.1, we recall and discuss several notions of a local algorithm. Then, in Section 4.2, a generalized version of the CSP is introduced. This generalized version is very useful for designing algorithms, which are described in Section 4.3 for each of the 10 properties. Finally, the results of Section 4.3 allow us to characterize in Section 4.4 the CSPs over a 3-element set solvable by local algorithms.

4.1. PARTIAL SOLUTIONS AND BOUNDED WIDTH. Many heuristic algorithms for constraint satisfaction involve 'constraint propagation,' which can be intuitively described as the derivation of new constraints from the original ones. This process can be formalized using various *consistency* concepts that make explicit additional constraints implied by the original constraints. Many such concepts are extensively studied, see, for example, the following books [Dechter 2003; Tsang 1993] and the following papers [Cooper 1989; Dechter 1992; Dechter and van Beek 1997; Feder and Vardi 1998; Freuder 1982; Jeavons et al. 1998a; van Hentenryck et al. 1992].

We say that a constraint satisfaction problem is solvable by a *local algorithm* if a certain kind of consistency is sufficient for the existence of a solution. We use two types of consistency. Let $\mathcal{P} = (V; A; \mathcal{C})$ be a constraint satisfaction problem, and $W \subseteq V$. The *restricted* problem $\mathcal{P}_W$ is defined to be $(W; A; \mathcal{C}_W)$ where, for each $\langle s, R\rangle \in \mathcal{C}$, there is $\langle s \cap W; R_{s\cap W}\rangle$ in $\mathcal{C}_W$. A solution to $\mathcal{P}_W$ is said to be a *partial solution* of $\mathcal{P}$ on W, and the set of all partial solutions on W is denoted $\mathcal{S}_W$. Making use of partial solutions, we define three properties of constraint satisfaction problems. Let $\mathcal{P} = (V; A; \mathcal{C})$ be a constraint satisfaction problem.

(1) $\mathcal{P}$ is said to be *k-consistent* if, for any subsets $V' \subseteq V$ containing $k-1$ elements and any subset $V' \subseteq V'' \subseteq V$ containing k elements, every partial solution on V' can be extended to a partial solution on V''.
(2) $\mathcal{P}$ is said to be *strong k-consistent* if it is l-consistent for all $l \leq k$.
(3) $\mathcal{P}$ is called *k-minimal* if, for any k-element subset $W \subseteq V$, any $\langle s, R\rangle \in \mathcal{C}$ and any $\mathbf{a} \in R$, there is $\langle s', R'\rangle \in \mathcal{C}$ with $W \subseteq s'$ and, the tuple $\mathbf{a}_{s\cap W}$ is a part of a solution from $\mathcal{S}_W$.

Most of the papers listed above consider the concepts of k-consistency and strong k-consistency. Minimality is much less studied. However, in our opinion, this type of consistency is more relevant for solving CSPs than the others. This is why we discuss the concept of minimality in more detail and compare it with the other types of consistency. As we shall see later, advantages of using minimality are especially clear when solving problems CSP(Γ) for infinite Γ.

Any constraint satisfaction problem instance $\mathcal{P}$ can be modified to obtain a strong k-consistent or a k-minimal problem instance $\mathcal{P}'$ without changing the set of solutions. In order to obtain a strong k-consistent problem, for every k-element subset $W \subseteq V$, we add to $\mathcal{P}$ the constraint $\langle W, \mathcal{S}_W\rangle$. A k-minimal problem can be obtained by applying algorithm k-MINIMALITY shown in Figure 1. The output of this algorithm is said to be the *k-minimal problem associated with* $\mathcal{P}$. It is well known and easily seen that the time complexity of this algorithm is $O(n^k m^2)$, where n is the number of variables and m is the total number of tuples in the constraint relations. The algorithms for establishing k-consistency and strong k-consistency are similar, have a similar time complexity, and can be found in, for example, Dechter [2003].

A class $\mathbf{C}$ of constraint satisfaction problems is said to be of *width k* if any problem instance $\mathcal{P}$ from $\mathbf{C}$ has a solution if and only if the strong k-consistent problem associated with $\mathcal{P}$ contains no constraint with empty constraint relation. If

INPUT. A problem instance $\mathcal{P} = (V; A; \mathcal{C})$.

OUTPUT. A k-minimal problem instance $\mathcal{P}' = (V; A; \mathcal{C}')$ equivalent to $\mathcal{P}$.

Step 1 **set** $\mathcal{P}' := (V; A; \mathcal{C}')$ where
$\mathcal{C}' = \mathcal{C} \cup \{\langle W, A^k\rangle \mid W \subseteq V,\ |W| = k\}$
and **set** $\mathcal{C}'' := \mathcal{C}'$, $\mathcal{P}'' = (V; A; \mathcal{C}'')$;
Step 2 **do**
Step 3 **set** $\mathcal{P}' := \mathcal{P}''$.
Step 4 **for each** $W \subseteq V$ with $|W| = k$ **do**
Step 5 **solve** the restricted problem $\mathcal{P}'_W$.
Step 6 **for each** constraint $C = \langle s, R\rangle \in \mathcal{C}''$, **replace** C with
$\langle s, R'\rangle$ where $R' = \{\mathbf{a} \in R \mid \mathbf{a}_{s\cap W} \in (\mathcal{S}'_W)_{s\cap W}\}$.
endfor
Step 7 **until** $\mathcal{P}'' = \mathcal{P}'$.
Step 8 **output** $\mathcal{P}'$.

FIG. 1. Algorithm k-MINIMALITY.

C is of width k for a certain k, then **C** is said to be of *bounded width*. Analogously, a class **C** of constraint satisfaction problems is said to be of *relational width k* if any problem instance $\mathcal{P}$ from **C** has a solution if and only if the k-minimal problem associated with $\mathcal{P}$ contains no constraint with empty constraint relation. If **C** is of relational width k for a certain k, then **C** is said to be of *bounded relational width*.

Every class of bounded width or bounded relational width is tractable, because, assuming k fixed, establishing k-consistency and k-minimality takes polynomial time.

As is easily seen, every k-minimal problem is also strong k-consistent. Therefore, every problem of width k has also relational width k; and, hence, every problem of bounded width is of bounded relational width. The converse is true in the important case of problems of the form CSP(Γ), where Γ is finite, or, equivalently, problems of the form HOM($\mathcal{H}$). Indeed, let k be the maximal arity of relations from Γ, $\mathcal{P} \in$ CSP(Γ) a problem instance, and $\mathcal{P}'$ the associated strong k-consistent problem. Then, the constraints added to $\mathcal{P}$ to obtain $\mathcal{P}'$ subsume the original constraints from $\mathcal{P}$. Therefore, the problem $\mathcal{P}''$ obtained from $\mathcal{P}'$ by removing the constraints from $\mathcal{P}$ is equivalent to $\mathcal{P}'$, and it is equivalent to the k-minimal problem associated with $\mathcal{P}$. This is not true for infinite languages. A simple example is provided by HORN-SATISFIABILITY (case (4) of Schaefer's Dichotomy Theorem). This problem has not bounded width, although any finite fragment of the corresponding constraint language gives rise to a problem of bounded width; but, as follows from the results of Jeavons et al. [1997] and Jeavons [1998b], this problem has relational width 1.

Sometimes a problem may have bounded width just because of certain polymorphisms of the constraint language. Several examples are provided by the following

PROPOSITION 11 [JEAVONS ET AL. 1997, 1998A]. *If a constraint language* Γ *has a semilattice polymorphism then* CSP(Γ) *is of relational width 1. If it has a majority polymorphism, then* CSP(Γ) *is of relational width* 3.

4.2. MULTISORTED CONSTRAINTS SATISFACTION PROBLEM. In Bulatov and Jeavons [2001a], it was shown that a generalized version of the constraint satisfaction problem, namely the *multisorted constraint satisfaction problem*, in which every variable is allowed to have its own domain, can also be studied using algebraic approach. Here, we use corresponding definitions and results from the paper cited as an auxiliary tool, mostly for describing algorithms, so we present them very briefly.

For any collection of sets $\mathcal{A} = \{A_i \mid i \in I\}$, and any list of indices $(i_1, i_2, \dots, i_m) \in I^m$, a subset R of $A_{i_1} \times A_{i_2} \times \cdots \times A_{i_m}$, together with the list $(i_1, i_2, \dots, i_m)$, will be called a *multisorted relation* over $\mathcal{A}$ with arity m and *signature* $(i_1, i_2, \dots, i_m)$. For any such relation R, the signature of R will be denoted $\sigma(R)$.

A *multisorted constraint language* is a set of multisorted relations. Given any multisorted constraint language, we can define a corresponding class of multisorted constraint satisfaction problems, in the following way.

Definition 3 (*Multisorted Constraint Satisfaction Problem*). Let Γ be a multisorted constraint language over a collection of sets $\mathcal{A} = \{A_i \mid i \in I\}$. The *multisorted constraint satisfaction problem over* Γ, denoted by m-CSP(Γ) is defined to be the decision problem with instance $(V; \mathcal{A}; \delta; \mathcal{C})$ where

—V is a set of variables;

—δ is a mapping from V to I, called the *domain function*;

—$\mathcal{C}$ is a set of constraints, where each constraint $C \in \mathcal{C}$ is a pair $\langle s, R\rangle$, such that
  - —$s = (v_1, \dots, v_{m_C})$ is a tuple of variables of length m_C, called the *constraint scope*;
  - —R is an element of Γ with arity m_C and signature $(\delta(v_1), \dots, \delta(v_{m_C}))$, called the *constraint relation*.

The question is whether there exists a solution, that is, a function φ, from V to $\bigcup_{A\in\mathcal{A}} A$, such that, for each variable $v \in V$, $\varphi(v) \in A_{\delta(v)}$, and for each constraint $\langle s, R\rangle \in \mathcal{C}$, with $s = (v_1, \dots, v_m)$, the tuple $(\varphi(v_1), \dots, \varphi(v_m))$ belongs to R?

Tractable and NP-complete multisorted constraint languages are defined in the same way as ordinary ones.

It is possible to introduce an algebraic structure of the multisorted CSP in a very similar way to that for the usual one. We need to define a suitable extension of the notion of a polymorphism.

Let $\mathcal{A}$ be a collection of sets. An n-ary *multisorted operation* f on $\mathcal{A}$ is defined by a collection of *interpretations* $\{f^A \mid A \in \mathcal{A}\}$, where each f^A is an n-ary operation on the corresponding set A. The multisorted operation f on $\mathcal{A}$ is said to be a *polymorphism* of a multisorted relation R over $\mathcal{A}$ with signature $(\delta(1), \dots, \delta(m))$ if, for any $(a_{11}, \dots, a_{m1}), \dots, (a_{1n}, \dots, a_{mn}) \in R$, we have

$$f\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} f^{A_{\delta(1)}}(a_{11}, \dots, a_{1n}) \\ \vdots \\ f^{A_{\delta(m)}}(a_{m1}, \dots, a_{mn}) \end{pmatrix} \in R.$$

For any given set of multisorted relations Γ, the set of all those multisorted operations, which are polymorphisms of *every* relation in Γ is denoted by $\mathsf{m\text{-}Pol}\,\Gamma$.

The next theorem establishes the remarkable fact that many of the known one-sorted tractable classes listed in Proposition 3 can be combined in an almost arbitrary way to obtain new multisorted tractable classes.

Note that a multisorted operation t is said to be *idempotent* if every its interpretation t^A is an idempotent operation.

PROPOSITION 12 [BULATOV AND JEAVONS 2001A, 2003]. *Let* Γ *be a multisorted constraint language over a collection of finite sets* $\mathcal{A} = \{A_1, \ldots, A_n\}$.

If, for each $A_i \in \mathcal{A}$, m-Pol Γ *contains a multisorted operation* f_i *such that*

—$f_i^{A_i}$ *is a constant operation, or*
—$f_i^{A_i}$ *is a semilattice operation, or*
—$f_i^{A_i}$ *is a majority operation, or*
—f_i *is idempotent and* $f_i^{A_i}$ *is an affine operation,*

then m-CSP(Γ) *is tractable.*

It is also possible to use algebras in the study of multisorted constraint satisfaction problems. We note first that algebras can be grouped into families, which share the same set of basic operations.

A collection of algebras $\mathcal{A} = \{(A_i; F^{A_i}) \mid i \in I\}$ is said to be a collection of *similar* algebras if there exists some fixed set F of multisorted operations over the sets $\{A_i \mid i \in I\}$ such that each set of basic operations F^{A_i} is the set of interpretations of the functions in F on the set A_i. The set F here is called the set of basic operations of the collection $\mathcal{A}$. For any collection of similar algebras $\mathcal{A} = \{(A_i; F^{A_i}) \mid i \in I\}$ with basic operations F, m-Inv $\mathcal{A}$ is defined to be the set of all multisorted relations R over the sets $\{A_i \mid i \in I\}$ such that $F \subseteq$ m-Pol R.

A collection of algebras $\mathcal{A}$ will be called *tractable* if the set of multisorted relations m-Inv $\mathcal{A}$ is tractable. Similarly, $\mathcal{A}$ will be called *NP-complete* if m-Inv $\mathcal{A}$ is NP-complete.

PROPOSITION 13 [BULATOV AND JEAVONS 2001A]. *Let* $\mathbb{A}_1, \ldots, \mathbb{A}_l, \mathbb{B}_1, \ldots, \mathbb{B}_n$, *be similar finite algebras, each* $\mathbb{B}_i$ *have either a constant or semilattice or majority term operation, and let* $\mathbb{A}_1, \ldots, \mathbb{A}_l$ *have affine term operations whose interpretations on other algebras are idempotent. Then* $\{\mathbb{A}_1, \ldots, \mathbb{A}_l, \mathbb{B}_1, \ldots, \mathbb{B}_n\}$ *is tractable.*

PROPOSITION 14 [BULATOV AND JEAVONS 2001A]. *Let* $\mathbb{A}_1, \ldots, \mathbb{A}_l$ *be similar finite algebras and every* $\mathbb{A}_i$ *have either a constant or semilattice or majority term operation. Then* m-CSP($\{\mathbb{A}_1, \ldots, \mathbb{A}_l, \mathbb{B}_1, \ldots, \mathbb{B}_n\}$) *is of relational width* 3.

A class of similar algebras naturally arises when we consider the collection of all factors of an algebra. This is the way in which multisorted constraint satisfaction problems appear in this article. We say that a problem instance $\mathcal{P} = (V; A; \mathcal{C}) \in$ CSP($\mathbb{A}$) where $\mathbb{A}$ is a 3-element algebra is *2-valued* if, for any $v \in V$, $\mathcal{S}_{\{v\}}$ contains at most 2 elements. Such a problem can be treated as a multisorted problem over the family of all proper subalgebras of $\mathbb{A}$. If $\mathbb{A}$ satisfies the condition (NO-G-SET), then by the results of Post [1941], every 2-element subalgebra of $\mathbb{A}$ has a term operation which is either constant or semilattice or majority or affine. If $\mathbb{A}$ is idempotent, then the collection of its subalgebras satisfies the conditions of Proposition 12. Therefore, $\mathcal{P}$ can be solved in polynomial time.

COROLLARY 4. *If a* 3*-element idempotent algebra satisfies the condition* (NO-G-SET)*, then any* 2*-valued problem instance from* CSP($\mathbb{A}$) *can be solved in polynomial time.*

Most of the "good" properties of relations from Theorem 3 allows us, first, to reduce an arbitrary problem instance to a 2-valid problem instance, and, second, to solve the obtained instance as a multisorted problem instance by making use of the algorithms from Bulatov and Jeavons [2001a].

4.3. WHY 'GOOD' PROPERTIES ARE GOOD. In this section, we show that, for each of the 10 properties, every algebra satisfying this property is tractable and provide algorithms solving the corresponding CSPs.

4.3.1. *Relations Invariant with Respect to a Special Operation.* In (8), the tractability of $\mathbb{A}$ follows from Proposition 6. As is shown in Jeavons et al. [1998a], CSP($\mathbb{A}$) is of width 3 in this case. In (9), CSP($\mathbb{A}$) is of width 3 that was proved in Bulatov and Jeavons [2000] and Bulatov [2006a]. The result of Bulatov [2002] states that any finite algebra with a Mal'tsev term operation is tractable, and the solution algorithm in this case is similar to algorithms from linear algebra.

4.3.2. *The Partial Zero Property.* In this case, any problem instance can be reduced to a 2-valued one. To show this, we observe that if $\mathbb{A}$ satisfies the partial zero property, and a 1-minimal problem instance $\mathcal{P} = (V; A; \mathcal{C}) \in$ CSP($\mathbb{A}$) has a solution φ, then $\mathcal{P}$ also has the solution ψ such that $\psi(v) = \varphi(v)$ if $\mathcal{S}_{\{v\}} \notin Z$, and $\psi(v) = z_{\mathcal{S}_{\{v\}}}$ otherwise. Indeed, for any constraint $\langle s, R\rangle \in \mathcal{C}$ with $s = (v_1, \dots, v_k)$, we have $\mathbf{a} = (\varphi(v_1, \dots, \varphi(v_k)) \in R$, the tuple $\mathbf{b} = (\psi(v_1), \dots, \psi(v_k))$ is obtained from $\mathbf{a}$ as in Definition 1(1) and therefore $\mathbf{b} \in R$.

Thus, to solve $\mathcal{P}$, we make the problem $\mathcal{P}' = (V; A; \mathcal{C}')$, where, for every $\langle s, R\rangle \in \mathcal{C}$, we include into $\mathcal{C}'$ the constraint $\langle s, R'\rangle$ with $R' = \{\mathbf{a} \in R \mid \mathbf{a}[v] = z_{\mathcal{S}_v}$ for every v such that $\mathcal{S}_v \in Z\}$. Since $A \in Z$, the obtained problem is 2-valued and, therefore, it can be solved in polynomial time.

Notice also that if every 2-element subalgebra of $\mathbb{A}$ has a term semilattice or majority operation, then CSP($\mathbb{A}$) is of bounded relational width.

4.3.3. *The Replacement Property.* In this case, any problem instance can also be reduced to a 2-valued one. If $\mathbb{A}$ satisfies the $(a-b)$-replacement property, and a 1-minimal problem instance $\mathcal{P} = (V; A; \mathcal{C}) \in$ CSP($\mathbb{A}$) has a solution φ, then the mapping $\psi: V \to A$ such that $\psi(v) = b$ if $a, b \in \mathcal{S}_{\{v\}}$, $\varphi(v) = a$, and $\psi(v) = \varphi(v)$ otherwise, is a solution to $\mathcal{P}$. We therefore may reduce $\mathcal{P}$ to a 2-valued problem instance $\mathcal{P}' = (V; A; \mathcal{C}')$, where for each $C = \langle s, R\rangle \in \mathcal{C}$ there is $C' = \langle s, R'\rangle \in \mathcal{C}'$ such that $\mathbf{a} \in R'$ if and only if $\mathbf{a} \in R$ and $\mathbf{a}[v] \neq a$ whenever $a, b \in \mathcal{S}_{\{v\}}$.

4.3.4. *The Extendibility Property.* We prove that, in this case, CSP($\mathbb{A}$) is of width 3. Suppose that $\mathbb{A}$ satisfies the B-extendibility property for a subuniverse B of $\mathbb{A}$, and take a 3-minimal problem instance $\mathcal{P} = (V; A; \mathcal{C})$.

LEMMA 1. *Let $W \subseteq V$ be the set $\{v \in V \mid B \subseteq \mathcal{S}_{\{v\}}\}$. There is $\mathbf{a} \in B^{|W|}$ such that, for any $v, w \in W$, we have $\binom{\mathbf{a}[v]}{\mathbf{a}[w]} \in \mathcal{S}_{\{v,w\}}$.*

PROOF. Since $\mathcal{P}$ is 3-minimal, for any $v, w \in W$, there is a constraint $\langle s, R\rangle \in \mathcal{C}$ such that $v, w \in s$. Furthermore, since $\mathbb{A}$ satisfies the B-extendibility property, for any $a \in B$, there are $b \in B$ and $\mathbf{b} \in R$ such that $\mathbf{b}[v] = a$, $\mathbf{b}[w] = b$. Therefore, $\mathcal{S}_{\{v,w\}} \cap B^2 \neq \emptyset$. Let $W' \subseteq W$ be a maximal set such that there is $\mathbf{a} \in B^{|W'|}$ with $\binom{\mathbf{a}[v]}{\mathbf{a}[w]} \in \mathcal{S}_{\{v,w\}}$ for any $v, w \in W'$. If $W' = W$, then we are done; otherwise, take $w \in W - W'$. Let $B = \{a, b\}$. The maximality of W' means that there are $u, v \in W'$ such that $\binom{\mathbf{a}_{W'}[u]}{a} \notin \mathcal{S}_{\{u,w\}}$, $\binom{\mathbf{a}_{W'}[v]}{b} \notin \mathcal{S}_{\{v,w\}}$. Since $\mathcal{P}$ is 3-minimal, there is a constraint $\langle s, R\rangle \in \mathcal{C}$ such that $u, v, w \in s$. We have $\binom{\mathbf{a}_{W'}[u]}{\mathbf{a}_{W'}[v]} \in R_{\{u,v\}}$, and, by the

INPUT. A problem instance $\mathcal{P} = (V; A; \mathcal{C})$.

OUTPUT. A problem instance $\mathcal{P}' = (V; A; \mathcal{C}')$ equivalent to $\mathcal{P}$ and such that, for every class W' of $\theta(\mathcal{P}')$, $\mathcal{P}'_{W'}$ has a solution ψ with $\psi(v) \in B$ for any $v \in W'$.

Step 1 **set** $\mathcal{P}' = (V; A; \mathcal{C}') := \mathcal{P}$
Step 2 **do**
Step 3 $\quad\mathcal{P}'' := \mathcal{P}'$
Step 4 $\quad$**for each** class W' of $\theta(\mathcal{P})$ **do**
Step 5 $\quad\quad$**make** the problem $\mathcal{P}'_{W'} = (W'; B; \mathcal{C}'_{W'})$ where, for each $\langle s, R\rangle \in \mathcal{C}$, we include the constraint $\langle s \cap W', (R_{s\cap W'}) \cap B^{|s\cap W'|}\rangle \in \mathcal{C}'_{W'}$
Step 6 $\quad\quad$**solve** the problem $\mathcal{P}'_{W'}$
Step 7 $\quad\quad$**if** $\mathcal{P}'_{W'}$ has no solution **then**
Step 8 $\quad\quad$**for each** constraint $\langle s, R\rangle \in \mathcal{C}'$, **remove** from R all the tuples $\mathbf{a}$ such that $\mathbf{a}[v] \in B$ for some $v \in s \cap W'$
$\quad$**endfor**
Step 9 $\quad\mathcal{P}' =$3-MINIMALITY$(\mathcal{P}')$
Step 10 $\quad$**compute** W for $\mathcal{P}'$ and $\theta(\mathcal{P}')$
Step 11 **until** $\mathcal{P}'' = \mathcal{P}'$
Step 12 **output** $\mathcal{P}'$

FIG. 2. Algorithm RECTANGULARITY.

B-extendibility property, there exists $\mathbf{b} \in R$ such that $\mathbf{b}[u] = \mathbf{a}_{W'}[u]$, $\mathbf{b}[v] = \mathbf{a}_{W'}[v]$ and $\mathbf{b}[w] \in B$, that contradicts the assumptions made. $\square$

Finally, the B-extendibility property of $\mathbb{A}$ implies that the mapping $\varphi: V \to A$, where

$$\varphi(v) = \begin{cases} \mathbf{a}[v], & \text{if } v \in W, \\ c, & \text{if } \{c\} = \mathcal{S}_{\{v\}} \cap B, \\ c, & \text{if } \mathcal{S}_{\{v\}} = \{c\}, \end{cases}$$

is a solution to $\mathcal{P}$.

4.3.5. *The Rectangularity and Semirectangularity Properties.* Suppose that $\mathbb{A}$ is B-rectangular or B-semirectangular, and $\{c\} = A - B$. We show that any problem instance in this case can be reduced to a 2-valued one. Take a problem instance $\mathcal{P} = (V; A; \mathcal{C})$ from CSP($\mathbb{A}$). Without loss of generality, we may assume that $\mathcal{P}$ is 3-minimal. Let W denote the set of all variables $v \in V$ with $B \subseteq \mathcal{S}_{\{v\}}$. Let $\theta(\mathcal{P})$ be the equivalence relation on W generated by (i.e., the transitive closure of) $\bigcup_{\langle s,R\rangle\in\mathcal{C}} \theta_B(R)$. Notice that, since $\mathcal{P}$ is 3-minimal, for any $\langle s, R\rangle \in \mathcal{C}$, any $u, v \in s \cap W$ such that $(u, v) \in \theta(\mathcal{P})$, and any $\mathbf{a} \in R$, either $\mathbf{a}[u], \mathbf{a}[v] \in B$, or $\mathbf{a}[u] = \mathbf{a}[v] = c$. Then, we apply to $\mathcal{P}$ the algorithm shown in Figure 2. Obviously, the obtained problem instance $\mathcal{P}'$ has a solution if and only if the original problem instance has and, for every class W' of $\theta(\mathcal{P}')$, $\mathcal{P}'_{W'}$ has a solution ψ with $\psi(v) \in B$ for any $v \in W'$.

Suppose first that $\mathbb{A}$ is B-rectangular. Then, if $\mathcal{P}'$ has no empty constraint, then there is a solution φ to $\mathcal{P}'$ such that $\varphi(v) \in B$ whenever $B \subseteq \mathcal{S}_{\{v\}}$. Indeed, let $W_1, \ldots, W_k$ be the classes of $\theta(\mathcal{P}')$, and ψ_i a solution to $\mathcal{P}'_{W_i}$, $i \in \{1, \ldots, k\}$. It follows straightforwardly from the B-rectangularity that the mapping $\varphi: V \to A$, where

$$\varphi(v) = \begin{cases} \psi_i(v), & \text{if } v \in W_i, \\ a, & \text{if } \mathcal{S}_{\{v\}} = \{a, c\}, a \in B, \\ b, & \text{if } \mathcal{S}_{\{v\}} = \{b\}, b \in A, \end{cases}$$

is a solution to $\mathcal{P}'$.

Now, suppose that $\mathbb{A}$ satisfies the B-semirectangular property. Denote by $\mathcal{P}^\eta$ the *quotient-problem*, that is, the multisorted problem $(V; \{B_0 = A/_\eta\} \cup \{B_i \mid i \in I\}; \delta; \mathcal{C}^\eta)$, where

—B_i, $i \in I$, are the 2-element subuniverses of $\mathbb{A}$;
—$\delta(v) = i$ if and only if $\mathcal{S}_{\{v\}} = B_i$, and $\delta(v) = 0$ if $\mathcal{S}_{\{v\}} = A$;
—for each $\langle s, R\rangle \in \mathcal{C}'$, there is $\langle s, R^\eta\rangle \in \mathcal{C}^\eta$, where $\mathbf{b} \in R^\eta$ if and only if there is $\mathbf{a} \in R$ such that

$$\mathbf{b}[v] = \begin{cases} \mathbf{a}[v], & \text{if } R_v \neq A, \\ (\mathbf{a}[v])^\eta, & \text{if } R_v = A. \end{cases}$$

It is not hard to see that if $\mathcal{P}'$ has a solution, then $\mathcal{P}^\eta$ has a solution (see also Bulatov and Jeavons [2000]). Let φ be a solution to $\mathcal{P}^\eta$, and $\psi_1, \ldots, \psi_k$ solutions to $\mathcal{P}'_{W_1}, \ldots, \mathcal{P}'_{W_k}$. Notice that the variables $v \in V$ such that $\mathcal{S}_{\{v\}} = B$ constitute one of the classes $W_1, \ldots, W_k$. Let it be W_k (it may be empty). The mapping ψ, where

$$\psi[v] = \begin{cases} \varphi(v), & \text{if } v \notin W, \\ \psi_k(v), & \text{if } v \in W_k, \\ \psi_i(v), & \text{if } v \in W_i, i \neq k, \text{ and } \varphi(v) \neq c^\eta, \\ c, & \text{if } v \in W_i, i \neq k, \text{ and } \varphi(v) = c^\eta, \end{cases}$$

is a solution to $\mathcal{P}'$. Indeed, take a constraint $\langle s, R\rangle \in \mathcal{C}'$, $s = (v_1, \ldots, v_m)$. Since for each $i \in \underline{k}$ such that $\varphi(v) = B$, $v \in W_i$, ψ_i is a solution to $\mathcal{P}'_{W_i}$, the tuple $(\psi(v_{i_1}), \ldots, \psi(v_{i_k}))$, where $s \cap W = (v_{i_1}, \ldots, v_{i_k})$, belongs to $R_{s\cap W_i}$. Moreover, φ is a solution to the quotient-problem; therefore, there is $\mathbf{b} \in R$ such that $\mathbf{b}[v] = \varphi(v)$ when $v \in s - W$, $\mathbf{b}[v] = c$ when $\varphi(v) = c^\eta$ and $\mathbf{b}[v] \in B$ when $\varphi(v) = B$. The semirectangularity of $\mathbb{A}$ implies that $(\psi(v_1), \ldots, \psi(v_m)) \in R$.

Finally, the quotient-problem is 2-valued and, by Corollary 4, can be solved in polynomial time.

4.3.6. *The Splitting and Semisplitting Properties.* If $\mathbb{A}$ satisfies the splitting property, then, for any 1-minimal problem instance $\mathcal{P} = (V; A; \mathcal{C})$, let U denote the set $\{v \in V \mid \mathcal{S}_{\{v\}} = A\}$ and $U' = V - U$. As can be easily seen, $\mathcal{P}_{U'}$ is 2-valued and any solution to $\mathcal{P}_{U'}$ can be arbitrarily extended to a solution to $\mathcal{P}$.

Suppose that B is a 2-element subuniverse of $\mathbb{A}$, $\mathbb{A}$ satisfies the B-semisplitting property and there is a term operation f such that $f|_B$ is the majority operation. A problem instance is said to be *irreducible* if every its constraint relation is irreducible. Every 3-minimal problem instance $\mathcal{P} = (V; A; \mathcal{C})$ can be reduced to an equivalent irreducible problem instance in polynomial time.

In order to do this, denote by ζ the binary relation on V such that $(u, v) \in \zeta$ if and only if $\mathcal{S}_{\{u,v\}}$ is the graph of a bijective mapping $\pi_{u,v}\colon \mathcal{S}_{\{u\}} \to \mathcal{S}_{\{v\}}$. Since $\mathcal{P}$ is 3-minimal, $\mathcal{S}_{\{u,v\}} \circ \mathcal{S}_{\{v,w\}} \supseteq \mathcal{S}_{\{u,w\}}$ for any $u, v, w \in V$, where $\circ$ denotes the multiplication of binary relations; hence ζ is an equivalence relation. Choose a representative from each class of ζ, and let W denote the set of the representatives.

INPUT. A problem instance $\mathcal{P} = (V; A; \mathcal{C})$.

OUTPUT. An irreducible problem instance $\mathcal{P}' = (V; A; \mathcal{C}')$ equivalent to $\mathcal{P}$.

Step 1 **for each** constraint $\langle s, R\rangle \in \mathcal{C}$ and any $\mathbf{a} \in R$ **do**
Step 2 **replace a** with **b** where $\mathbf{b}[v] = \pi_{v,v_\zeta}(\mathbf{a}[v])$, $v \in s$
Step 3 **replace** every $v \in s$ with v_ζ
Step 4 by omitting repetitions of entries in s we obtain its subsequence, s'; **replace** $\langle s, R\rangle \in \mathcal{C}$ with $\langle s', R_{s'}\rangle$
endfor
Step 5 **output**($\mathcal{P}$)

FIG. 3. Algorithm IRREDUCIBILITY.

Then, for no pair $v, w \in W$ of variables, $\mathcal{S}_{\{v,w\}}$ is the graph of a bijective mapping; and, for any $v \in V - W$, there is $v_\zeta \in W$ such that $\mathcal{S}_{\{v_\zeta,v\}}$ is the graph of a bijective mapping. Then, we use algorithm IRREDUCIBILITY, see Figure 3.

Now, let $\mathcal{P} = (V; A; \mathcal{C}) \in \mathrm{CSP}(\mathbb{A})$ be a 3-minimal irreducible problem instance, and consider the instance $\mathcal{P}' = (V; A; \mathcal{C}')$, where for each $\langle s, R\rangle \in \mathcal{C}$ there is $\langle s, R'\rangle \in \mathcal{C}'$ with $R' = \{\mathbf{a} \in R \mid \mathbf{a}[v] \in B$ for all $v \in s$ such that $R_v = A\}$. Since the problem instance $\mathcal{P}'$ is 2-valued, the following lemma proves that $\mathcal{P}$ can be solved in polynomial time.

LEMMA 2. *$\mathcal{P}$ and $\mathcal{P}'$ are equivalent.*

PROOF. Clearly, if $\mathcal{P}'$ has a solution, then $\mathcal{P}$ has a solution. Conversely, let $\mathcal{P}$ have a solution, and set $U = \{v \in V \mid \mathcal{S}_{\{v\}} = A\}$ and $U' = V - U$. By condition (i) from the definition of the semisplitting property, $\mathcal{P}'$ has a solution if and only if the problem $\mathcal{P}'_U$ that is $\mathcal{P}'$ restricted to U, and the problem $\mathcal{P}'_{U'}$ that is $\mathcal{P}'$ restricted to U', both have solutions. Since $\mathcal{P}'_{U'} = \mathcal{P}_{U'}$, the instance $\mathcal{P}'_{U'}$ has a solution. By condition (ii), for any $v, w \in U$, we have $\mathcal{S}'_{\{v,w\}} = \mathcal{S}_{\{v,w\}} \cap B^2$, where $\mathcal{S}'_{\{v,w\}}$ denotes the set of partial solutions to $\mathcal{P}'_U$ for $\{v, w\}$. Moreover, as $\mathcal{P}'$ is 3-minimal, for any $u \in U$ every such partial solution can be extended to a solution from $\mathcal{S}'_{\{v,w,u\}}$. Therefore, $\mathcal{P}'_U$ is strong 3-consistent. Finally, since $\mathcal{P}'_U$ is a problem over B and the subalgebra of $\mathbb{A}$ with the universe B has a majority term operation, by Proposition 11, it has a solution. □

4.4. PROBLEMS OF BOUNDED WIDTH. The results of the previous subsection allow us to characterize the constraint satisfaction problems over a 3-element set solvable by local algorithms, in particular, those of bounded width and bounded relational width. Analyzing thoroughly the 10 properties we are able to characterize those constraint satisfaction problems over a 3-element domain, which have bounded (relational) width.

THEOREM 4. *Let Γ be a tractable constraint language over a 3-element set A.*

(1) *If Γ has a nonsurjective unary polymorphism, then* CSP(Γ) *is of bounded relational width if and only if there is a unary polymorphism f with $f(f(x)) = f(x)$ of Γ such that either $|f(A)| = 1$ or $|f(A)| = 2$ and there exist $g \in \mathsf{Pol}\ (\Gamma)|'_{f(A)}$ which is a semilattice or majority operation.*
(2) *If every unary polymorphism of Γ is surjective, then* CSP(Γ) *is of bounded relational width if and only if $\mathbb{A}_\Gamma$ itself or every 2-element factor of $\mathbb{A}_\Gamma$ has a semilattice or majority or binary commutative conservative term operation.*

If CSP(Γ) *is of bounded relational width, then it is of relational width* 3.

COROLLARY 5. *Let $\mathcal{H}$ be a tractable 3-element relational structure with the universe H.*

(1) *If $\mathcal{H}$ has a nonsurjective unary polymorphism, then $\mathsf{HOM}(\mathcal{H})$ is of bounded width if and only if there is a unary polymorphism f with $f(f(x)) = f(x)$ of $\mathcal{H}$ such that either $|f(H)| = 1$ or $|f(H)| = 2$ and there exist $g \in \mathsf{Pol}\,(\mathcal{H})|'_{f(A)}$ which is a semilattice or majority operation.*

(2) *If every unary polymorphism of $\mathcal{H}$ is surjective, then $\mathsf{HOM}(\mathcal{H})$ is of bounded width if and only if $\mathbb{A}_{\mathcal{H}}$ itself or every 2-element factor of $\mathbb{A}_{\mathcal{H}}$ has a semilattice or majority or binary commutative conservative operation.*

PROOF OF THEOREM 4. As a benchmark problem which is not of bounded (relational) width we use the problem of solving a system of linear equations over a 2- or 3-element field. Thus, if we show that the problem CSP($\mathsf{Inv}\ \{x - y + z\}$), where $x - y + z$ is the affine operation of such a field, can be reduced to CSP(Γ), then CSP(Γ) is not of bounded (relational) width.

(1) If there is a polymorphism f of Γ with $|f(A)| = 1$, then f is a constant operation and every instance from CSP(Γ) has a solution. If there is a polymorphism f of Γ with $|f(A)| = 2$ and $f(f(A)) = f(A)$, then by Proposition 4.4 [Jeavons 1998b] any instance $\mathcal{P} = (V; A; \mathcal{C})$ is equivalent to the instance $\mathcal{P}' = (V; f(A); \mathcal{C}')$, where $\mathcal{C}'$ is obtained from $\mathcal{C}$ by replacing every constraint $\langle s, R\rangle$ with $\langle s; f(R)\rangle$. As is easily seen, if $\mathcal{P}$ is 3-minimal, then $\mathcal{P}'$ is 3-minimal as well, and $\mathcal{P}'$ contains a constraint with empty constraint relation if and only if $\mathcal{P}$ does. If $f(\Gamma)$ has a semilattice or majority polymorphism, then, by Proposition 11, if $\mathcal{P}'$ has no empty constraint relation, then it has a solution. Since f is a polymorphism, $f(R) \subseteq R$ for every constraint relation R. Therefore $\mathcal{P}$ has a solution.

If $f(\Gamma)$ has no semilattice or majority polymorphism, then, by (1), CSP($\mathsf{Inv}\ \{x - y + z\}$), where $x - y + z$ is the affine operation of a 2-element field, is reducible to CSP($f(\Gamma)$).

(2) Notice first that the algebra $\mathbb{A}'$ in Theorem 3 can be chosen to be $\mathbb{A}$ unless it satisfies conditions considered in Section 6.6. However, even in that case if $\mathbb{A}$ itself or each of its 2-element factor has a semilattice or majority term operation, then $\mathbb{A}'$ or each of its 2-element factor has a semilattice or majority term operation. Therefore, we may assume that $\mathbb{A}_\Gamma$ satisfies one of the 10 properties.

If $\mathbb{A}_\Gamma$ has a majority or commutative conservative term operation then CSP(Γ) is of relational width 3 by Proposition 11 and Theorem 24 from Bulatov and Jeavons [2000]. If $\mathbb{A}_\Gamma$ has a Mal'tsev term operation and each of its 2-element factor has a semilattice or majority term operation, then, by Corollary 3.5 from Bulatov [2006b], one of the following ternary operations g and h is a term operation of $\mathbb{A}_\Gamma$ and therefore a polymorphism of Γ:

$$
\begin{array}{r|ccc}
g(0,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 \\
2 & 0 & 0 & 2
\end{array},
\quad
\begin{array}{r|ccc}
g(1,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
2 & 1 & 1 & 2
\end{array},
\quad
\begin{array}{r|ccc}
g(2,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 2 & 2 \\
1 & 2 & 1 & 2 \\
2 & 2 & 2 & 2
\end{array},
$$

$$
\begin{array}{r|ccc}
h(0,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 \\
2 & 0 & 0 & 2
\end{array},
\quad
\begin{array}{r|ccc}
h(1,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
2 & 1 & 1 & 2
\end{array},
\quad
\begin{array}{r|ccc}
h(2,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 0 & 2 \\
1 & 1 & 1 & 2 \\
2 & 2 & 2 & 2
\end{array}
$$

As is easily seen, both operations are majority, therefore CSP(Γ) is of relational width 3 by Proposition 11.

If $\mathbb{A}_\Gamma$ satisfies the partial zero property, then CSP(Γ) is of width 3, as is observed in Section 4.3.2.

If $\mathbb{A}_\Gamma$ satisfies the replacement property, then every 3-minimal problem instance can be reduced to a 2-valid problem instance, which is also 3-minimal. By the assumption made, for every domain B of this 2-valid instance, there is a term operation f such that $f|_B$ is a semilattice or majority operation. By Proposition 11, the problem instance has a solution.

If $\mathbb{A}_\Gamma$ satisfies the extendibility property, then CSP(Γ) is of relational width 3, as is shown in Section 4.3.4.

Let $\mathbb{A}_\Gamma$ satisfy the rectangularity or semirectangularity property. We use the notation from Section 4.3.5. By the assumption made, there is a term operation f of $\mathbb{A}_\Gamma$ such that $f|_B$ is a semilattice or majority operation. Therefore, if $\mathcal{P}$ is a 3-minimal problem instance, then every problem instance of the form $\mathcal{P}'_{W_i}$ has a solution. Hence, the instance $\mathcal{P}'$, the output of the algorithm RECTANGULARITY, is equal to $\mathcal{P}$. Furthermore, if $\mathbb{A}_\Gamma$ is rectangular, then the assignment φ constructed in Section 4.3.5 is a solution to $\mathcal{P}$.

If $\mathbb{A}_\Gamma$ is semirectangular, then the quotient-problem instance $\mathcal{P}^\theta$ is also 3-minimal. Since every domain of the quotient-problem is a factor of $\mathbb{A}_\Gamma$, Proposition 14 implies that the quotient-problem has a solution. As we have already observed, every problem instance of the form $\mathcal{P}_{W_i}$ has a solution. Therefore, the mapping ψ defined in Section 4.3.5 is a solution to $\mathcal{P}$.

If $\mathbb{A}_\Gamma$ satisfies the splitting property and $\mathcal{P}$ is 3-minimal, then $\mathcal{P}_{U'}$ has a solution by Proposition 14, while $\mathcal{P}_U$ has one by an obvious reason.

Finally, let $\mathbb{A}_\Gamma$ satisfies the semisplitting property. Observe that if a problem instance $\mathcal{P} \in$ CSP(Γ) is 3-minimal and contains no empty constraints, then so does the outcome of the algorithm IRREDUCIBILITY applied to $\mathcal{P}$. Moreover, as is noticed in the proof of Lemma 2 the 2-valued problem instance $\mathcal{P}'$ constructed in Section 4.3.6 is 2-minimal and strong 3-consistent. By Proposition 14, $\mathcal{P}'$ and, therefore, $\mathcal{P}$ has a solution.

Conversely, if there is a 2-element factor $\mathbb{B}$ of $\mathbb{A}$ without a semilattice or majority term operation, then, by Proposition 1, CSP($\mathbb{B}$) $=$ CSP($\mathsf{Inv}\ \{x - y + z\}$), where $x - y + z$ is the affine operation of a 2-element field; and, by Proposition 10, CSP($\mathbb{B}$) is polynomial time reducible to CSP(Γ). If $\mathbb{A}$ has no 2-element factor, then $\mathbb{A}$ is strictly simple (see Section 5 and Section 6.3). Since $\mathbb{A}_\Gamma$ has no semilattice, majority or binary commutative conservative term operation, by Proposition 15, every its term operation is a term operation of a 3-element Abelian group. Therefore, CSP($\mathsf{Inv}\ \{x - y + z\}$) is polynomial time reducible to CSP(Γ). $\square$

5. *Recognizing Tractable Cases*

In a practical perspective, we need a method that allows us to recognize whether a given constraint language Γ is tractable. In this section, we provide such a method for constraint languages on a 3-element set. This method is also the first step towards a polynomial time algorithm solving the uniform problem. Such an algorithm will be outlined in Section 7.

We pose the question from Meta-Problem as a combinatorial problem and investigate its complexity.

TRACTABLE-LANGUAGE PROBLEM. Is a given finite constraint language Γ on a finite set tractable?

We assume that the constraint language is given explicitly, that is as a list of relations, each of which is represented as a list of tuples.

Schaefer's Dichotomy Theorem [Schaefer 1978] does not solve this problem satisfactorily. Indeed, it can be easily verified whether a relation is of type (1) or (2); however, the way of recognising the types (3)–(6) is not obvious (see also Kolaitis and Vardi [2000a]). Proposition 8, the algebraic version of Schaefer's result, fills this gap: to check the tractability of a Boolean constraint language, one just has to verify whether all relations from the language are invariant under one of the six Boolean operations.

In the general case, such a method can hopefully be derived from a description of tractable algebras. For example, in Bulatov and Jeavons [2001b], a polynomial time algorithm has been exhibited that checks whether a finite algebra, whose basic operations are given explicitly by their operation tables, satisfies the condition (NO-G-SET). Therefore, if Conjecture holds, then the tractability of an algebra can be tested in polynomial time. In particular, this algorithm works in the case of 3-element algebras.

However, the algorithm does not solve TRACTABLE-LANGUAGE problem even under the assumption of Conjecture, because in this problem we are given a constraint language, not an algebra. Actually, we need to solve the problem

NO-G-SET-LANGUAGE PROBLEM. Given a finite constraint language Γ on a finite set, does the algebra $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ satisfy (NO-G-SET)?

By Theorem 5.4(3) from Bulatov and Jeavons [2001b], this problem is NP-complete. However, its restricted version remains tractable.

NO-G-SET-LANGUAGE(k) PROBLEM. Given a finite constraint language Γ on a finite set A such that $|A| \leq k$, does the algebra $\mathbb{A}_{\Gamma^{\mathrm{id}}}$ satisfy (NO-G-SET)?

This means that the tractability of a constraint language on a 3-element set can be tested in polynomial time.

THEOREM 5. *There is a polynomial-time algorithm that given a constraint language Γ on a 3-element set determines whether Γ is tractable.*

Theorem 5 together with Theorems 1 and 3 imply Theorem 2.

An example of such an algorithm is provided by the general algorithm from Bulatov and Jeavons [2001b]. That algorithm employs some deep algebraic results and sophisticated constructions. In the particular case of a 3-element domain, we may avoid using of "hard" algebra and apply a simpler and easier algorithm.

To this end, notice that if an idempotent 3-element algebra $\mathbb{A}$ has a 2-element subuniverse or a non-trivial congruence, and there is a term operation f which is not a projection on the corresponding subalgebra or quotient-algebra, then f witnesses that $\mathbb{A}$ itself is also not a G-set. We therefore have two cases to consider.

Case 1. $\mathbb{A}$ has no 2-element subuniverse and no proper congruence.

Such an algebra is said to be *strictly simple*. There is a complete description of finite strictly simple algebras [Szendrei 1990]. In particular, if a strictly simple algebra satisfies the condition (NO-G-SET), then one of the following operations is

INPUT A finite constraint language Γ on a 3-element set A.

OUTPUT 'YES' if Γ is tractable, 'NO' otherwise.

Step 1 **find** all the unary operations on A that preserves each relation from Γ
Step 2 **if** there is a unary non-identity operation f such that $f(f(x)) = f(x)$ **then** take one with a minimal range, and replace Γ with $f(\Gamma)$, and A with $f(A)$
Step 3 **add** the relations $\{(a)\}$, $a \in A$, to Γ
Step 4 **find** the set F of all ternary operations preserving each relation from Γ
Step 5 **find** the set S of all 2-element subsets from A and the set C of all proper equivalence relations invariant under operations from F
Step 6 **if** $S = C = \emptyset$ **then do**
Step 7 **if** F contains either a majority operation, or the operation t_a for some $a \in A$, or the Mal'tsev operation $x - y + z$ of an Abelian group, **then output**('YES')
Step 8 **otherwise output**('NO')
endif
Step 9 **otherwise**, **for each** $B \in S$ (each $\theta \in C$), **do**
Step 10 **check** if there is $f \in F$ such that $f\big|_B$ (f^θ) is one of the Boolean operations $\wedge$, $\vee$, $(x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$, $x + y + z$
Step 11 **if not then output**('NO')
endif
Step 12 **output**('YES')

FIG. 4.

its term operation: a majority operation, the Mal'tsev operation $x - y + z$ of an Abelian group, or the operation

$$t_0(x, y) = \begin{cases} 0, & \text{if } 0 \in \{x, y\}, \\ x & \text{otherwise} \end{cases}$$

for some element $0 \in A$. In the last case, $\mathbb{A}$ satisfies the partial zero property for $Z = \{B \mid B \text{ is a subuniverse of } \mathbb{A}, \text{ and } 0 \in B\}$, and $z_B = 0$ for $B \in Z$.

Case 2. $\mathbb{A}$ has either a 2-element subalgebra or a proper congruence.

In this case, $\mathbb{A}$ satisfies the condition (NO-G-SET) if and only if every its 2-element subalgebra and every proper factor-algebra (which is also 2-element) is not a G-set. In its turn, the latter condition holds if and only if, for any 2-element subuniverse B of $\mathbb{A}$ and any congruence θ, there is a polymorphism f of Γ such that $f\big|_B$ (or f^θ) is one of the following four Boolean operations: $\wedge$, $\vee$, the majority operation $(x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$, the affine operation $x - y + z$. Indeed, since $\mathbb{A}$ is idempotent, a constant cannot be its term operation.

It is well known and easy to verify that the subuniverses and congruences of a k-element algebra are completely determined by its k-ary term operations. Hence, we may restrict ourselves with finding ternary polymorphisms of a constraint language Γ. In the first three steps, the algorithm shown in Figure 4 constructs the language Γ^{id}.

This algorithm is polynomial time, because the hardest step, namely finding the set F, requires inspecting of all ternary operations on a 3-element set; and, since their number does not depend on Γ, takes cubic time.

6. *Proof of Theorem* 3

Everywhere in this section, $\mathbb{A} = (A; F)$, where $A = \{0, 1, 2\}$, is a 3-element algebra satisfying the conditions of Theorem 3.

6.1. OUTLINE OF THE PROOF. In this section, we sketch the proof of Theorem 3. Our general strategy is to show that the algebra $\mathbb{A}$ has a term operation (sometimes two) satisfying certain specific conditions, which guarantee that $\mathbb{A}$ posesses one of the 10 properties listed in Theorem 3. In Section 6.2, we identify several such properties of operations on a 3-element set. In most cases—there are only a few exceptions—the algebra $\mathbb{A}$ has a term operation with one of those properties.

Then we split the proof into four cases depending on the number of nontrivial subuniverses and congruences of the algebra $\mathbb{A}$. A 3-element algebra may have at most three nontrivial (i.e., 2-element) subuniverses and at most three proper congruences (i.e., those, which are not the equality relation or the total relation).

Case 1. In Section 6.3, we consider the case when $\mathbb{A}$ has neither nontrivial subuniverse, nor proper congruence; such an algebra is called *strictly simple*. A complexity classification of CSPs arising from strictly simple algebras is known [Bulatov et al. 2005], so the only thing to do is to show that in the tractable cases the algebra $\mathbb{A}$ satisfies some of the 10 properties.

Case 2. Section 6.4 studies the case when $\mathbb{A}$ has no proper congruences, and at least one of the 2-element subsets is a subuniverse and at least one is not. It is not hard to see that the latter condition implies that $\mathbb{A}$ has a binary term operation f such that $f(a, b) \notin \{a, b\}$, where $\{a, b\}$ is the 2-element subset that is not a subuniverse. We show (Lemma 7) that f can be chosen from a very small set of 16 binary operations. Some of these operations yield one of the required properties (Lemma 8), for the remaining operations we use the fact that $\mathbb{A}$ satisfies the condition (NO-G-SET) and that it has no proper congruences. The former condition and Schaefer's Dichotomy Theorem (Proposition 5) imply that there is a term operation g such that the restriction $g|_B$ onto the nontrivial subuniverse B is either a semilattice operation, or a majority operation, or an affine operation. Using the operations f and g we infer (Lemma 9) one of the 10 properties. Finally, for some of the 16 operations, the condition (NO-G-SET) is not sufficient and we have to use the latter condition. In this case, it can be shown (Lemma 10) that there is a binary or ternary term operation g of a very special form that certifies that certain equivalence relation is not a congruence of $\mathbb{A}$. As before, using the operations f and g we infer (Lemma 10 and 11) that $\mathbb{A}$ has one of the 10 properties.

Case 3. This case is studied in Section 6.5; it includes *conservative* algebras (both, with a proper congruence and without), that is algebras such that every 2-element subset of the universe is a subuniverse. By Proposition 10 and Schaefer's Dichotomy Theorem, for every 2-element subset B, there is a term operation f such that the restriction $f|_B$ is either a semilattice operation, or a majority operation, or an affine operation. We split into five cases depending on the combination of the types of such operations on all three 2-element subsets. In the first case, for some B, the restriction $f|_B$ is a semilattice operation. In this case, we easily derive that $\mathbb{A}$ satisfies one of the required properties. So, in the remaining cases, every restriction of the form $f|_B$ is either a majority or affine operation. Lemma 12 shows that all operations with such restrictions can be combined into a single operation f. In the second and third cases, we assume that the restrictions $f|_B$ are either all majority operations, or all affine operations, respectively, and therefore f is a majority or affine operation itself. The fourth case deals with the situation when two of the restrictions are majority operations and one is an affine operation. Then Lemma 13 shows that $\mathbb{A}$ satisfies the semisplitting property. The last case, when one of the restrictions is a majority operation and the others are affine operations, is the hardest. In this case, we need to consider two subcases depending on whether $\mathbb{A}$ has a proper congruence, and to study more closely the structure of invariant relations (Lemmas 14 and 15). Then, we use these structural properties to prove that $\mathbb{A}$ has either the splitting property or the semirectangular property.

Case 4. In Section 6.6, we study the case when the algebra $\mathbb{A}$ has a nontrivial congruence θ. Then it can be shown that the 2-element class B of θ is a subuniverse. Therefore, there are two 2-element algebras related to $\mathbb{A}$: the subalgebra $\mathbb{B}$ with the universe B and the quotient-algebra $\mathbb{A}/_\theta$. Since $\mathbb{A}$ satisfies the condition (NO-G-SET) and by Schaefer's Dichotomy Theorem, there are two operations, f and g, that guarantee (NO-G-SET) for $\mathbb{B}$ and for $\mathbb{A}/_\theta$, respectively. Thus, both of them are either a semilattice, or majority, or affine operations. The proof splits again into five cases. In the first two cases, $f\big|_B$ or g^θ, respectively, is a semilattice operation; in the remaining three cases, $f\big|_B$ and g^θ are either majority or affine operations, and we proceed accordingly the number of nontrivial subuniverses. If $\mathbb{A}$ has three subuniverses, then it is conservative; this case is studied in Section 6.5. If $\mathbb{A}$ has two subuniverses then the results of Section 6.4 allows us to infer one of the 10 properties for $\mathbb{A}$. If $\mathbb{A}$ has only one subuniverse then the results of Section 6.4 yield that only 3 of the 16 operations listed in Lemma 7 can be a term operation of $\mathbb{A}$. For two of them the proof is simple, but for the third one the proof is more elaborate. In Lemmas 18 and 19, we manage to show that either one of the previous cases takes place, or g is of a very restricted form. In the latter case, we modify the algebra $\mathbb{A}$ by throwing out some of its term operations. The resulting algebra still satisfies the condition (NO-G-SET) and is conservative. Therefore, we again get one of the previous cases.

It is straightforward to check that the cases considered cover all possibilities, and therefore Theorem 3 is proved.

6.2. PREREQUISITES. In this section, we state and prove some auxiliary statements showing that the presence of term operations with satisfying certain conditions implies one of the 10 properties for the algebra $\mathbb{A}$. An element $a \in A$ is said to be a *right-zero* [*left-zero*] with respect to an operation $t(x, y)$ if $t(x, a) = a$ [$t(a, x) = a$] for any $x \in A$. An operation of any of the following two types guarantees the tractability of an algebra.

Let $A = \{a, b, c\}$. A binary operation f is said to be an $(a - b)$-operation if b, c are left- [right-]zeroes, and $\{f(a, a), f(a, b), f(a, c)\} = \{a, b\}$ [$\{f(a, a), f(b, a), f(c, a)\} = \{a, b\}$]. It is said to be a *zero-operation* if one of the following conditions holds:

a is a right-zero, $a \in \{f(a, b), f(c, b)\}$ and $\{f(a, c), f(b, c)\} \cap \{a, b\} \neq \emptyset$;
a is a left-zero, $a \in \{f(b, a), f(b, c)\}$ and $\{f(c, a), f(c, b)\} \cap \{a, b\} \neq \emptyset$.

LEMMA 3. *Let an algebra $\mathbb{A} = (\{a, b, c\}, F)$ have a term $(a - b)$-operation. Then $\mathbb{A}$ satisfies the $(a - b)$-replacement property.*

PROOF. Suppose first that b, c are left-zeroes. Take a (n-ary) relation $R \in \mathsf{Pol}\ F$ and a tuple $\mathbf{a} \in R$. Let, for $k \leq n$, a tuple $\mathbf{a}_k$ be defined as follows:

$$\mathbf{a}_k[i] = \begin{cases} \mathbf{a}[i], & \text{if } \mathbf{a}[i] \in \{b, c\}, \\ b, & \text{if } \mathbf{a}[i] = a, i \leq k, \end{cases}$$

if $f(a,b)=b$; and

$$\mathbf{a}_k[i] = \begin{cases} \mathbf{a}[i], & \text{if } \mathbf{a}[i] \in \{b,c\},\\ b, & \text{if } \mathbf{a}[i] = a,\ i \le k, \text{ and } R_i = A, \end{cases}$$

otherwise. We prove that $\mathbf{a}_k$ belongs to R. Since $\mathbf{a}_n$ is the tuple required in the $(a-b)$-replacement property, this proves the lemma.

The tuple $\mathbf{a}_0$ can be set to be $\mathbf{a}$. Suppose that a tuple $\mathbf{a}_k$ is already found. If $\mathbf{a}_k[k+1] \in \{b,c\}$, or $R_{k+1} = \{a,b\}$ and $f(a,b)=a$, then set $\mathbf{a}_{k+1} = \mathbf{a}_k$. Otherwise set $\mathbf{a}_{k+1} = f(\mathbf{a}_k, \mathbf{b})$, where $\mathbf{b} \in R$ is such that $\mathbf{b}[k+1] = b$ if $f(a,b)=b$, and $\mathbf{b}[k+1] = c$ if $f(a,c)=b$.

The case when b, c are right-zeroes is quite similar. □

LEMMA 4. *Let an algebra $\mathbb{A} = (\{a,b,c\}, F)$ have a term operation that is a zero-operation. Then $\mathbb{A}$ satisfies the partial zero property.*

PROOF. Suppose first that a is a left-zero with respect to f. Then Z is the set consisting of A and all the 2-element subuniverses B of $\mathbb{A}$ such that $f|_B$ is a semilattice operation. Notice that if a belongs to such a subuniverse, then $f(a,x) = f(x,a) = a$, for any x from this subuniverse. Therefore, $z_C = a$ for any $C \in Z$ with $a \in C$, and $z_{\{b,c\}} = f(b,c)$ if $\{b,c\} \in Z$.

Take a subuniverse R of $\mathbb{A}^n$, and a tuple $\mathbf{a} \in R$. We prove that, for all $k \le n$, a tuple $\mathbf{a}_k$ with

$$\mathbf{a}_k[i] = \begin{cases} z_{R_i}, & \text{if } R_i \in Z \text{ and } i \le k,\\ \mathbf{a}[i], & \text{if } R_i \notin Z \end{cases}$$

belongs to R.

Set $\mathbf{a}_0 = \mathbf{a}$, and suppose that $\mathbf{a}_k$ is already found. Then

$$\mathbf{a}_{k+1} = \begin{cases} \mathbf{a}_k, & \text{if } R_{k+1} \notin Z, \text{ or } \mathbf{a}_k[k+1] = z_{R_{k+1}};\\ f(\mathbf{a}_k, \mathbf{b}), & \text{if } \mathbf{a}_k[k+1] = b,\ a \in R_{k+1}, \text{ and } \mathbf{b} \in R \text{ is such}\\ & \text{that } \mathbf{b}[k+1] = x \text{ and } f(b,x) = a;\ \text{or}\\ & \mathbf{a}_k[k+1] = c,\ a \in R_{k+1},\ a \in \{f(c,a), f(c,b)\},\\ & \text{and } \mathbf{b} \in R \text{ is such that } \mathbf{b}[k+1] = x \text{ and}\\ & f(c,x) = a;\\ f(f(\mathbf{a}_k, \mathbf{b}), \mathbf{c}), & \text{if } \mathbf{a}_k[k+1] = c,\ a \in R_{k+1}, \text{ and } \mathbf{b}, \mathbf{c} \in R \text{ are such}\\ & \text{that } \mathbf{b}[k+1] = x,\ \mathbf{c}[k+1] = y \text{ where } f(c,x) = b,\\ & f(b,y) = a;\\ f(\mathbf{a}_k, \mathbf{b}), & \text{if } \{b,c\} \in Z,\ x = f(b,c),\ \{y\} = \{b,c\} - \{x\},\\ & \mathbf{a}_k[k+1] = y, \text{ and } \mathbf{b} \in R \text{ is such that}\\ & \mathbf{b}[k+1] = x. \end{cases}$$

If a is a right-zero, the proof is quite similar. □

An important particular type of zero-operations rises from zero-elements. An element $a \in A$ is said to be a *zero-element* with respect to a binary operation $f(x,y)$, if $f(a,x) = f(x,a) = a$, for any $x \in A$.

LEMMA 5. *If a is a zero-element with respect to $f(x, y)$, then f is a zero-operation.*

We also need two simple observations that will be frequently used.

LEMMA 6

(1) *If $f(x, y)$ is an idempotent operation on a 2-element set, then f is either a projection or a semilattice operation.*
(2) *If $f(x, y)$ is an idempotent operation on a 2-element set, then $f(x, f(y, x))$ $[f(f(y, x), y)]$ is a semilattice operation if f is a semilattice operation, and is the first* [*second*] *projection otherwise.*

We consider four cases depending on the number of 2-element subalgebras and proper congruences of the algebra $\mathbb{A}$. Recall that an algebra is said to be *simple* if it has at most two congruences: the equality relation and the total binary relation.

6.3. STRICTLY SIMPLE ALGEBRAS. A *strictly simple algebra* is a simple algebra that has no subalgebras but one-element and the algebra itself. Finite strictly simple surjective algebras have been described by Szendrei [1990], and the complexity of the corresponding CSPs was found in Bulatov et al. [2005]. To formulate Szendrei's result, we need some notation.

Let G be a permutation group on a set A. By $\mathrm{R}(G)$, we denote the set of operations on A preserving each relation of the form $\{(a, g(a)) \mid a \in A\}$ where $g \in G$. Let also $\mathrm{F}(G)$ denote the set of idempotent members of $\mathrm{R}(G)$.

Let ${}_F\overline{A} = (A; +, F)$ be a finite vector space over a finite field F, $\mathsf{End}\ {}_F\overline{A}$ the endomorphism ring of ${}_F\overline{A}$. Then one can consider $\overline{A}$ as a module over $\mathsf{End}\ {}_F\overline{A}$. This module is denoted by ${}_{(\mathsf{End}\ {}_F\overline{A})}\overline{A}$.

Finally, F_k^0 denotes the set of all operations preserving the relation

$$X_k^0 = \{(a_1, \dots, a_k) \in A^k \mid a_i = 0 \text{ for at least one } i,\ 1 \le i \le k\},$$

where 0 is some fixed element of A, and let $\mathrm{F}_\omega^0 = \bigcap_{k=2}^{\infty} \mathrm{F}_k^0$.

Algebras are said to be *term equivalent* if their sets of term operations are equal.

PROPOSITION 15 (COROLLARY 3.10 OF SZENDREI [1990]). *Let $\mathbb{A}$ be a finite idempotent strictly simple algebra. Then it is term equivalent to one of the following algebras:*

(*a*) $(A, \mathrm{F}(G))$ *for a permutation group G on A such that every non-identity member of G has at most one fixed point;*

(*b*) (A, F), *where F is the set of all idempotent term operations of ${}_{(\mathsf{End}\ {}_K\overline{A})}\overline{A}$ for some vector space ${}_K\overline{A}$ over a finite field K;*

(*c*) $(A, \mathrm{F}(G) \cap \mathrm{F}_k^0)$ *for some k ($2 \le k \le \omega$), some element $0 \in A$ and some permutation group G on A such that 0 is the unique fixed point of every non-identity member of G;*

(*d*) (A, F), *where $|A| = 2$ and F contains a semilattice operation;*

(*e*) *a two-element algebra with empty set of basic operations.*

In Bulatov et al. [2005], tractable strictly simple algebras have been characterised: a finite idempotent strictly simple algebra is tractable if and only if it is of type (a), (b), (c), (d); otherwise it is NP-complete. As is easily seen, exactly those algebras satisfy the condition of Theorem 3. In the same paper, we noticed that in the case (a) the dual discriminator operation, that is the majority operation

$$d(x, y, z) = \begin{cases} y, & \text{if } y = z, \\ x & \text{otherwise;} \end{cases}$$

and in the case (b) the Mal'tsev operation $x - y + z$ of the vector space are term operations of the algebra. In the case (d) the algebra has the term operation

$$r(x, y) = \begin{cases} 0, & \text{if } 0 \in \{x, y\}, \\ x, & \text{otherwise,} \end{cases}$$

and 0 is a zero-element with respect this operation; in the case (d) the algebra has a term semilattice operation. Since any semilattice has a zero-element, Theorem 3 holds for 3-element strictly simple algebras.

6.4. Simple Algebras with One or Two Subalgebras. In this section, we assume that $\mathbb{B} = \{0, 1\}$ is a subalgebra of $\mathbb{A}$, but at least one of $\{0, 2\}, \{1, 2\}$ is not.

We start with showing that $\mathbb{A}$ has one of the following binary term operations:

$$(1)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 2 \end{array},\ (2)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 2 \\ 1 & 0 & 1 & 2 \\ 2 & 0 & 2 & 2 \end{array},\ (3)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 1 & 2 \end{array},\ (4)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 0 & 1 & 0 \\ 2 & 2 & 1 & 2 \end{array},$$

$$(5)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 0 & 2 & 2 \end{array},\ (6)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 0 \\ 2 & 2 & 0 & 2 \end{array},\ (7)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array},\ (8)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array},$$

$$(9)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 2 & 2 \end{array},\ (10)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array},\ (11)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array},\ (12)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array},$$

$$(13)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array},\ (14)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array},\ (15)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array},\ (16)\ \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array}.$$

Lemma 7. *If* $\{0, 1\}$ *is a subuniverse of* $\mathbb{A}$*, but at least one of* $\{0, 2\}, \{1, 2\}$ *is not, then* $\mathbb{A}$ *either has a binary term operation* f *and a zero-element with respect to* f*, or a zero-operation, or a* $(2 - a)$*-operation,* $a \in \{0, 1\}$*, or one of the operations*
(*a*) (1), (2), (4), (6), (8), (10), (11), (12), (14), (16) *if* $\{1, 2\}$ *is not a subuniverse;*
(*b*) (1), (3), (5), (7), (8), (9), (11), (13), (15) *if* $\{0, 2\}$ *is not a subuniverse.*

Proof. Without loss of generality, we may assume that $\{1, 2\}$ is not a subuniverse. Indeed, the case when $\{0, 2\}$ is not a subuniverse, the proof is dual in the sense that 0 and 1 are swopped. Since $\{1, 2\}$ is not a subuniverse of $\mathbb{A}$, there is a term operation $f(x_1, \dots, x_n)$, and $a_1, \dots, a_n \in \{1, 2\}$ such that $f(a_1, \dots, a_n) = 0$. We may assume that $a_1 = \dots = a_k = 1, a_{k+1} = \dots = a_n = 2$. Then, the operation $f(\underbrace{x, \dots, x}_{k}, y, \dots, y)$ also destroys the set $\{1, 2\}$, and therefore f can be chosen to be binary.

On the other hand, $B = \{0, 1\}$ is a subuniverse. Hence, f preserves B, and we have four cases depending on what is the restriction of f onto B, a semilattice operation, the first or the second projection.

Case 1. $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & \\ 1 & 0 & 1 & 0 \\ 2 & & & 2 \end{array}$, that is $f|_B$ is a semilattice operation, $0 \geq 1$.

Subcase 1.1. $f(2, 1) = 0$.
If $f(2, 0) = 0$, then f is a zero-operation; if $f(2, 0) = 1$, then $f(x, f(x, y))$ is a zero-operation. If $f(2, 0) = 2$ then $g(x, y) = f(x, f(x, y))$ is the operation (10) in the case $f(0, 2) \in \{0, 1\}$, and 2 is a zero-element with respect to $g(g(x, y), y)$ in the case $f(0, 2) = 2$.

Subcase 1.2. $f(2, 1) = 1$.
In this case, we may obtain Subcase 1.1 by substituting $f(f(x, y), f(y, x))$.

Subcase 1.3. $f(2, 1) = 2$.

Subcase 1.3.1. $f(2, 0) = 0$. The operation f is a zero-operation.

Subcase 1.3.2. $f(2, 0) = 1$.
If $f(0, 2) \in \{0, 1\}$, then 0 is a zero-element with respect to the operation $f(f(x, y), y)$. In the case $f(0, 2) = 2$, 0 is a zero-element with respect to $f(f(x, y), x)$.

Subcase 1.3.3 $f(2, 0) = 2$.
If $f(0, 2) \in \{0, 1\}$, then $f(x, y)$ is one of the operations (10), (11). If $f(0, 2) = 2$, then 2 is a zero-element with respect to $f(f(x, y), y)$.

Case 2. $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & \\ 1 & 1 & 1 & 0 \\ 2 & & & 2 \end{array}$, that is $f|_B$ is a semilattice operation, $1 \geq 0$.

Subcase 2.1. $f(2, 1) \in \{0, 1\}$.

Subcase 2.1.1. $f(2, 0) \in \{0, 1\}$.
In this case, 1 is a zero-element with respect to $g(x, y) = f(x, f(x, y))$ if $f(2, 1) = 1$, and with respect to $g(g(x, y), y)$ if $f(2, 1) = 0$.

Subcase 2.1.2. $f(2, 0) = 2$.
If $f(2, 1) = 1$, then 1 is a zero-element with respect to the operation $f(x, f(x, y))$. In the case $f(2, 1) = 0$, $f(f(x, y), y)$ is a zero-operation if $f(0, 2) = 0$, or 1 is a zero-element with respect to $f(f(y, x), y)$ if $f(0, 2) = 1$, or f is the operation (6) if $f(0, 2) = 2$.

Subcase 2.2. $f(2, 1) = 2$.
If $f(2, 0) \in \{0, 1\}$, then by substituting $f(x, f(y, x))$ we get Subcase 2.1. Consider the case when $f(2, 0) = 2$. If $f(0, 2) = 0$, then f is the operation (12); if $f(0, 2) = 1$, then f is the operation (8); and if $f(0, 2) = 2$, then 2 is a zero-element with respect to $f(f(x, y), y)$.

Case 3. $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & \\ 1 & 1 & 1 & 0 \\ 2 & & & 2 \end{array}$, that is $f|_B$ is the first projection.

Subcase 3.1. $f(2,1)=1$.

Subcase 3.1.1. $f(0,2)=0$. In this case, f is a zero-operation.

Subcase 3.1.2. $f(0,2)=1$.
If $f(2,0)=0$, then, for $g(x,y)=f(f(x,y),x)$, the operation $g(x,g(x,y))$ is the operation (1). If $f(2,0)\in\{1,2\}$, then $f(f(x,y),y)$ is a $(2-1)$-operation.

Subcase 3.1.3. $f(0,2)=2$.
If $f(2,0)=0$, then $f(x,f(y,x))$ is a $(2-0)$-operation. If $f(2,0)=1$, then $f(y,f(y,x))$ is a zero-operation. If $f(2,0)=2$, then f is the operation (4).

Subcase 3.2. $f(2,1)=0$.

Subcase 3.2.1. $f(0,2)=0$. In this case, $f(y,x)$ is a zero-operation.

Subcase 3.2.2. $f(0,2)=1$. The operation $f(f(x,y),y)$ is a $(2-0)$- or a $(2-1)$-operation.

Subcase 3.2.3. $f(0,2)=2$.
If $f(2,0)=0$, then $f(f(x,y),x)$ is the operation (14); if $f(2,0)=1$, then $f(f(x,y),x)$ is a zero-operation. Finally, in the case $f(2,0)=2$, set $h(x,y)=f(f(x,y),y)$. Then 2 is a zero-element with respect to the operation $h(x,h(x,y))$.

Subcase 3.3. $f(2,1)=2$.

Subcase 3.3.1. $f(2,0)\in\{0,1\}$.

In this case, the operation table of $f(x,f(y,x))$ is $\begin{array}{c|ccc} & 0 & 1 & 2\\ \hline 0 & 0 & 0 & \\ 1 & 1 & 1 & 0\\ 2 & & 0/1 & 2\end{array}$, and therefore, we get one of Subcases 3.1 and 3.2.

Subcase 3.3.2. $f(2,0)=2$.
If $f(0,2)=0$, then f is the operation (14). In the case $f(0,2)=1$, f is the operation (16). Finally, if $f(0,2)=2$, then 2 is a zero-element with respect to $f(f(x,y),y)$.

Case 4. $\begin{array}{c|ccc} f & 0 & 1 & 2\\ \hline 0 & 0 & 1 & \\ 1 & 0 & 1 & 0\\ 2 & & & 2\end{array}$, that is $f|_B$ is the second projection.

Subcase 4.1. $f(2,1)=1$.

Subcase 4.1.1. $f(2,0)=0$.
If $f(0,2)\in\{0,2\}$, then f is a $(2-0)$-operation. If $f(0,2)=1$, then $f(f(x,y),y)$ is the operation (1).

Subcase 4.1.2. $f(2,0)=1$. In this case, f is a zero-operation.

Subcase 4.1.3. $f(2,0)=2$.
If $f(0,2)=0$, then $f(f(x,y),x)$ is a $(2-0)$-operation; if $f(0,2)=1$, then f is a zero-operation; and if $f(0,2)=2$, then f is the operation (4).

Subcase 4.2. $f(2,1)=0$.

Consider the operation $g(x,y)=f(x,f(x,y))$. Its operation table is $\begin{array}{c|ccc} g & 0 & 1 & 2\\ \hline 0 & 0 & 1 & \\ 1 & 0 & 1 & 0\\ 2 & & f(2,0) & 2\end{array}$.

Therefore, if $f(2,0) = 1$ or 2, then we get Subcase 4.1 or Subcase 4.3, respectively. If $f(2,0) = 0$ then $g(2,0) = g(2,1) = 0$, and g is a zero-operation.

Subcase 4.3. $f(2,1) = 2$.

Subcase 4.3.1. $f(2,0) = 0$. In this case, f is a zero-operation.

Subcase 4.3.2. $f(2,0) \in \{1,2\}$.

Set $g(x,y) = f(x, f(x,y))$. Its operation table is $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & f(0,2) \\ 1 & 0 & 1 & 0 \\ 2 & 2 & 2 & 2 \end{array}$. If $f(0,2) = 0$, then $g(x, g(y,x))$ is the operation (14). If $f(0,2) = 2$, then g is a zero-operation. Finally, in the case $f(0,2) = 1$, the operation $f(x, f(y,x))$ is the operation (16). $\square$

If $\mathbb{A}$ has a zero-element with respect to a binary term operation, or it has a zero-operation, or a $(2-a)$-operation, then by Lemmas 3,4, $\mathbb{A}$ satisfies either the partial zero-property, or the $(2-a)$-replacement property. We, therefore, have to show that if $\mathbb{A}$ has one of the numbered term operations, then $\mathbb{A}$ satisfies one of the 10 properties. Sometimes we do not need any other operations.

LEMMA 8. *Let $\mathbb{A}$ have a term operation f which is one of the operations* (4), (5), (6), *or* (9). *Then*
$\mathbb{A}$ *satisfies* $(2-0)$*-replacement property if f is* (4)*;*
$\mathbb{A}$ *satisfies* $(2-1)$*-replacement property if f is* (5)*;*
$\mathbb{A}$ *satisfies the* $\{1,2\}$*-extendibility property if f is* (6)*;*
$\mathbb{A}$ *satisfies the* $\{0,2\}$*-extendibility property if f is* (9).

PROOF. Let f be the operation (4). Take an (n-ary) relation $R \in \mathsf{Inv}\ F$. We prove that, for any $\mathbf{a} \in R$, and any $k \le n$, there is $\mathbf{a}_k \in R$ such that

$$\mathbf{a}_k[i] = \begin{cases} 1, & \text{if } \mathbf{a}_k[i] = 1, \\ 0, & \text{if } \mathbf{a}_k[i] \in \{0,2\},\ i \le k, \text{ and } R_i = \{0,1,2\}, \\ 2, & \text{if } i \le k, \text{ and } R_i = \{0,2\}, \\ 2, & \text{if } 1 \notin R_i \text{ and } \mathbf{a}[i] = 2, \\ 0 \text{ or } 2 & \text{otherwise.} \end{cases}$$

Clearly, $\mathbf{a}_n$ is the tuple required in the $(2-0)$-replacement property.

Since $\mathbf{a}_0$ can be set to be $\mathbf{a}$, we have the base case of induction. Further, suppose that there is $\mathbf{a}_k \in R$ with the required properties.

Case 1. $\mathbf{a}_k[k+1] = 1$, or $\mathbf{a}[k+1] = 2$ and $1 \notin R_{k+1}$, or $\mathbf{a}_k[k+1] = 0$ and $R_{k+1} \in \{\{0,1,2\}, \{0,1\}, \{0\}\}$.
In this case, set $\mathbf{a}_{k+1} = \mathbf{a}_k$.

Case 2. $\mathbf{a}_k[k+1] = 0$ and $R_i = \{0,2\}$.
There is $\mathbf{b} \in R$ with $\mathbf{b}[k+1] = 2$. By the induction hypothesis, there is $\mathbf{b}_k \in R$. It can be straightforwardly verified that the tuple $\mathbf{a}_{k+1} = f(\mathbf{b}_k, \mathbf{a}_k)$ satisfies the required conditions.

Case 3. $\mathbf{a}_k[k+1] = 2$ and $1 \in R_{k+1}$.
This case is very similar to the previous one, but $\mathbf{b}$ is to be chosen such that $\mathbf{b}[k+1] = 1$.

Now let f be the operation (6). We prove that $\mathbb{A}$ satisfies the $\{1,2\}$-extendibility property. To this end, notice first that the operation $g(x,y,z) = f(f(x, f(y,z)), f(f(x,y),z))$ is the majority operation on $\{1,2\}$. Moreover, $f(f(x,y),y)$ is the operation (3). The required result follows from Lemma 9.

The arguments for the operations (5) and (9) are quite similar. $\square$

Sometimes we need to use the condition (NO-G-SET).

LEMMA 9. *Let $\mathbb{A}$ have a term operation f which is one of the operations* (1), (2), (3), $B = \{0,1\}$ *if f is* (1), $B = \{0,2\}$ *if f is* (2), $B = \{1,2\}$ *if f is* (3), *and a term operation g which is either a semilattice or majority or affine operation on B. Then*

if $g|_B$ is a semilattice operation, then $\mathbb{A}$ has a semilattice operation and therefore, satisfies the partial zero property;

if $g|_B$ is a majority operation, then $\mathbb{A}$ satisfies the B-extendibility property;

if $g|_B$ is an affine operation, then $\mathbb{A}$ is $\{0,1\}$*-rectangular.*

PROOF. We prove the lemma in the case when f is the operation (1); the other two cases are quite similar.

If a term operation g is such that $g|_{\{0,1\}}$ is a semilattice operation, then $g(f(x,y), f(y,x))$ is a semilattice operation on A.

Now, suppose that g is a majority operation on B and $R \in \mathsf{Inv}\ F$ an (n-ary) relation. We show first that, for any $k, l \in \underline{n}$ with $R_k = R_l = A$ and $a \in B$, there is $b \in B$ such that $\binom{a}{b} \in R_{\{k,l\}}$; and, moreover, for any $\binom{a}{b} \in R_{\{k,l\}}$ such that $a, b \in B$, there is $\mathbf{a} \in R$ with $\mathbf{a}[k] = a$, $\mathbf{a}[l] = b$ and $\mathbf{a}[i] \in B$ for all i such that $R_i = A$. Without loss of generality, we may assume that $k = 1$, $l = 2$ and $\mathbf{a}_1 \in R$ is such that $\mathbf{a}_1[1] = a$. Then, there is $\mathbf{b} \in R$ with $\mathbf{b}[2] \in B$. We also set $\mathbf{c} = f(\mathbf{b}, \mathbf{a}_1)$. As is easily seen, $\mathbf{c}[1] = \mathbf{a}[1] = a$, $\mathbf{c}[2] \in B$. Furthermore, let $\binom{a}{b} \in R_{\{1,2\}} \cap B^2$, and let $\mathbf{a}_2 \in R$ be such that $\mathbf{a}_2[1] = a$, $\mathbf{a}_2[2] = b$. Then, for any $k \le n$, there is $\mathbf{a}_k \in R$ such that $\mathbf{a}_k[1] = a$, $\mathbf{a}_k[2] = b$, and $\mathbf{a}_k[i] \in B$ for all $i \le k$ such that $R_i = A$. Indeed, suppose that $\mathbf{a}_{k-1}$ is already found. Then, $\mathbf{a}_k$ can be chosen to be $\mathbf{a}_{k-1}$ if $\mathbf{a}_{k-1}[k] \in B$ or $R_k \neq A$. Otherwise, there is $\mathbf{b} \in R$ with $\mathbf{b}[k] \in B$, and the tuple $\mathbf{a}_k = f(\mathbf{b}, \mathbf{a}_{k-1})$ satisfies the required conditions. Finally, $\mathbf{a}$ can be chosen to be $\mathbf{a}_n$.

Furthermore, denote $W = \{i \in \underline{n} \mid R_i = A\}$ and take $\mathbf{a} \in B^{|W|}$ such that $\binom{\mathbf{a}[i]}{\mathbf{a}[j]} \in R_{\{i,j\}}$ for any $i, j \in W$. By what was proved above, for any $k, l \in W$, there is $\mathbf{a}_{k,l} \in R$ with $\mathbf{a}_{k,l}[k] = \mathbf{a}[k]$, $\mathbf{a}_{k,l}[l] = \mathbf{a}[l]$, and $\mathbf{a}_{k,l}[i] \in B$ whenever $i \in W$. Since g is a majority operation on B, by Theorem 3.5 from Jeavons et al. [1998a], there is $\mathbf{b} \in R$ such that $\mathbf{b}_W = \mathbf{a}$.

Let $R = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$, and $\mathbf{c} = f(\cdots f(\mathbf{c}_1, \mathbf{c}_2), \dots, \mathbf{c}_m)$. Then, $\mathbf{c}[i] = 0$ if $R_i = \{0,2\}$, and $\mathbf{c}[i] = 1$ if $R_i = \{1,2\}$; and, for the tuple $\mathbf{d} = f(\mathbf{c}, \mathbf{b})$, we have $\mathbf{d}_W = \mathbf{b}_W = \mathbf{a}$, $\mathbf{d}[i] = \mathbf{c}[i] = a$ if $R_i = \{a, 2\}$, $a \in B$.

Let g be a term operation that is an affine operation on $\{0,1\}$, and $R \in \mathsf{Inv}\ F$ an (n-ary) relation. Let also $W = \{u \mid 0, 1 \in R_u\}$, $W' = \underline{n} - W$, and $W_1, \dots, W_k \subseteq W$ be the blocks of $\theta_B(R)$. We have to prove that

$$R_W \cap \{0,1\}^{|W|} = \left(R_{W_1} \cap \{0,1\}^{|W_1|}\right) \times \cdots \times \left(R_{W_k} \cap \{0,1\}^{|W_k|}\right).$$

Notice first that replacing g with $g(x, f(x,y), z)$, we may assume that $g(x, 2, z) = z$ whenever $x, z \in \{0,1\}$.

Let $\mathbf{a}_i \in R_{W_i} \cap \{0,1\}^{|W_i|}$ for $i \in \underline{k}$. Suppose that we have proved that, for any $l-1$-element subset $I \subseteq \underline{k}$, there is $\mathbf{a} \in R$ with $\mathbf{a}_{W_i} = \mathbf{a}_i$ whenever $i \in I$. Take an l-element subset $J \subseteq \underline{k}$; without loss of generality we may assume $J = \{1,\ldots,l\}$ and $U = W_1 \cup \cdots \cup W_l$. There exists a tuple $\mathbf{b} \in R$ such that $\mathbf{b}_{W_1} \in \{0,1\}^{|W_1|}$, $\mathbf{b}_{W_l} = (2,\ldots,2)$, or vice-versa. It will not be loss of generality if we suppose that $\mathbf{b}_{W_l} = (2,\ldots,2)$, and there is $m < l$ such that, for any $i \le l$, $\mathbf{b}_{W_i} = (2,\ldots,2)$ if and only if $i \ge m$. There also exist $\mathbf{a}, \mathbf{c} \in R$ such that $\mathbf{a}_{W_i} = \mathbf{a}_i$ for all $1 \le i < m$, and $\mathbf{c}_{W_i} = \mathbf{a}_i$ for all $m \le i \le k$. By rearranging the coordinate positions, the tuples $\mathbf{a}_U, \mathbf{b}_U, \mathbf{c}_U$ can be viewed as consisting of four parts: the first one, U_1, includes those coordinate positions in which all the tuples have 0 or 1, this part is a subset of $W_1 \cup \cdots \cup W_{m-1}$ and is nonempty; the second part, U_2, equals to $(W_1 \cup \cdots \cup W_{m-1}) - U_1$, and consists of those coordinate positions in which $\mathbf{a}, \mathbf{b}$ have 0, 1 while $\mathbf{c}$ has 2; the third part, $U_3 \subseteq W_m \cup \cdots \cup W_l$, contains those positions in which $\mathbf{a}$ is 0, 1; finally, the last part, U_4, contains the remaining coordinate positions, and in each such a position the corresponding entry of $\mathbf{a}, \mathbf{b}$ equal 2. So, we may represent the tuples in the form $\begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \mathbf{a}^3 \\ \overline{2} \end{pmatrix}$, $\begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \overline{2} \\ \overline{2} \end{pmatrix}$ and $\begin{pmatrix} \mathbf{c}^1 \\ \overline{2} \\ \mathbf{c}^3 \\ \mathbf{c}^4 \end{pmatrix}$, respectively, where $\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3 \mathbf{b}^1, \mathbf{b}^2, \mathbf{c}^1, \mathbf{c}^3, \mathbf{c}^4$ consist of 0s and 1s. Then, set $\mathbf{d} = f(\mathbf{c}, \mathbf{b}) = \begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \mathbf{c}^3 \\ \mathbf{c}^4 \end{pmatrix}$ and $\mathbf{a}' = f(\mathbf{c}, \mathbf{a}) = \begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \mathbf{a}^3 \\ \mathbf{c}^4 \end{pmatrix}$. Finally, we have

$$\mathbf{e} = g(\mathbf{a}', \mathbf{b}, \mathbf{d}) = g\left(\begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \mathbf{a}^3 \\ \mathbf{c}^4 \end{pmatrix}, \begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \overline{2} \\ \overline{2} \end{pmatrix}, \begin{pmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \mathbf{c}^3 \\ \mathbf{c}^4 \end{pmatrix} \right) = \begin{pmatrix} \mathbf{a}^1 \\ \mathbf{a}^2 \\ \mathbf{c}^3 \\ \mathbf{c}^4 \end{pmatrix}.$$

The tuple $\mathbf{e}$ satisfies the condition $\mathbf{e}_{|W_i} = \mathbf{a}_i$ whenever $i \in J$ as required. $\square$

In the remaining cases, an operation from the list in Lemma 7 and the condition (NO-G-SET) are not sufficient. We have to use the fact that $\mathbb{A}$ is simple and therefore, for any notrivial equivalence relation it has a term operation that does not preserve this relation. First, we show that such an operation can be chosen to satisfy some conditions. Recall that θ denotes the equivalence relation whose classes are $\{0,1\}$ and $\{2\}$.

LEMMA 10

(a) If a simple algebra $\mathbb{A}$ has a term operation that is one of the operations (7), (10), (12), (13), (14), or (15), then it has a binary term operation destroying θ.

(b) If $\mathbb{A}$ is simple and has a term operation that is one of the operations (8), (11), or (16), then $\mathbb{A}$ has a binary term operation destroying θ or an operation $g(x, y, z)$ such that

—each of the operations $g(x, y, 2)$, $g(x, 2, y)$, $g(2, x, y)$ on $\{0,1\}$ either preserves the set $\{0,1\}$ or is the constant operation 2 or has the operation table

	0	1
0	0	2
1	2	1

,

—each of the operations $g(x, 2, 2)$, $g(2, x, 2)$, $g(2, 2, x)$ on $\{0,1\}$ either preserves the set $\{0,1\}$ or is the constant operation 2.

Proof. Since $\mathbb{A}$ is simple, there are an operation $f(x_1, \dots, x_n)$ and tuples $\mathbf{a}, \mathbf{b} \in A^n$ such $(\mathbf{a}[i], \mathbf{b}[i]) \in \theta$, but $(f(\mathbf{a}[1], \dots, \mathbf{a}[n]), f(\mathbf{b}[1], \dots, \mathbf{b}[n])) \notin \theta$. It is not hard to see that f, $\mathbf{a}, \mathbf{b}$ can be chosen such that $\mathbf{a}, \mathbf{b}$ differ only in one coordinate position. Without loss of generality let $\mathbf{a}[1] \neq \mathbf{b}[1]$, $\mathbf{a}[2] = \mathbf{b}[2] = \dots = \mathbf{a}[p] = \mathbf{b}[p] = 0$, $\mathbf{a}[p+1] = \mathbf{b}[p+1] = \dots = \mathbf{a}[r] = \mathbf{b}[r] = 1$, $\mathbf{a}[r+1] = \mathbf{b}[r+1] = \dots = \mathbf{a}[n] = \mathbf{b}[n] = 2$. Then, the operation

$$h(x, y, z, t) = f(x, \underbrace{y, \dots, y}_{p-1 \text{ times}}, \underbrace{z, \dots, z}_{r-p \text{ times}}, \underbrace{t, \dots, t}_{n-r \text{ times}})$$

also destroys θ.

Notice that if $(h(0,0,0,2), h(1,0,1,2)) \notin \theta$, then the operation $h(x, y, x, z)$ destroys θ. Otherwise, since $(h(1,0,1,2), h(0,0,1,2)) \notin \theta$, we have $(h(0,0,0,2), h(0,0,1,2)) \notin \theta$, and the operation $h(y, y, x, z)$ destroys θ. Thus, in each case, there is an operation $g(x, y, z)$ such that $(g(0,1,2), g(1,1,2)) \notin \theta$.

Consider the operation $g'(x, y) = g(x, y, 2)$. Since $(g'(0,1), g'(1,1)) \notin \theta$, we have 8 cases. Let f denote one of the operations listed in Lemma 10.

Case 1. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 0/1 & \\ 1 & & 2 \end{array}$, or $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 2 & \\ 1 & & 0/1 \end{array}$.

In this case, $g(x, x, y)$ destroys θ, and we have a binary operation with this property.

Case 2. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 0 & 2 \\ 1 & & 0 \end{array}$.

For the operation $h(x, y) = g(g(x, x, y), x, y)$, we have $h(0,2) = 0$, $h(1,2) = 2$. We get a binary operation destroying θ.

Case 3. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 1 & 2 \\ 1 & & 1 \end{array}$.

For the operation $h(x, y) = g(x, g(x, x, y), y)$, we have $h(0,2) = 2$, $h(1,2) = 1$. We again get a binary operation destroying θ.

Case 4. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 0/1 & 2 \\ 1 & 0/1 & 0/1 \end{array}$.

If f is one of (8), (10), (11), (12), (14), or (16), then $f(0,2) \in \{0,1\}$, $f(1,2) = 0$. Hence, for the operation $h(x, y) = g(f(x, y), x, y)$, we have $h(1,2) = g(0,1,2) = 2$ and $h(0,2) = g(0/1, 0, 2) \in \{0,1\}$, that is, h destroys θ. If f is one of (7), (13), or (15), then $f(0,2) = f(1,2) = 1$. Therefore, for the operation $h(x, y) = f(x, f(x, y), y)$, we have $h(0,2) = g(0,1,2) = 2$ and $h(1,2) = g(1,1,2) \in \{0,1\}$, that is, again h destroys θ.

Case 5. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 2 & 0/1 \\ 1 & 2 & 2 \end{array}$.

This case is quite analogous to the previous one.

Case 6. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 2 & 0/1 \\ 1 & 0/1 & 2 \end{array}$.

If f is one of (10), (12), (14), then $f(0,2) = f(1,2) = 0$. Hence, for the operation $h(x, y) = g(f(x, y), x, y)$, we have $h(0,2) = g(0,0,2) = 2$ and

$h(1,2) = g(0,1,2) \in \{0,1\}$; thus, h destroys θ. Analogously, if f is one of (7), (13), or (15), then $f(0,2) = f(1,2) = 1$. Therefore, for h obtained in the same way, we have $h(0,2) \in \{0,1\}$, $h(1,2) = 2$, that is, h destroys θ. Finally, if $f(0,2) = 1$, $f(1,2) = 0$, then, by substituting $g(f(x,z), y, z)$, we obtain Case 8.

Case 7. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 1 & 2 \\ 1 & 2 & 0 \end{array}$.

The operation $g(g(x,x,z), g(y,y,z), z)$ satisfies the conditions of the Case 8.

Case 8. $\begin{array}{c|cc} g' & 0 & 1 \\ \hline 0 & 0 & 2 \\ 1 & 2 & 1 \end{array}$.

If $f(0,2) = f(1,2) = 0$, that is, f is one of the operations (10), (12), or (14), then the operation $h(x,y) = g(f(x,y), x, y)$ satisfies the conditions $h(0,2) = g(0,0,2) = 0$, $h(1,2) = g(0,1,2) = 2$; therefore, it destroys θ. If $f(0,2) = f(1,2) = 1$, that is, f is one of the operations (7), (13), or (15), then, h satisfies the conditions $h(0,2) = g(1,0,2) = 2$, $h(1,2) = g(1,1,2) = 1$, and again destroys θ.

Finally, if $f(0,2) = 1$, $f(1,2) = 0$, that is, f is one of the operations (8), (11), or (16), then each of the operations $g(2,y,z)$, $g(x,2,z)$ either satisfies the same conditions as $g(x,y,2)$ does or preserves $\{0,1\}$ or is the constant operation 2 or a binary operation destroying θ can be derived. Analogously, each of the operations $g(2,2,x)$, $g(2,x,2)$, $g(x,2,2)$ either preserves $\{0,1\}$ or is the constant operation 2 or a binary operation destroying θ can be derived. The lemma is proved. □

Now we use the obtained operation to infer one of the 10 properties for $\mathbb{A}$.

LEMMA 10. *If $\mathbb{A}$ has a binary operation destroying the equivalence relation θ and one of the operations* (7), (8), (10), (11), (12), (13), (14), (15), *or* (16)*, then either* 2 *is a zero-element with respect to some binary term operation of $\mathbb{A}$, or $\mathbb{A}$ has a term operation that is either a binary conservative commutative operation, or a zero-operation, or a* $(2-a)$*-operation,* $a \in \{0,1\}$*, or a* $(1-2)$*-operation and* $\{0,2\}$ *is a subuniverse of $\mathbb{A}$.*

PROOF. Let g be the term operation destroying θ, $(g(0,2), g(1,2)) \notin \theta$, and f denote one of the operations listed.

Suppose first that $g(0,2) = 0$, $g(1,2) = 2$, and consider four cases.

Case 1. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & & & 2 \end{array}$.

Subcase 1.1. $g(2,0), g(2,1) \in \{0,1\}$.

The operation $g(x, g(x,y))$ is either a zero-operation or a $(2-0)$-operation.

Subcase 1.2. $g(2,0) = 0$, $g(2,1) = 2$. In this case, g is a zero-operation itself.

Subcase 1.3. $g(2,0) = 2$, $g(2,1) = 1$.

If f is one of the operations (7), (8), (10), (11), (12), or (13), that is, $f|_{\{0,1\}}$ is a semilattice operation, then $f(g(x,y), x)$ satisfies the conditions of Case 3 or Case 4. Otherwise, by substituting $g(x, f(y,x))$, we get Subcase 1.4 or Subcase 1.1.

Subcase 1.4. All other possibilities.

The operation $g(x, g(x,y))$ is a zero operation.

Case 2. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 \\ 2 & & & 2 \end{array}$.

Subcase 2.1. $0 \in \{g(2, 0), g(2, 1)\}$. In this case, g is a zero-operation.

Subcase 2.2. $g(2, 0) = g(2, 1) = 1$ or $g(2, 0) = 1$, $g(2, 1) = 2$.
If $g(2, 0) = g(2, 1) = 1$, then $g(x, g(y, x))$ is a $(2-1)$-operation. Then, if $g(2, 0) = 1$, $g(2, 1) = 2$, then by substituting $g(x, g(x, y))$ we get Subcase 2.4.

Subcase 2.3. $g(2, 0) = 2$, $g(2, 1) = 1$.
In this case, if $f(0, 2) = f(1, 2)$, that is, f is one of (7), (10), (12), (13), (14), or (15), then $g(f(x, y), y)$ is a zero-operation. Otherwise, the operation $g(x, f(y, x))$ satisfies the conditions of Subcase 2.2.

Subcase 2.4. $g(2, 0) = g(2, 1) = 2$.
Since g is a $(1-2)$-operation, if $\{0, 2\}$ is a subuniverse, then we are done. Otherwise, by Lemma 7, one of the operations (1), (5), (7), (8), (9), (11), (13), or (15) is a term operation of $\mathbb{A}$. Therefore, either one of the operations (7), (8), (11), (13), (15), or (16) is a term operation of $\mathbb{A}$; or the conditions of Lemma 8 or Lemma 9 hold and there is a term zero-operation of $\mathbb{A}$ or $\mathbb{A}$ satisfies one of the following conditions: $\{0, 1\}$- or $\{0, 2\}$-extendibility, $(2 - 1)$-replacement property, $\{0, 1\}$-rectangularity. In the former case, denote this term operation by h. If h is one of (7), (13), or (15), that is, $h(0, 2) = h(1, 2) = 1$, then 2 is a zero-element with respect to the operation $g(h(x, y), y)$. If h is one of (8), (11), or (16), then $g(h(x, y), y)$ satisfies the conditions of one of Cases 5–8.

Case 3. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & & & 2 \end{array}$.

If $g(2, 0) \neq 2$ or $g(2, 1) \neq 2$, then g is a zero-operation. So, suppose that $g(2, 0) = g(2, 1) = 2$. Since g is the operation (2), if $\{0, 2\}$ is a subuniverse, then by Lemma 9 either a zero-operation is a term operation of $\mathbb{A}$ or $\mathbb{A}$ is $\{0, 2\}$-rectangular or $\mathbb{A}$ satisfies the $\{0, 2\}$-extendibility property.

If $\{0, 2\}$ is not a subuniverse, then, as in Subcase 2.4, either the conditions of Lemma 8 or Lemma 9 hold, or one of (7), (8), (11), (13), (15), or (16) is a term operation of $\mathbb{A}$. In the latter case, denote this operation by h. If h is one of (7), (13), or (15), then 2 is a zero-element with respect to the operation $g(h(x, y), y)$. If f is one of (8), (11), or (16), then $g(h(x, y), y)$ satisfies the conditions of one of Cases 5–8.

Case 4. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 2 \\ 2 & & & 2 \end{array}$.

If $g(2, 1) = 1$, then g is a zero-operation. If $g(2, 1) = 0$, then $g(g(x, y), y)$ is a zero-operation. Finally, let $g(2, 1) = 2$. Then, if $g(2, 0) = 2$, then g is a zero operation; and if $g(2, 0) = 1$, then $g(x, g(x, y))$ is a zero-operation. If $g(2, 0) = 0$, then g is a conservative commutative operation.

Case 5. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 0 & 1 & 0 \\ 2 & & & 2 \end{array}$.

In this case, either g itself or the operation $g(x, g(x, y))$ is a zero-operation or a $(2 - 0)$-operation.

Case 6. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 0 \\ 2 & & & 2 \end{array}$.

Subcase 6.1. $g(2, 0) = 0$, $g(2, 1) = 0$.
Set $h(x, y) = g(x, g(x, y))$; its operation table is $\begin{array}{c|ccc} h & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 2 \end{array}$. Then the $h(x, h(y, x))$ is a $(2 - 0)$-operation.

Subcase 6.2. $g(2, 1) = 2$, $g(2, 0) \in \{1, 2\}$.
The operation $g(g(x, y), y)$ is a zero-operation.

Subcase 6.3. All other possibilities.
In this case, for the operation $h(x, y) = g(x, g(x, y))$, we have $1 \in \{h(2, 0), h(2, 1)\}$, and $h(2, 1) = 1$. Therefore, h is a zero operation.

Case 7. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 2 & & & 2 \end{array}$.

Subcase 7.1. $g(2, 0) = 0$. The operation g is a zero-operation.

Subcase 7.2. $g(2, 0) = 1$.
If $g(2, 1) \in \{0, 2\}$, then $g(x, g(x, y))$ is a zero-operation. If $g(2, 1) = 1$, then $g(g(y, x), y)$ is a zero-operation.

Subcase 7.3. $g(2, 0) = 2$.
If $g(2, 1) = 2$, then 2 is a zero-element with respect to $g(g(x, y), y)$. If $g(2, 1) = 0$, then, for the operation $h(x, y) = g(x, g(x, y))$, we have $h(2, 1) = 2$, and we get the previous case. Finally, if $g(2, 1) = 1$, then $g(g(y, x), y)$ falls into one of the previous cases.

Case 8. $\begin{array}{c|ccc} g & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 0 \\ 2 & & & 2 \end{array}$.

If $g(2, 0) = 2$, $g(2, 1) = 0$, then g is the operation (6), and, by Lemma 8, $\mathbb{A}$ satisfies the $\{1, 2\}$-extendibility property. If $g(2, 0) = g(2, 1) = 2$, then g is a zero-operation; if $g(2, 1) = 2$, $g(2, 0) \in \{0, 1\}$, then $g(g(x, y), y)$ is a zero-operation. Furthermore, if $g(2, 0) = g(2, 1) = 0$, then $g(x, g(y, x))$ is a zero-operation. In all other cases, $g(x, g(x, y))$ is a zero-operation.

The proof in the case when $1 \in \{g(0, 2), g(1, 2)\}$ is quite similar. $\square$

LEMMA 11. *If $\mathbb{A}$ satisfies the conditions of Lemma* 10 *and has no binary operation destroying θ, then $\mathbb{A}$ has a term operation which is either a zero-operation or a* $(2 - 0)$*-operation or a* $(2 - 1)$*-operation or the operation* (1).

PROOF. Let f denote the operation (16), and g the ternary operation satisfying the conditions of Lemma 10. Consider the operation $g'(x, y) = g(x, x, y)$; its operation table is $\begin{array}{c|ccc} g' & 0 & 1 & 2 \\ \hline 0 & 0 & & 0 \\ 1 & & 1 & 1 \\ 2 & & & 2 \end{array}$. The restriction of g' onto $\{0, 1\}$ is either a projection or a semilattice operation. If $g'(2,0) = g'(2,1) \in \{0,1\}$ and $g'|_{\{0,1\}}$ is the first projection, then g' is either $(2-0)$- or $(2-1)$-operation. If $g'|_{\{0,1\}}$ is the second projection, then $g'(y, g'(y,x))$ is a zero operation. In the case when $g'|_{\{0,1\}}$ is a semilattice operation, g' is a zero-operation. Furthermore, if $g'(2,0) = 0$, $g'(2,1) = 1$, then either g' itself or $g'(y,x)$ is the operation (1), or g' is a zero-operation. In the case $g'(2,0) = 1$, $g'(2,1) = 0$, the operation $g'(x, g(x,y))$ is either a zero-operation or the operation (1) or the operation (1) with permuted variables.

The only case remaining to consider is $g'(2,0) = g'(2,1) = 2$. If the restriction of g' onto $\{0,1\}$ is a semilattice operation, then $f(g(x,y),y)$ is one of the operations (8), (11). By Lemma 10, there exists a binary term operation of $\mathbb{A}$ that destroys the equivalence relation θ, a contradiction with the conditions of Lemma 11. If $g'|_{\{0,1\}}$ is the second projection, then consider the operation $h(x,y,z) = g(x,y,g(z,z,x))$. It is not hard to check that h satisfied the conditions applied to g in Lemma 11, but the operation table of $h(x,x,y)$ is $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$. Finally, if $g'|_{\{0,1\}}$ is the first projection, then the operation table of $h(x,y) = g(x, f(x,y), y)$ is $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{array}$; therefore, h is a zero-operation. □

6.5. CONSERVATIVE ALGEBRAS. An algebra is called *conservative* if every its subset is a subuniverse.

If $\mathbb{B}$ is a 2-element subalgebra of $\mathbb{A}$, then $\mathbb{B}$ is tractable; therefore, by Schaefer's theorem, there is a term operation f of $\mathbb{A}$ such that $f|_B$ is either a semilattice or majority or affine operation. We have 5 cases depending on operations of which kind provide the tractability of 2-element subalgebras.

6.5.1. $\mathbb{A}$ has a term operation f whose restriction on a 2-element subuniverse is a semilattice operation. Suppose that f is a semilattice operation on $B = \{0,1\}$ and $f(0,1) = f(1,0) = 1$. Then, by Lemma 6(1), the restriction of f onto any other 2-element subuniverse is either a semilattice operation or a projection. Moreover, by Lemma 6(2), replacing f with $f(x, f(y,x))$, all the projections may be assumed to be first projections. We consider three subcases.

Case 1. f is a semilattice operation on all three 2-element subuniverses.
In this case, f is a commutative conservative binary operation.

Case 2. f is a semilattice operation on two of the 2-element subuniverses and it is the first projection on the third one.

Subcase 2.1. $f|_{\{0,2\}}$ is a semilattice operation.
Then, the Cayley table of f is one of the following $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 2 & 2 \end{array}$, $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$. In the first case, f is a zero-operation; therefore, by Lemma 4, $\mathbb{A}$ satisfies the partial zero property. In the second case, f is the operation (2); hence, by Lemma 9, $\mathbb{A}$ satisfies the $\{1,2\}$-rectangularity property.

Subcase 2.2. $f|_{\{1,2\}}$ is a semilattice operation.

Then, the Cayley table of f is one of the following $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 2 \end{array}$, $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{array}$. In the first case, 1 is a zero-element with respect to f. In the second case, f is a zero-operation. Therefore, in both cases, $\mathbb{A}$ satisfies the partial zero property.

Case 3. f is a semilattice operation on one of the 2-element subuniverses, and is the first projection on the remaining ones.

In this case, the Cayley table of f is $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$; therefore f is a $(0-1)$-operation. By Lemma 3, $\mathbb{A}$ satisfies the $(0-1)$-replacement property.

Before considering the remaining cases, we observe some properties of ternary term operations of $\mathbb{A}$ in the case when no 2-element subalgebra of $\mathbb{A}$ has a semilattice term operation. An operation $f(x, y, z)$ is said to be *minority* if the identities $f(x, x, y) = f(x, y, x) = f(y, x, x) = y$ hold. There is only one minority operation on a 2-element set, namely, the affine operation $x - y + z$. We show that the operations of $\mathbb{A}$ yielding majority and minority restrictions on its subuniverses can be combined into a single operation.

LEMMA 12. *Let $\mathbb{A}$ be such that the restriction of every its binary term operation onto every 2-element subuniverse is a projection. Then, there exists a term operation $h(x, y, z)$ of $\mathbb{A}$ such that, for any 2-element subuniverse B of $\mathbb{A}$, the restriction $h|_B$ is the majority operation if there is a term operation f of $\mathbb{A}$ such that $f|_B$ is the majority operation, and $h|_B$ is the minority operation otherwise.*

PROOF. Let $B \subseteq A$, $|B| = 2$, $f' = f|_B$, and $f_1(x, y) = f'(x, y, y)$, $f_2(x, y) = f'(y, x, y)$, $f_3(x, y) = f'(y, y, x)$. If one of these operations is not a projection, then it must be a semilattice operation, a contradiction to the assumptions made.

CLAIM 1. If $f_i|_B(x, y) = y$ for some i and f' is not a projection, then there is a term operation g of $\mathbb{A}$ such that $g|_B$ is the majority operation.

Let $f_1(x, y) = y$. Then there are 3 possibilities. (a) f' is a majority operation and we are done; (b) $f_2(x, y) = x$, $f_3(x, y) = y$ or $f_2(x, y) = y$, $f_3(x, y) = x$, and f' is a projection, which is impossible; (c) $f_2(x, y) = f_3(x, y) = x$. In this case, set $g(x, y, z) = f(f(x, y, z), y, z)$. We have

$$\begin{aligned} g|_B(x, y, y) &= f'(f'(x, y, y), y, y) = f'(y, y, y) = y, \\ g|_B(y, x, y) &= f'(f'(y, x, y), x, y) = f'(x, x, y) = y, \\ g|_B(y, y, x) &= f'(f'(y, y, x), y, x) = f'(x, y, x) = y, \end{aligned}$$

that means that $g|_B$ is a majority operation.

Denote by B_1, B_2, B_3 the sets $\{0, 1\}, \{0, 2\}, \{1, 2\}$ respectively. Using Claim 1 it is not hard to conclude that there are ternary term operations g_1, g_2, g_3 of $\mathbb{A}$ such that $g_i|_{B_i}$ is the majority operation if there is a term operation f of $\mathbb{A}$ such that $f|_{B_i}$ is the majority operation, $g_i|_{B_i}$ is the minority otherwise, and $g_i|_{B_j}$ is either the majority operation, or the minority operations, or the first projection for $j \neq i$.

CLAIM 2. g_1, g_2, g_3 can be chosen such that their restrictions on each 2-element subset is either the majority or the minority operation.

To prove the claim it is enough to notice that, for any $i, j \in \{1, 2, 3\}$, the operation

$$g_{ij}(x, y, z) = g_j(g_i(x, y, z), g_i(y, z, x), g_i(z, x, y))$$

is either the majority or minority operation on B_i, B_j, and $g_{ij}|_{B_i} = g_i|_{B_i}$. Then, we may replace g_1, g_2, g_3 with

$$g_{23}(g_{12}(x, y, z), g_{12}(y, z, x), g_{12}(z, x, y)) \quad (1)$$
$$g_{31}(g_{23}(x, y, z), g_{23}(y, z, x), g_{23}(z, x, y)) \quad (2)$$
$$g_{12}(g_{31}(x, y, z), g_{31}(y, z, x), g_{31}(z, x, y)) \quad (3)$$

Finally, if all g_1, g_2, g_3 are minority operations on each 2-element subset, then any of them suits as h. Suppose that $g_1|_{B_1}, g_2|_{B_2}$ are majority operations, but $g_1|_{B_2}$ is the minority operation. Then, for $g'(x, y) = g_1(x, x, y)$, we have $g'|_{B_1}(x, y) = x$, $g'|_{B_2}(x, y) = y$; and the operation $g''(x, y, z) = g'(g_1(x, y, z), g_2(x, y, z))$ is the majority operation on both B_1, B_2. If $g''|_{B_3} \neq g_3|_{B_3}$, then we repeat this procedure for B_1, B_3. $\square$

6.5.2. All three 2-element subalgebras have a majority term operation, but no semilattice term operation. By Lemma 12, there is a term operation f that is a majority operation on each 2-element subset. This means that f is a majority operation on A; therefore, $\mathbb{A}$ has a majority term operation.

6.5.3. All three 2-element subalgebras have a minority term operation, but no semilattice or majority operation. By a similar reason, $\mathbb{A}$ has a Mal'tsev term operation.

6.5.4. Two of the 2-element subalgebras have a majority term operation, but no semilattice operation; and the third subalgebra has a minority term operation, but neither a semilattice nor majority operation. By Lemma 12, there is a term operation f of $\mathbb{A}$ such that $f|_{B_1}$ is the minority operation and $f|_{B_2}, f|_{B_3}$ are majority operations. Then, the operation $g(x, y) = f(x, y, y)$ is the first projection on B_1 and the second projection on B_2, B_3.

LEMMA 13. *The algebra $\mathbb{A}$ satisfies the B_2-semisplitting property.*

PROOF. For $I, J \subseteq \underline{n}$, where $I \cap J = \emptyset$, and $\mathbf{a} \in R_I$, $\mathbf{b} \in R_J$, we write $(\mathbf{a}, \mathbf{b})$ for the $|I| + |J|$-tuple $\mathbf{c} = (\mathbf{c}[i])_{i \in I \cup J}$ with $\mathbf{c}[i] = \mathbf{a}[i]$ if $i \in I$, and $\mathbf{c}[i] = \mathbf{b}[i]$ if $i \in J$.

Let $R \in \mathsf{Inv}\ F$ be an irreducible (n-ary) relation, $W = \{i \in \underline{n} \mid R_i = A\}$, and $W_j = \{i \in \underline{n} \mid R_i = B_i\}$, $j = 1, 2, 3$. We have to prove that

$$\emptyset \neq \left(R_W \cap B_2^{|W|}\right) \times R_{W_2 \cup W_3} \times R_{W_1} \subseteq R.$$

CLAIM 1. *For any $\mathbf{a} \in R_{W_1}$, $i \in W \cup W_2 \cup W_3$, and any $a \in R_i$, the tuple $(\mathbf{a}, a)$ belongs to $R_{W_1 \cup \{i\}}$.*

There is b such that $(\mathbf{a}, b) \in R_{W_1 \cup \{i\}}$ and $\mathbf{b} \in R_{W_1}$ such that $(\mathbf{b}, a) \in R_{W_1 \cup \{i\}}$. If $b \in \{0, 1\}$, then take a tuple of the form $(\mathbf{c}, 2) \in R_{W_1 \cup \{i\}}$. The tuple $\binom{\mathbf{a}}{2} = g\left(\binom{\mathbf{a}}{b}, \binom{\mathbf{c}}{2}\right)$ belongs to $R_{W_1 \cup \{i\}}$. If $a \in \{0, 1\}$ then $\binom{\mathbf{a}}{a} = g\left(\binom{\mathbf{a}}{2}, \binom{\mathbf{b}}{a}\right)$.

CLAIM 2. *For any $i, j \in W$ and any $(\mathbf{a}, a) \in R_{W_1 \cup \{i\}}$ with $a \in B_2$, there is $b \in B_2$ such that $(\mathbf{a}, a, b) \in R_{W_1 \cup \{i,j\}}$.*

Let us denote $R_{W_1 \cup \{i,j\}}$ by R''. Since $(\mathbf{a}, a) \in R_{W_1 \cup \{i\}}$, there is c such that $(\mathbf{a}, a, c) \in R''$. If $c \in B_2$ then we are done, so, suppose that $c = 1$.

Case 1. $a = 2$.
By Claim 1, there is $d \in A$ such that $(\mathbf{a}, d, 0) \in R''$. As is easily seen, the tuple $g\left(\begin{pmatrix} \mathbf{a} \\ d \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ 2 \\ 1 \end{pmatrix}\right)$ is as required.

Case 2. $a = 0$.
By Claim 1, there are $c, d \in A$ such that $(\mathbf{a}, c, 0), (\mathbf{a}, d, 2) \in R''$. If $c \in \{0, 2\}$ or $d \in \{0, 1\}$, then $g\left(\begin{pmatrix} \mathbf{a} \\ c \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ 0 \\ 1 \end{pmatrix}\right)$ or $g\left(\begin{pmatrix} \mathbf{a} \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ d \\ 2 \end{pmatrix}\right)$ is as required. Therefore, we may assume that $c = 1, d = 2$.

Since R is irreducible, there is $\mathbf{b} \in R_{W_1}$ such that $(\mathbf{b}, c', d') \in R''$ where $(c', d') \in \{(0/1, 2), (2, 0/1), (0, 0), (1, 1)\}$. In the first case, $g\left(\begin{pmatrix} \mathbf{a} \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{b} \\ 0/1 \\ 2 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 0 \\ 2 \end{pmatrix}$ is as required. In the second case, we have

$$g\left(\begin{pmatrix} \mathbf{a} \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} \mathbf{b} \\ 2 \\ 0/1 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 2 \\ 0/1 \end{pmatrix}, \quad g\left(\begin{pmatrix} \mathbf{a} \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ 2 \\ 0/1 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 2 \\ 0 \end{pmatrix},$$
$$g\left(\begin{pmatrix} \mathbf{a} \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ 0 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 0 \\ 0 \end{pmatrix};$$

and we get the required tuple. In the third case, $g\left(\begin{pmatrix} \mathbf{a} \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} \mathbf{b} \\ 0 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 0 \\ 0 \end{pmatrix}$ is as required. Finally, in the last case,

$$g\left(\begin{pmatrix} \mathbf{a} \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} \mathbf{b} \\ 1 \\ 1 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 1 \\ 1 \end{pmatrix} \text{ and } f\left(\begin{pmatrix} \mathbf{a} \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{a} \\ 1 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{a} \\ 0 \\ 0 \end{pmatrix};$$

we again obtain the required tuple.

CLAIM 3. *For any $i_1, i_2 \in W \cup W_2 \cup W_3$, any $(a_{i_1}, a_{i_2}) \in R_{\{i_1, i_2\}}$ such that $a_{i_j} \in B_2$ if $i_j \in W \cup W_2$, any $I \subseteq (W \cup W_1 \cup W_2) - \{i_1, i_2\}$ and any $\mathbf{a} \in R_{W_1}$, a tuple $\mathbf{b}$ with*

$$\mathbf{b}[i] = \begin{cases} a_{i_j}, & \text{if } i = i_j,\ j = 1, 2, \\ \mathbf{a}[i], & \text{if } i \in W_1 \end{cases}$$

and $\mathbf{b}[i] \in B_2$ for $i \in i \cap (W \cap W_2)$ belongs to $R_{W_1 \cup I \cup \{i_1, i_2\}}$.

The base case of induction, $I = \emptyset$, follows from Claim 2. Indeed, by Claim 2, there is c such that $(\mathbf{a}, a_{i_1}, c) \in R_{W_1 \cup \{i_1, i_2\}}$ and $c \in B_2$ whenever $i_1 \in W \cup W_2$; and there is $\mathbf{c}$ such that $(\mathbf{c}, a_{i_1}, a_{i_2}) \in R_{W_1 \cup \{i_1, i_2\}}$. Then $g\left(\begin{pmatrix} \mathbf{a} \\ a_{i_1} \\ c \end{pmatrix}, \begin{pmatrix} \mathbf{c} \\ a_{i_1} \\ a_{i_2} \end{pmatrix}\right)$ is the required tuple.

Let $I \neq \emptyset$. Without loss of generality, let $i_1 = 1, i_2 = 2$, and $I = \{3, \dots, k\}$. Suppose that the claim holds for all $I \subseteq W \cup W_2 \cup W_3$ with $|I| \le k-3$. Hence, there is $b \in A$ such that $\mathbf{c} = (a_1, a_2, \dots, a_{k-1}, b, \mathbf{a}) \in R_{\{1,2\}\cup I\cup W_1}$, where $a_3, \dots, a_{k-1}$ satisfy the conditions of the claim. If $k \in W_2 \cup W_3$ or $b \in B_2$, then we are done; so let $k \in W$ and $b = 1$.

Case 1. $a_1 = 2$.
By Claim 1, $(0, \mathbf{a}) \in R_{\{k\}\cup W_1}$; therefore $\mathbf{d} = (c, b_2, \dots, b_{k-1}, 0, \mathbf{a}) \in R_{\{1,2\}\cup I\cup W_1}$ for certain $b_2, \dots, b_{k-1}$ satisfying the conditions of the claim. Set $\mathbf{e} = g(\mathbf{c}, \mathbf{b})$. Since $\{\mathbf{d}[i], \mathbf{c}[i]\} \neq \{0, 1\}$ for all $i \in \{2, \dots, k-1\}$, we have $\mathbf{e}[i] = \mathbf{c}[i] = a_i$. Then, $\mathbf{e}[1] = 2$ because $a_1 = 2$, and $\mathbf{e}[k] = g(0, 1) = 0$.

Case 2. $a_1 = 0$ or $a_1 = 1$ if $1 \in W_3$.
The tuple $(2, \mathbf{a})$ belongs to $R_{\{1\}\cup W_1}$ and, by Case 1, can be extended to $\mathbf{d} = (2, c_2, \dots, c_{k-1}, 0, \mathbf{a}) \in R_{\{1,\dots,k\}\cup W_1}$. The tuple $\mathbf{e} = g(\mathbf{d}, \mathbf{c})$ is as required. Indeed, if $i \in \{2, \dots, k-1\}$, then $\{\mathbf{c}[i], \mathbf{d}[i]\} \neq \{0, 1\}$; therefore, $\mathbf{e}[i] = \mathbf{c}[i] = a_i$, $\mathbf{e}[1] = g(2, 0) = 0$ (or $\mathbf{e}[1] = g(2, 1) = 1$ if $1 \in W_3$), $\mathbf{e}[k] = g(0, 1) = 0$.

CLAIM 4. *For any $I \subseteq W \cup W_2 \cup W_3$, any $\mathbf{b} \in R_I$ such that $\mathbf{b}[i] \in B_2$ whenever $i \in W$ and any $\mathbf{a} \in R_{W_1}$, the tuple $(\mathbf{b}, \mathbf{a})$ is in $R_{I\cup W_1}$.*

By Claim 3, there are tuples $(\mathbf{b}, \mathbf{c}), (\mathbf{d}, \mathbf{a}) \in R_{I\cup W_1}$ with $\mathbf{d}[i] \in B_2$ whenever $i \in W$. It is easy to see that $g\left(\binom{\mathbf{d}}{\mathbf{a}}, \binom{\mathbf{b}}{\mathbf{c}}\right) = \binom{\mathbf{b}}{\mathbf{a}}$.

To finish the proof of Lemma 13, we just should put $I = W \cup W_2 \cup W_3$ in Claim 4. $\square$

By the assumptions made, f is a majority operation on B_2, and therefore, in this case, $\mathbb{A}$ satisfies the conditions of Theorem 3(5).

6.5.5. One of the 2-element subalgebras has the majority term operation, but no semilattice operation; the two others have the minority term operation, but neither a semilattice nor majority operation. By Lemma 12, $\mathbb{A}$ has a term operation f which is the majority operation on B_1 and the minority operation on B_2, B_3. Then, the operation $g(x, y) = f(x, x, y)$ is the first projection on B_1 and the second projection on B_2, B_3.

First, we show that every binary relation invariant with respect to the term operations of $\mathbb{A}$ has a very restricted form. Let θ denote the equivalence relation whose classes are $0^\theta = \{0, 1\}$ and $2^\theta = \{2\}$. The class containing an element $a \in A$ will be denoted by a^θ; and, for an n-ary relation R, we set $R^\theta = \{(a_1^\theta, \dots, a_n^\theta) \mid (a_1, \dots, a_n) \in R\}$.

LEMMA 14. *Let $R \in \mathsf{Inv}\, F$ be a binary relation such that $R_1 = R_2 = A$. Then, R is either the identity relation or A^2 or the graph of the nonidentity bijection with the fixed point 2 or $R^\theta \in \{\mu_1, \mu_2\}$, where*

$$\mu_1 = \begin{pmatrix} 0^\theta & 2^\theta \\ 0^\theta & 2^\theta \end{pmatrix}, \quad \mu_2 = \begin{pmatrix} 0^\theta & 2^\theta \\ 2^\theta & 0^\theta \end{pmatrix}.$$

PROOF. Suppose first that R is the graph of a mapping φ and $\varphi(2) \neq 2$, say, $\varphi(2) = 0$. Then, denoting $b = \varphi^{-1}(1)$ we have $g\left(\binom{b}{1}, \binom{2}{0}\right) = \binom{2}{1} \in R$, which contradicts the assumptions made. Thus, $\varphi(2) = 2$.

Suppose that R is neither the graph of a bijective mapping nor $R^\theta \in \{\mu_1, \mu_2\}$. Then, R contains one of the tuples $\binom{0}{2}, \binom{1}{2}, \binom{2}{0}, \binom{2}{1}$, and either the tuple $\binom{2}{2}$ or a tuple of the form $\binom{a}{b}$, where $a, b \in \{0, 1\}$.

If $\binom{2}{0} \in R$, then there is $a \in A$ such that $\binom{a}{1} \in R$; consequently, $\binom{2}{1} = g\left(\binom{a}{1}, \binom{2}{0}\right) \in R$. Analogously, if $\binom{2}{1} \in R$, then $\binom{2}{0} \in R$; and if one of the tuples $\binom{0}{2}, \binom{1}{2}$ belongs to R, then the other also belongs to R. Since $R_1 = A$, we have two cases to consider.

Case 1. $\binom{0}{2}, \binom{1}{2}, \binom{2}{2} \in R$.

Since $R_2 = A$, there is $\binom{a}{b}$ in R, where $b \in \{0, 1\}$. Then $\binom{2}{b} = f\left(\binom{2}{2}, \binom{a}{2}, \binom{a}{b}\right) \in R$; therefore, $\binom{2}{0}, \binom{2}{1} \in R$. Moreover, for any $c, d \in A$, we have $\binom{c}{d} = f\left(\binom{c}{2}, \binom{2}{2}, \binom{2}{d}\right) \in R$; hence, $R = A^2$.

Case 2. $\binom{0}{2}, \binom{1}{2}, \binom{2}{0}, \binom{2}{1} \in R$.

In this case, there is $\binom{a}{b}$ with $a, b \in \{0, 1\}$, therefore, $\binom{2}{2} = f\left(\binom{a}{2}, \binom{a}{b}, \binom{2}{b}\right) \in R$, and we get the previous case. $\square$

We consider two cases depending on whether $\mathbb{A}$ has a nontrivial congruence.

Case I. $\mathbb{A}$ is simple.

In this case, θ does not belong to $\mathsf{Inv}\ F$; moreover, any binary relation R such that $R^\theta \in \{\mu_1, \mu_2\}$ and R is not the graph of a mapping does not belong to $\mathsf{Inv}\ F$ as well. Indeed, every such relation R is of the form $\begin{pmatrix} 0 & 1 & 2 & 2 \\ 2 & 2 & 0 & 1 \end{pmatrix}$ or $\begin{pmatrix} a & a & b & 2 \\ c & d & d & 2 \end{pmatrix}$, where $\{a, b\} = \{c, d\} = \{0, 1\}$. Then, $\theta = R \circ R^{-1} \in \mathsf{Inv}\ F$ that contradicts the assumptions made. This imposes strong restrictions on all relations invariant with respect to the term operations of $\mathbb{A}$.

LEMMA 15. *Let $R \in \mathsf{Inv}\ F$ be an (n-ary) relation such that $R_1 = \cdots = R_n = A$ and $R_{\{i,j\}} = A^2$ for any $i, j \in \{1, \ldots, n\}$. Then, $R = A^n$.*

PROOF. We prove the lemma by induction. To prove the base case $n = 3$, take a ternary relation R from $\mathsf{Inv}\ F$, and, for $a \in A$, denote $R^a = \{(b, c) \in A^2 \mid (a, b, c) \in R\}$. Each R^a satisfies the conditions of Lemma 14; therefore, it is either the graph of a bijection with the fixed point 2, or A^2. (Notice that in both cases $\binom{2}{2} \in R^a$.) Since $R^0 \cup R^1 \cup R^2 = A^2$, one of R^0, R^1, R^2 is A^2.

Case 1. $R^2 = A^2$.

For any $a, b, c \in \{0, 1\}$, we have $\begin{pmatrix} a \\ b \\ c \end{pmatrix} = f\left(\begin{pmatrix} a \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ b \\ c \end{pmatrix}\right) \in R$. Therefore, $\{0, 1\}^2 \subseteq R^a$ and, by Lemma 14, $R^a = A^2$. Thus, $R = A^3$.

Case 2. $R^a = A^2, a \in \{0, 1\}$.

In this case, for any $b, c \in \{0, 1\}$, we have $\begin{pmatrix} 2 \\ b \\ c \end{pmatrix} = f\left(\begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} a \\ 2 \\ 2 \end{pmatrix}, \begin{pmatrix} a \\ b \\ c \end{pmatrix}\right) \in R$. Therefore, $R^2 = A^2$ and we get the previous case.

To prove the induction step, suppose that the claim of the lemma holds for $n > 2$ and $R \in \mathsf{Inv}\ F$ is an ($(n+1)$-ary) relation. As before, let $R^a = \{(a_2, \dots, a_{n+1}) \mid (a, a_2, \dots, a_{n+1}) \in R\}$. By the induction hypothesis, $R^a = A^n$ for any $a \in A$; therefore, $R = A^{n+1}$. $\square$

Finally, the conditions established in Lemma 15 on the structure of invariant relations yield that $\mathbb{A}$ satisfies the splitting property.

LEMMA 16. *The algebra $\mathbb{A}$ satisfies the splitting property.*

PROOF. Let $R \in \mathsf{Inv}\ F$ be an (n-ary) relation, and $W = \{i \mid R_i = A\}$, $W_j = \{i \mid R_i = B_j\}$, $i = 1, 2, 3$. We prove that $R = R_W \times R_{W_1} \times R_{W_2 \cup W_3}$, that is even a stronger condition.

Let us prove first that $R = R_{W \cup W_2 \cup W_3} \times R_{W_1}$. Take $\mathbf{a}' \in R_{W \cup W_2 \cup W_3}$, $\mathbf{b}' \in R_{W_1}$ and $\mathbf{a}, \mathbf{b} \in R$ such that $\mathbf{a}_{W \cup W_2 \cup W_3} = \mathbf{a}'$, $\mathbf{b}_{W_1} = \mathbf{b}'$. By Lemma 15, $R_W = A^{|W|}$, therefore, there is $\mathbf{c} \in R$ such that $\mathbf{c}[i] = 2$ for all $i \in W$. For $\mathbf{d} = g(g(\mathbf{b}, \mathbf{c}), \mathbf{a})$ we have

$\mathbf{d}[i] = g(g(\mathbf{b}[i], \mathbf{c}[i]), \mathbf{a}[i]) = g(g(\mathbf{b}[i], 2), \mathbf{a}[i]) = g(2, \mathbf{a}[i]) = \mathbf{a}[i]$ if $i \in W$;
$\mathbf{d}[i] = g(g(\mathbf{b}[i], \mathbf{c}[i]), \mathbf{a}[i]) = \mathbf{a}[i]$ if $i \in W_2 \cup W_3$;
$\mathbf{d}[i] = g(g(\mathbf{b}[i], \mathbf{c}[i]), \mathbf{a}[i]) = g(\mathbf{b}[i], \mathbf{c}[i]) = \mathbf{b}[i]$ if $i \in W_1$.

Then, we prove that $R_{W \cup W_2 \cup W_3} = R_W \times R_{W_2 \cup W_3}$. Let $v \in W$ and $R' = R_{\{v\} \cup W_2 \cup W_3}$. Without loss of generality we may assume that $v = 1$ and $W_2 \cup W_3 = \{2, \dots, k\}$. Notice first that if $\binom{1}{\mathbf{a}} \in R'$, then $\binom{0}{\mathbf{a}} \in R'$, and vice-versa. Indeed, there is $\mathbf{b} \in R'_{W_2 \cup W_3}$ such that $\binom{0}{\mathbf{b}} \in R'$ and $\binom{0}{\mathbf{a}} = g\left(\binom{0}{\mathbf{b}}, \binom{1}{\mathbf{a}}\right) \in R'$.

Furthermore, if $(2, \mathbf{a}), (0, \mathbf{a}) \in R'$ for a certain $\mathbf{a}$, then $(0, \mathbf{b}), (1, \mathbf{b}), (2, \mathbf{b}) \in R'$ for every $\mathbf{b} \in R_{W_2 \cup W_3}$. This follows from the equalities

$$\begin{pmatrix} 2 \\ \mathbf{b} \end{pmatrix} = f\left(\begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} 0 \\ \mathbf{a} \end{pmatrix}, \begin{pmatrix} 2 \\ \mathbf{a} \end{pmatrix}\right) = f\left(\begin{pmatrix} 1 \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} 1 \\ \mathbf{a} \end{pmatrix}, \begin{pmatrix} 2 \\ \mathbf{a} \end{pmatrix}\right),$$
$$\begin{pmatrix} 0 \\ \mathbf{b} \end{pmatrix} = f\left(\begin{pmatrix} 2 \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} 2 \\ \mathbf{a} \end{pmatrix}, \begin{pmatrix} 0 \\ \mathbf{a} \end{pmatrix}\right).$$

Hence, either $R' = A \times R_{W_2 \cup W_3}$, or $R_{W_2 \cup W_3} = S' \cup S''$ with $S' \cup S'' = \emptyset$ and $R' = (\{2\} \times S') \cup (\{0, 1\} \times S'')$.

In the former case, let $P(x, x_2, \dots, x_k)$ be the predicate corresponding to R'. We set

$$P_Q(y, z) = \exists x_2, \dots, x_k(P(y, x_2, \dots, x_k) \wedge P(z, x_2, \dots, x_k)).$$

The relation Q is the equivalence relation with the classes $\{0, 1\}$, $\{2\}$, that contradicts the simplicity of $\mathbb{A}$.

Finally, for $\mathbf{a} \in R_{W_2 \cup W_3}$, denote $R^{\mathbf{a}} = \{\mathbf{b} \in R_W \mid (\mathbf{b}, \mathbf{a}) \in R_{W \cup W_2 \cup W_3}\}$. By what was proved above, $(R^{\mathbf{a}})_i = A$ for any $i \in W$ and $\mathbf{a} \in R_{W_2 \cup W_3}$. Hence, by Lemma 15, $R^{\mathbf{a}} = A^{|W|}$; therefore, $R_{W \cup W_2 \cup W_3} = R_W \times R_{W_2 \cup W_3}$. The lemma is proved. $\square$

Case II. $\mathbb{A}$ is not simple.

In this case, the B_1-semirectagularity property holds, as the following lemma shows.

LEMMA 17. *The algebra $\mathbb{A}$ satisfies the B_1-semirectangular property.*

PROOF. Let $R \in \mathsf{Inv}$ be an (n-ary) relation, $W = \{i \mid 0, 1 \in R_i\}$, and $W_1, \ldots, W_k$ the classes of $\theta_{B_1}(R)$. It will be convenient for us to denote by $\overline{2}$ the tuple consisting of 2s; the length of this tuple will always be clear from the context. Suppose that $\mathbf{b} = (\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_k) \in R$, where $\mathbf{b}_0 \in R_{W'}$, $W' = \underline{n} - W$, $\mathbf{b}_i \in R_{W_i}$, $i \in \underline{k}$, and $\mathbf{b}_i = \overline{2}$ for $i \in \underline{k} - I$ and $\mathbf{b}_i \in R_{W_i} \cap \{0, 1\}^{|W_i|}$ for $i \in I$. We have to prove that, for any $\mathbf{a}_i \in R_{W_i} \cap \{0, 1\}^{|W_i|}$, $i \in I$, the tuple $(\mathbf{b}_0, \mathbf{d}_1, \ldots, \mathbf{d}_k)$ with

$$\mathbf{d}_i = \begin{cases} \mathbf{a}_i & \text{if } i \in I, \\ \overline{2} & \text{otherwise} \end{cases}$$

belongs to R.

We prove by induction that, for any $J = \{i_1, \ldots, i_l\} \subseteq I$, the tuple $\mathbf{c}_J = (\mathbf{b}_0, \mathbf{a}_{i_1}, \ldots, \mathbf{a}_{i_l})$ belongs to R_{W_J}, $W_J = W' \cup W_{i_1} \cup \cdots \cup W_{i_l}$. The base case of induction is obvious: $\mathbf{c}_\emptyset = \mathbf{b}$. The next case is $|J| = 1$. It will not be loss of generality if we assume $J = \{1\}$. There are $\mathbf{c}_0 \in R_{W'}$ and $\mathbf{c}_2 \in R_{W_2}$ such that $\mathbf{c} = (\mathbf{c}_0, \mathbf{a}_1, \mathbf{c}_2) \in R_{W' \cup W_2}$. The tuple $g(\mathbf{c}, \mathbf{b}) \in R_{W' \cup W_2}$ has the form $(\mathbf{b}_0, \mathbf{a}_1, \mathbf{c}_2')$ as required.

Let us suppose now that we have proved what is required for all 1-element sets J. Take J with $|J| = 2$; as usual it can be supposed to be $\{1, 2\}$. By what was proved, there are $\mathbf{d}_1 \in R_{W_1}$, $\mathbf{d}_2 \in R_{W_2}$ such that $(\mathbf{b}_0, \mathbf{d}_1, \mathbf{a}_2), (\mathbf{b}_0, \mathbf{a}_1, \mathbf{d}_2) \in R_{W' \cup W_1 \cup W_2}$. The tuples $\mathbf{d}_1, \mathbf{d}_2$ can be assumed to be from $\{0, 1\}^{|W_1|}$, $\{0, 1\}^{|W_2|}$ respectively. Indeed, if $\mathbf{d}_1 = \overline{2}$, then

$$g\left(\begin{pmatrix} \mathbf{b}_0 \\ \overline{2} \\ \mathbf{a}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}\right) = \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{a}_2 \end{pmatrix}.$$

Furthermore, since W_1, W_2 are different classes of $\theta_{B_1}(R)$, there are $\mathbf{c}_0 \in R_{W'}$ and $\mathbf{c}_1 \in R_{W_1}$ [or $\mathbf{c}_2 \in R_{W_2}$] such that $(\mathbf{c}_0, \mathbf{c}_1, \overline{2}) \in R_{W' \cup W_1 \cup W_2}$ (respectively, $(\mathbf{c}_0, \overline{2}, \mathbf{c}_2) \in R_{W' \cup W_1 \cup W_2}$) and $\mathbf{c}_1 \in \{0, 1\}^{|W_1|}$ (respectively, $\mathbf{c}_2 \in \{0, 1\}^{|W_2|}$). The tuple $\mathbf{c}_1$ can be chosen to be $\mathbf{a}_1$. Indeed,

$$\begin{pmatrix} \mathbf{c}_0 \\ \mathbf{a}_1 \\ \overline{2} \end{pmatrix} = g\left(\begin{pmatrix} \mathbf{b}_0 \\ \mathbf{a}_1 \\ \mathbf{d}_1 \end{pmatrix}, \begin{pmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \overline{2} \end{pmatrix}\right) \in R_{W' \cup W_1 \cup W_2}.$$

Then, we have

$$\begin{pmatrix} \mathbf{b}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \end{pmatrix} = g\left(\begin{pmatrix} \mathbf{c}_0 \\ \mathbf{a}_1 \\ \overline{2} \end{pmatrix}, \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{d}_1 \\ \mathbf{a}_2 \end{pmatrix}\right) \in R_{W' \cup W_1 \cup W_2},$$

as required.

Then, suppose that the inclusion $\mathbf{c}_J \in R_{W_J}$ is already proved for all J with $|J| < m$, and $K \subseteq \{1, \ldots, k\}$ is such that $|K| = m$. Again, without loss of generality, assume that $K = \{1, \ldots, m\}$.

By induction hypothesis, there are tuples of the form $\mathbf{b}^1 = (\mathbf{b}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{m-1}, \mathbf{d}^1_m)$, $\mathbf{b}^2 = (\mathbf{b}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{m-2}, \mathbf{d}^2_{m-1}, \mathbf{a}_m)$, $\mathbf{b}^3 = (\mathbf{b}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{m-3}, \mathbf{d}^3_{m-2}, \mathbf{a}_{m-1}, \mathbf{a}_m)$ in R. The tuples $\mathbf{d}^1_m, \mathbf{d}^2_{m-1}, \mathbf{d}^3_{m-2}$ can be chosen to be distinct from $\overline{2}$. Indeed, if, say, $\mathbf{d}^1_m = \overline{2}$, then, since $\mathbf{a}_i, \mathbf{b}_i \in \{0, 1\}^{|W_i|}$ for $i \in \{1, \ldots, m-1\}$, the tuple $g(\mathbf{b}^1, \mathbf{b})$ is of the form $(\mathbf{b}_0, \mathbf{a}_1, \dots, \mathbf{a}_{m-1}, \mathbf{b}_m)$, where $\mathbf{b}_m \in \{0,1\}^{|W_m|}$. Finally, we have

$$\begin{pmatrix} \mathbf{b}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{m-3} \\ \mathbf{a}_{m-2} \\ \mathbf{a}_{m-1} \\ \mathbf{a}_m \end{pmatrix} = f\left(\begin{pmatrix} \mathbf{b}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{m-3} \\ \mathbf{a}_{m-2} \\ \mathbf{a}_{m-1} \\ \mathbf{d}^1_m \end{pmatrix}, \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{m-3} \\ \mathbf{a}_{m-2} \\ \mathbf{d}^2_{m-1} \\ \mathbf{a}_m \end{pmatrix}, \begin{pmatrix} \mathbf{b}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{m-3} \\ \mathbf{d}^3_{m-2} \\ \mathbf{a}_{m-1} \\ \mathbf{a}_m \end{pmatrix}\right) \in R.$$

Finally, let $\mathbf{c} \in R$ be a tuple with $\mathbf{c}_{W_I} = \mathbf{c}_I$. As is easily seen, the tuple $(\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_k) = g(\mathbf{c}, \mathbf{b})$ satisfies the conditions: $\mathbf{d}_0 = \mathbf{b}_0$, $\mathbf{d}_i = g(\mathbf{a}_i, \mathbf{b}_i) = \mathbf{a}_i$ if $i \in I$ and $\mathbf{d}_i = g(\mathbf{c}_i, \overline{2}) = \overline{2}$ if $i \notin I$. The lemma is proved. $\square$

6.6. NONSIMPLE ALGEBRAS. In this section, we assume that $\mathbb{A}$ has a proper congruence. Without loss of generality, let us suppose that the equivalence relation θ whose classes are $\{0,1\}$ and $\{2\}$ is a congruence of $\mathbb{A}$. Clearly, $\mathbb{A}$ may have other proper congruences, but θ will be sufficient for us. Since $\mathbb{A}$ is idempotent, for any term operation $f(x_1, \dots, x_n)$ and any $\mathbf{a} \in \{0,1\}^n$, we have

$$\begin{pmatrix} f(0,\dots,0) \\ f(\mathbf{a}[1],\dots,\mathbf{a}[n]) \end{pmatrix} = \begin{pmatrix} 0 \\ f(\mathbf{a}[1],\dots,\mathbf{a}[n]) \end{pmatrix} \in \theta.$$

Therefore, $f(\mathbf{a}[1], \dots, \mathbf{a}[n]) \in \{0,1\}$, that is, $B = \{0,1\}$ is a subuniverse. Denote by $\mathbb{B}$ the subalgebra with the universe B. The condition (NO-G-SET) implies that there are term operations f, g of $\mathbb{A}$ such that f is a semilattice, majority or minority operation on B, and g^θ is a semilattice, majority or minority operation. We consider five cases depending on the types of $f|_B$ and of g^θ.

Case 1. $f|_B$ is a semilattice operation.

The operation f^θ is idempotent; hence, it is either a projection or a semilattice operation. In the latter case, we get Case 2. In the former case, we may assume that f^θ is the first projection that is $f^\theta(x,y) = x$. Consider first the case $f(1,0) = f(0,1) = 1$. We prove that $\mathbb{A}$ or a certain its reduct satisfies the $(0-1)$-replacement property.

Since f preserves θ, the operation table of $f'(x,y) = f(f(x,y),y)$ is

	0	1	2
0	0	1	0/1
1	1	1	1
2	2	2	2

therefore, f' is a $(0-1)$-operation. If $\{1,2\}$ is a subuniverse of $\mathbb{A}$, then, by Lemma 3, $\mathbb{A}$ satisfies the $(0-1)$-replacement property. Otherwise, if g^θ is a semilattice operation, then we get Case 2. So, suppose that g^θ is a minority or a majority operation, and consider the operation

$$g'(x,y,z) = f'(f'(f'(g(x,y,z),x),y),z).$$

If $g(x,y,z) \in \{1,2\}$, then $g'(x,y,z) = g(x,y,z)$. If $g(x,y,z) = 0$ and $x, y, z \in \{1,2\}$, then, as is easily seen, $g'(x,y,z) = 1$. Therefore, $\{1,2\}$ is a subuniverse of the algebra $\mathbb{A}' = (A; f', g')$, and g'^θ is a minority or a majority operation. Hence,

$\mathbb{A}'$ satisfies the condition (NO-G-SET) and the $(0-1)$-replacement property. The case when $f(0,1) = f(1,0) = 0$ is quite analogous.

Case 2. g^θ is a semilattice operation.

If $g^\theta(2^\theta, 0^\theta) = g^\theta(0^\theta, 2^\theta) = 2^\theta$, then 2 is a zero-element with respect to g. So suppose that $g^\theta(2^\theta, 0^\theta) = g^\theta(0^\theta, 2^\theta) = 0^\theta$.

Subcase 1. $g|_{\{0,1\}}$ is a projection.

Without loss of generality, we may assume that $g|_{\{0,1\}}$ is the first projection. As can be straightforwardly verified, the operation table of $h(x, y) = g(x, g(x, y))$ is one of the following:

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & 2 \end{array}, \quad \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 2 \end{array}, \quad \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 1 & 2 \end{array}.$$

In the first 2 cases, h is a $(2-0)$- or $(2-1)$-operation; in the third case, h is the operation (1). By Lemmas 3 and 9, $\mathbb{A}$ satisfies one of the conditions listed in Theorem 3.

Subcase 2. $g|_{\{0,1\}}$ is a semilattice operation.

Suppose that $g(0,1) = g(1,0) = 1$. Set $h(x, y) = g(x, g(x, y))$ and $h'(x, y) = h(h(x, y), y)$. Then, 1 is a zero-element with respect to h'.

In the following three cases, we assume that there are term operations f, g such that $f|_{\{0,1\}}, g^\theta$ are minority or majority operations.

Case 3. $\mathbb{A}$ is conservative.

In this case, we are in the conditions of Section 6.5.

Case 4. Exactly one of $\{0, 2\}$, $\{1, 2\}$ is a subuniverse.

Without loss of generality, suppose that $\{0, 2\}$ is a subuniverse, but $\{1, 2\}$ is not. Then $\mathbb{A}$ has one of the operations listed in Lemma 7. By Lemmas 9 and 8, some of them give rise to one of the 10 properties of $\mathbb{A}$. Since $\{0, 2\}$ is a subuniverse, $\mathbb{B}$ has no semilattice term operation and θ is a congruence, only the operation (14) remains. This operation is a $(1-0)$-operation and, since $\{0, 2\}$ is a subuniverse, by Lemma 3, $\mathbb{A}$ satisfies the $(1-0)$-replacement property.

Case 5. $\{0, 1\}$ is the only subuniverse of $\mathbb{A}$.

As above, by making use of Lemmas 7, 9, and 8, we have to consider only the cases when (16) or both (14) and (15) are term operations of $\mathbb{A}$.

Subcase 1. (14) and (15) are term operations of $\mathbb{A}$.

Let $r(x, y)$ denote the operation (14). For any term operation $h'(x_1, \ldots, x_n)$ of $\mathbb{A}$, the operation

$$\hat{h}(x_1, \ldots, x_n) = r(\cdots r(h(x_1, \ldots, x_n), x_1) \cdots x_n)$$

satisfies the conditions: $\hat{h}(x_1, \ldots, x_n) = 2$ if and only if $h(x_1, \ldots, x_n) = 2$, otherwise, if $2 \in \{x_1, \dots, x_n\}$, then $\hat{h}(x_1, \dots, x_n) = 0$ and $\hat{h}(x_1, \dots, x_n) = h(x_1, \dots, x_n)$ whenever $\{x_1, \dots, x_n\} \subseteq \{0, 1\}$. As is easily seen, the operation $\hat{f}|_{\{0,1\}}$ [correspondingly $\hat{g}^\theta$] is a minority or a majority operation if $f|_{\{0,1\}}$ [correspondingly g^θ] is a minority or a majority operation.

Let us consider the reduct of $\mathbb{A}$, $\mathbb{A}' = (A; \hat{f}, \hat{g})$. The algebra $\mathbb{A}'$ satisfies the condition (NO-G-SET) and $\{0, 2\}$ is a subuniverse of $\mathbb{A}'$. We are in the conditions of Case 4.

Subcase 2. (16) is a term operation of $\mathbb{A}$.

LEMMA 18. *Let g be a minority operation on $\mathbb{A}/_\theta$, and h the operation* (16). *Then either $\mathbb{A}$ satisfies the conditions of one of the previous cases, or it has a term operation g' that is minority on $\mathbb{A}/_\theta$, $g'|_{\{0,1\}} = g|_{\{0,1\}}$ and g' preserves* $\{0, 2\}, \{1, 2\}$.

PROOF. Since, for any $x, y \in \{0, 1\}$, the equalities $g(2, x, y) = g(x, 2, y) = g(x, y, 2) = 2$ hold, we just have to show that there is a term operation g' which is a minority operation on $\mathbb{A}/_\theta$ and $g'(2, 2, x) = g'(2, x, 2) = g'(x, 2, 2) = x$. Suppose first that $g(2, 2, 0) = g(2, 2, 1) = a \in \{0, 1\}$. Then, the operation table of $g(y, y, x)$ is $\begin{array}{c|ccc} & 0 & 1 & 2\\ \hline 0 & 0 & & a\\ 1 & & 1 & a\\ 2 & 2 & 2 & 2\end{array}$, and we get Case 1. Then, suppose that $g(2, 2, 1) = 0$, $g(2, 2, 0) = 1$. In this case, for the operation $g'(x, y, z) = g(x, y, h(z, x))$, we have $g'(x, y, 2) = g(x, y, 2)$, $g'|_{\{0,1\}} = g|_{\{0,1\}}$ and $g'(2, 2, x) = x$. Repeating the same procedure for all three variables we derive the required operation. $\square$

LEMMA 19. *Let g be a majority operation on $\mathbb{A}/_\theta$, and h the operation* (16). *Then, either $\mathbb{A}$ satisfies the conditions of one of the previous cases, or it has a term operation g', which is majority on $\mathbb{A}/_\theta$, $g'|_{\{0,1\}} = g|_{\{0,1\}}$ and g' preserves* $\{0, 2\}, \{1, 2\}$.

PROOF. The proof is quite similar to that of Lemma 18, but the required operation g' must satisfy the conditions $g'(2, x, x) = g'(x, 2, x) = g'(x, x, 2) = x$. Suppose first that $g(2, 0, 0) = g(2, 1, 1) = a \in \{0, 1\}$. Then, the operation table of $g(y, x, x)$ is $\begin{array}{c|ccc} & 0 & 1 & 2\\ \hline 0 & 0 & & a\\ 1 & & 1 & a\\ 2 & 2 & 2 & 2\end{array}$, and we get Case 1. Then, suppose that $g(2, 1, 1) = 0$, $g(2, 0, 0) = 1$. In this case, for the operation $g'(x, y, z) = h(g(x, y, z), x)$, we have $g'(x, y, z) = 2$ if and only if $g(x, y, z) = 2$, $g'|_{\{0,1\}} = g|_{\{0,1\}}$, $g'(x, 2, x) = g(x, 2, x)$, $g'(x, x, 2) = g(x, x, 2)$ and $g'(2, x, x) = x$. This implies, in particular, that g'^θ is a majority operation. Repeating the same procedure for all pairs of variables, we derive the required operation. $\square$

Since $g|_{\{0,1\}}$ is an idempotent operation and the subalgebra of $\mathbb{A}$ with the universe $\{0, 1\}$ has no semilattice term operation, we have $g|_{\{0,1\}}(x, x, y), g|_{\{0,1\}}(x, y, x), g|_{\{0,1\}}(y, x, x) \in \{x, y\}$. If g^θ and $g|_{\{0,1\}}$ are minority [majority] operations, then applying Lemma 18 [Lemma 19] to g, we get a minority [majority] operation on A. Noticing that a minority operation is a Mal'tsev operation, the algebra $\mathbb{A}$ has a Mal'tsev [majority] term operation. Otherwise, an operation h' such that $h'^\theta(x, y) = x$, $h'|_{\{0,1\}}(x, y) = y$ can be derived from g.

LEMMA 20. *Let f, g be term operations of $\mathbb{A}$ such that $f|_{\{0,1\}}$, g^{θ} are majority or minority operations. Then, there is a term operation g' of $\mathbb{A}$ such that $g'^{\theta} = g^{\theta}$, $g'|_{\{0,1\}} = f|_{\{0,1\}}$.*

PROOF. By the observation before the lemma, there is a term operation h' of $\mathbb{A}$ such that $h'^{\theta}(x, y) = x$, $h'|_{\{0,1\}}(x, y) = y$. Set $g'(x, y, z) = h'(g(x, y, z), f(x, y, z))$. Since, for any $x, y, z \in \{0, 1\}$, $f(x, y, z), g(x, y, z) \in \{0, 1\}$, we have $g'|_{\{0,1\}}(x, y, z) = f|_{\{0,1\}}(x, y, z)$. Furthermore, the equality $h^{\theta}(x, y) = x$ implies $g'^{\theta}(x, y, z) = g^{\theta}(x, y, z)$. The lemma is proved. $\square$

Finally, applying Lemmas 18 and 19, we get an operation g'' which preserves $\{0, 2\}$ and $\{1, 2\}$ and such that each of g''^{θ}, $g''|_{\{0,1\}}$ is a majority or minority operation. If either both of these operations are minority or both are majority, then $\mathbb{A}$ has a Mal'tsev or a majority term operation. Otherwise, the algebra $\mathbb{A}' = (A; g'')$ is conservative and a reduct of $\mathbb{A}$. We are going to show that θ is the only proper congruence of $\mathbb{A}'$. Indeed, as was observed above, an operation $h'(x, y)$ such that $h'^{\theta}(x, y) = x$, $h'|_{\{0,1\}}(x, y) = y$ is derivable from g''. Since g'' preserves all 2-element subsets of A, so does h'; therefore, its operation table is $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$. The pair $\binom{h'(0,1)}{h'(2,1)} = \binom{1}{2}$ witnesses that h' destroys the equivalence relation with classes $\{0, 2\}, \{1\}$; while the pair $\binom{h'(1,0)}{h'(2,0)}$ witnesses that h' destroys the equivalence relation with classes $\{1, 2\}, \{0\}$. Thus, $\mathbb{A}'$ satisfies the condition (NO-G-SET), and we get Case 3.

7. *A Practical Guide to Solving Constraint Satisfaction Problems on a* 3*-Element Set*

The main goal of this section is to prove Theorem 2 by providing a polynomial time algorithm solving the uniform problem. We also attempt to improve the algorithm solving the meta-problem. In Section 5, we showed how one can decide whether a finite constraint language on a 3-element set is tractable. However, that algorithm can only recognize whether the language is tractable or not, but does not give any clue on which of the algorithms listed in Section 4.3 is to be used to solve the problem. In the previous section, we carried out extensive study of the properties of 3-element algebras related to the complexity of corresponding constraint problems. Here we show how these results can be used in solving CSPs.

Let Γ be a constraint language on a 3-element set A. Notice first that all unary and binary polymorphisms can be easily computed. Therefore, using the algorithm from Section 5, we may assume that $\mathbb{A}_{\Gamma}$ is idempotent. Knowing the binary polymorphisms also allows us to find all subalgebras of $\mathbb{A}_{\Gamma}$, because if, for a 2-element subset $B \subseteq A$, there is a polymorphism that destroys B, then there is also a binary polymorphism with this property. Notice also that zero-operations and $(a - b)$-operations are binary, therefore, if Γ has a polymorphism of this form, it will be found. Although, in general, one needs the ternary polymorphisms to compute the congruences of $\mathbb{A}_{\Gamma}$, we shall see that in our case this can be avoided.

Below, we summarize information gained in Section 6 to decide whether there exists a polynomial time algorithm solving CSP(Γ); and if exists, then which of the 10 properties can be used.

The Partial Zero-Property. This property holds if there is a polymorphism which is a zero-operation, in particular, a semilattice operation (see Section 6.3, Lemmas 7, 9, 10, and 11; Subcases 2.1 and 2.2 in Section 6.5.1; or Case 2 in Section 6.6).

The Splitting Property. This property holds if $\mathbb{A}_\Gamma$ is conservative, simple and has a term operation f that is a majority operation on one of the 2-element subsets and a minority operation on the other two 2-element subsets. In this case, $\mathbb{A}' = (A; f)$ satisfies the condition (NO-G-SET). The congruences of $\mathbb{A}'$ are the equivalence relations invariant under f, therefore they can be easily found.

The $(a-b)$*-Replacement Property.* This property holds if there is a polymorphism which is an $(a-b)$-operation (see Lemmas 7, 10, 11, Case 3 in Section 6.5.1 and Cases 1, 2, 4 in Section 6.6) or one of the operations (4) and (5),

The B-Rectangularity Property. This property holds if one of the following conditions holds.

—The operation (3) is a polymorphism of Γ and there is a ternary polymorphism g such that $g|_{\{1,2\}}$ is an affine operation (Lemma 9). In this case, replacing g with $f(x, f(y, f(z, g(x, y, z))))$, where f denotes the operation (3), we may assume that $g(x, y, z) \subseteq \{1, 2\}$ unless $x = y = z = 0$.
—$\mathbb{A}_\Gamma$ is conservative and there exists a binary polymorphism whose restriction onto a certain 2-element subset is a semilattice operation.

The B-Semirectangular Property. This property holds if $\mathbb{A}_\Gamma$ is conservative, not simple and has a term operation f that is a majority operation on one of the 2-element subsets and a minority operation on the other two 2-element subsets. In this case $\mathbb{A}' = (A; f)$ satisfies the condition (NO-G-SET). The congruences of $\mathbb{A}'$ are the equivalence relations invariant under f; therefore, they can be easily found.

The B-Semisplitting Property. This property holds if $\mathbb{A}_\Gamma$ is conservative, two of the 2-element subalgebras have the majority term operation, but no semilattice operation, and the third subalgebra has the minority term operation, but neither a semilattice nor majority operation (Lemma 13).

The B-Extendibility Property. This property holds if one of the following conditions holds.

—The operation (2) is a polymorphism of Γ and there is a ternary polymorphism g such that $g|_{\{0,2\}}$ is a majority operation (Lemma 9). In this case, replacing g with $f(x, f(y, f(z, g(x, y, z))))$, where f denotes the operation (2), we may assume that $g(x, y, z) \subseteq \{0, 2\}$ unless $x = y = z = 1$.
—One of the operations (6) or (9) is a polymorphism of Γ (Lemma 8).

A Majority Term Operation. On a 3-element set there exist only 3^6 majority operations; hence, all the majority polymorphisms can be efficiently found.

A Binary Conservative Commutative Term Operation. Such an operation is binary, therefore, it can be found when computing the binary polymorphisms of Γ.

A Mal'tsev Term Operation. By Corollary 3.5 of Bulatov [2006b], if $\mathbb{A}_\Gamma$ has a Mal'tsev term operation, then it also has one of the following five Mal'tsev operations: $x - y + z$, where $+, -$ are the operations of an Abelian group, the operation p, the operation m defined by the following Cayley tables

$$
\begin{array}{r|ccc}
p(0,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 1 & 2 \\
1 & 1 & 0 & 0 \\
2 & 2 & 0 & 0
\end{array},\quad
\begin{array}{r|ccc}
p(1,x,y) & 0 & 1 & 2 \\ \hline
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 2 \\
2 & 1 & 2 & 1
\end{array},\quad
\begin{array}{r|ccc}
p(2,x,y) & 0 & 1 & 2 \\ \hline
0 & 2 & 2 & 0 \\
1 & 2 & 2 & 1 \\
2 & 0 & 1 & 2
\end{array}
$$

$$
\begin{array}{r|ccc}
m(0,x,y) & 0 & 1 & 2 \\ \hline
0 & 0 & 1 & 2 \\
1 & 1 & 0 & 2 \\
2 & 2 & 2 & 0
\end{array},\quad
\begin{array}{r|ccc}
m(1,x,y) & 0 & 1 & 2 \\ \hline
0 & 1 & 0 & 2 \\
1 & 0 & 1 & 2 \\
2 & 2 & 2 & 1
\end{array},\quad
\begin{array}{r|ccc}
m(2,x,y) & 0 & 1 & 2 \\ \hline
0 & 2 & 2 & 0 \\
1 & 2 & 2 & 1 \\
2 & 0 & 1 & 2
\end{array}
$$

and the two operations obtained from m using two cyclic permutations of the elements 0, 1, and 2. Clearly, the presence of one of them can be easily detected.

REFERENCES

ALLEN, J. 1994. *Natural Language Understanding*. Benjamin Cummings.

BULATOV, A. 2002. Mal'tsev constraints are tractable. Tech. Rep. PRG-RR-02-05, Computing Laboratory, University of Oxford, Oxford, UK.

BULATOV, A. 2006a. Combinatorial problems raised from 2-semilattices. *J. Alg.*, submitted for publication.

BULATOV, A. 2006b. Three-element Mal'tsev algebras. *Acta Sci. Math (Szeged)*. To appear.

BULATOV, A., AND JEAVONS, P. 2000. Tractable constraints closed under a binary operation. Tech. Rep. PRG-TR-12-00, Computing Laboratory, University of Oxford, Oxford, UK.

BULATOV, A., AND JEAVONS, P. 2001a. Algebraic approach to multisorted constraints. Tech. Rep. PRG-RR-01-18, Computing Laboratory, University of Oxford, Oxford, UK.

BULATOV, A., AND JEAVONS, P. 2001b. Algebraic structures in combinatorial problems. Tech. Rep. MATH-AL-4-2001, Technische universität Dresden, Dresden, Germany.

BULATOV, A., AND JEAVONS, P. 2003. An algebraic approach to multisorted constraits. In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP'03)* (Kinsale, Ireland). 197–202.

BULATOV, A., JEAVONS, P., AND KROKHIN, A. 2001. The complexity of maximal constraint languages. In *Proceedings of the 33rd Annual ACM Simposium on Theory of Computing* (Hersonissos, Crete, Greece). ACM, New York, 667–674.

BULATOV, A., JEAVONS, P., AND KROKHIN, A. 2005. Classifying the complexity of constraints using finite algebras. SIAM *J. Comput. 34*, 720–742.

BURRIS, S., AND SANKAPPANAVAR, H. 1981. *A Course in Universal Algebra*. Graduate Texts in Mathematics, vol. 78. Springer-Verlag, New York-Berlin.

COOPER, M. 1989. An optimal k-consistency algorithm. *Artif. Intel. 41*, 89–95.

CREIGNOU, N., KHANNA, S., AND SUDAN, M. 2001. *Complexity Classifications of Boolean Constraint Satisfaction Problems*. SIAM Monographs on Discrete Mathematics and Applications, vol. 7. SIAM, Philadelphia, PA.

DALMAU, V. 2000. Computational complexity of problems over generalized formulas. Ph.D. dissertation, Department LSI of the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain.

DECHTER, R. 1992. From local to global consistency. *Artif. Intel. 55*, 1, 87–107.

DECHTER, R. 2003. *Constraint processing*. Morgan-Kaufmann, San Francisco, CA.

DECHTER, R., AND DECHTER, A. 1996. Structure-driven algorithms for truth maintenance. *Artif. Intel. 82*, 1–2, 1–20.

DECHTER, R., AND MEIRI, I. 1994. Experimental evaluation of preprocessing algorithms for constraint satisfaction problems. *Artif. Intel. 68*, 211–241.

DECHTER, R., AND PEARL, J. 1988. Network-based heuristics for constraint satisfaction problems. *Artif. Intel. 34*, 1, 1–38.

DECHTER, R., AND VAN BEEK, P. 1997. Local and global relational consistency. *Theoret. Comput. Sci. 173*, 1, 283–308.

DENECKE, K., AND WISMATH, S. 2002. *Universal Algebra and Applications in Theoretical Computer Science*. Chapman and Hall/CRC Press.

FEDER, T., AND VARDI, M. 1998. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput. 28*, 57–104.

FREUDER, E. 1982. A sufficient condition for backtrack-free search. *J. ACM 29*, 1, 24–32.

GOTTLOB, G., LEONE, L., AND SCARCELLO, F. 2000. A comparison of structural CSP decomposition methods. *Artif. Intel. 124*, 2, 243–282.

JEAVONS, P. 1998a. Constructing constraints. In *Proceedings 4th International Conference on Constraint Programming—CP'98* (Pisa, Oct.). Lecture Notes in Computer Science, vol. 1520. Springer-Verlag, New York, 2–16.

JEAVONS, P. 1998b. On the algebraic structure of combinatorial problems. *Theoret. Comput. Sci. 200*, 185–204.

JEAVONS, P., COHEN, D., AND COOPER, M. 1998a. Constraints, consistency and closure. *Artif. Intel. 101*, 1–2, 251–265.

JEAVONS, P., COHEN, D., AND GYSSENS, M. 1997. Closure properties of constraints. *J. ACM 44*, 527–548.

JEAVONS, P., COHEN, D., AND PEARSON, J. 1998b. Constraints and universal algebra. *Ann. Math. Artif. Intel. 24*, 51–67.

KOLAITIS, P. 2003. Constraint satisfaction, databases, and logic. In *Proceedings of the 17th International Joint Conference on Artificial Intellignece (IJCAI'03)*.

KOLAITIS, P., AND VARDI, M. 2000a. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci. 61*, 302–332.

KOLAITIS, P., AND VARDI, M. 2000b. A game-theoretic approach to constraint satisfaction. In *Proceedings of the 17th National (US) Conference on Artificial Intelligence (AAAI'00)*. 175–181.

KUMAR, V. 1992. Algorithms for constraint satisfaction problems: A survey. *AI Magazine 13*, 1, 32–44.

LADNER, R. 1975. On the structure of polynomial time reducibility. *J. ACM 22*, 155–171.

MCKENZIE, R., MCNULTY, G., AND TAYLOR, W. 1987. *Algebras, Lattices and Varieties*. Vol. I. Wadsworth and Brooks, California.

MONTANARI, U. 1974. Networks of constraints: Fundamental properties and applications to picture processing. *Inf. Sci. 7*, 95–132.

NADEL, B. 1995. Constraint satisfaction in Prolog: Complexity and theory-based heuristics. *Inf. Sci. 83*, 3–4, 113–131.

NADEL, B., AND LIN, J. 1991. Automobile transmission design as a constraint satisfaction problem: Modeling the kinematik level. *Artif. Intel. Eng. Des. Anal. Manuf. (AI EDAM) 5*, 3, 137–171.

PÖSCHEL, R., AND KALUŽNIN, L. 1979. *Funktionen- und Relationenalgebren*. DVW, Berlin, Germany.

POST, E. 1941. *The Two-Valued Iterative Systems of Mathematical Logic*. Annals Mathematical Studies, vol. 5. Princeton University Press.

SCHAEFER, T. 1978. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing (STOC'78)*. ACM, New York, 216–226.

SCHWALB, E., AND VILA, L. 1998. Temporal constraints: A survey. *Constraints 3*, 2–3, 129–149.

SZENDREI, A. 1990. Simple surjective algebras having no proper subalgebras. *J. Austral. Math. Soc. (Ser. A) 48*, 434–454.

TSANG, E. 1993. *Foundations of Constraint Satisfaction*. Academic Press, London, England.

VAN HENTENRYCK, P., DEVILLE, Y., AND TENG, C.-M. 1992. A generic arc-consistency algorithm and its specializations. *Artif. Intel. 57*, 291–321.

VARDI, M. 2000. Constraint satisfaction and database theory: A tutorial. In *Proceedings of 19th ACM Symposium on Priciples of Database Systems (PODS'00)*. ACM, New York.

RECEIVED NOVEMBER 2003; REVISED OCTOBER 2005; ACCEPTED OCTOBER 2005