Tractable Constraint Satisfaction Problems on a 3-element set

Andrei A. Bulatov
Computing Laboratory, University of Oxford

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 which 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 paper we generalise this result to a classification of the complexity of CSPs on a 3-element domain. The main result states that every subclass of the CSP is either tractable or NP-complete, and the criterion separating them is that conjectured in [7; 9]. We also 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 [20] and [7; 9].

Categories and Subject Descriptors: []:

General Terms: Algorithms, Theory

Additional Key Words and Phrases:

1. INTRODUCTION

In the Constraint Satisfaction Problem (CSP) [29] we aim to find an assignment to a set of variables subject specified constraints. This problem can also be reformulated as (1) the problem of deciding whether there is a model of a conjunctive formula, as (2) the problem of deciding the existence of a homomorphism between two finite relational structures $\mathbf{A}$ and $\mathbf{B}$ (this is the form we use throughout the paper), and also as (3) the problem of deciding if the evaluation $Q(D)$ of a conjuntive query Q on a database D is non-empty; see [16; 25].

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 [17], graph colorability, scheduling problems, linear systems and many others. One 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

Oxford, UK, *e-mail: Andrei.Bulatov@comlab.ox.ac.uk*

be applied in many different areas such as database theory [25; 37], temporal and spatial reasoning [35], machine vision [29], belief maintenance [13], technical design [31], natural language comprehension [1], programming language analysis [30], etc.

The general CSP is NP-complete; however, certain restrictions on the allowed form of the constraints involved may ensure tractability. Therefore, one of the main approaches in study of the CSP is identifying tractable subclasses of the general CSP obtained in this way [16; 18; 11; 20; 34]. Developments in this direction provide an efficient algorithm solving a particular problem, if the problem falls in one of the known tractable subclasses, or assist in speeding up of general superpolynomial algorithms [14; 15; 27]. To formalise the idea of restricting the allowed constraints, we make use of the notion of a *constraint language* [19], which is simply a set of possible relations that can be used to specify constraints in a problem. The ultimate goal of this research direction is to find the precise border between tractable and intractable constraint languages.

This goal was achieved by Schaefer [34] in the important case of Boolean constraints; 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. 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. For more dichotomy results for Boolean CSPs, and a short survey of dichotomy results for other cases the reader is referred to [11].

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, in [16], used database technique and group theory to identify some large tractable families of constraints; Jeavons and co-authors have characterised many tractable and NP-complete constraint languages using invariance properties of constraints [20; 21; 22]; in [9], a possible form of a dichotomy result for the CSP on finite domains was conjectured; in [8], a dichotomy result was proved for certain type of constraint languages on a 3-element domain. In this paper we generalise the results of [34] and [8], and prove the dichotomy conjecture from [9] for the constraint satisfaction problem on a 3-element domain. In particular, we completely characterise tractable constraint languages in this case, and prove that the rest are NP-complete. The main result will be precisely stated at the end of Section 2.

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 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 is showed in [22; 23; 20], when studying the complexity of constraint languages we may restrict ourselves with 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 [33]. However, already the class of relational clones on a 3-element set contains continuum many elements, and is believed to be incomprehensible.

Another problem tackled here is referred to, in [11], as the *meta-problem*: given constraint language determine if it gives rise to a tractable problem class. Making use of the dichotomy theorem obtained we exhibit an effective algorithm solving the meta-problem for the CSP on a 3-element domain.

The technique used in this paper relies upon the idea that was developed in [9; 7; 20] (and also mentioned in [16] 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. 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 [3; 5; 21]. 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.

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 Montanary [29].

DEFINITION 1. *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, $\{C_1, \ldots, C_q\}$,

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 *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.*

Is was observed in [16], that the constraint satisfaction problem can be equivalently reformulated in the form of the Homomorphism Problem. We need more definitions.

A *vocabulary* is a finite set of relational symbols $R_1, \ldots, R_n$ each of which has a fixed arity. A *relational structure* over the vocabulary $R_1, \ldots, R_n$ is a tuple $\mathcal{H} = (H; R_1^{\mathcal{H}}, \ldots, R_n^{\mathcal{H}})$ such that A is a non-empty 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 $R_1, \ldots, R_n$. A *homomorphism* from $\mathcal{G}$ to $\mathcal{H}$ is a mapping $\varphi: G \to H$ from the universe of $\mathcal{G}$ to the universe H of $\mathcal{H}$ such that, for every relation $R^{\mathcal{G}}$ of $\mathcal{G}$ and every tuple $(\mathbf{a}[1], \ldots, \mathbf{a}[m]) \in R^{\mathcal{G}}$, we have that $(\varphi(\mathbf{a}[1]), \ldots, \varphi(\mathbf{a}[m])) \in R^{\mathcal{H}}$.

DEFINITION 2. *Let* $\mathfrak{H}$ *be a class of relational structures. over a vocabulary* $R_1, \ldots, R_n$*. In the* homomorphism problem associated with $\mathfrak{H}$ ($\mathsf{HOM}(\mathfrak{H})$) *the question is, given a structure* $\mathcal{H} \in \mathfrak{H}$ *over a vocabulary* $R_1, \ldots, R_n$*, and a structure* $\mathcal{G}$ *over the same vocabulary, whether there exists a homomorphism from* $\mathcal{G}$ *to* $\mathcal{H}$*.*

The intuition behind the mentioned equivalence between the constraint satisfaction and homomorphism problems is that the source structure in the homomorphism problem 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, $\Gamma = \{R_1, \ldots, R_n\}$, on a set H is finite, then $\mathrm{CSP}(\Gamma)$ can be identified with $\mathsf{HOM}(\{\mathcal{H}\})$ (which often is denoted simply $\mathsf{HOM}(\mathcal{H})$) where $\mathcal{H} = (H; R_1, \ldots, R_n)$. If Γ is infinite then $\mathrm{CSP}(\Gamma)$ is equivalent to $\mathsf{HOM}(\mathfrak{H})$ where $\mathfrak{H}$ is the class of structures $\mathcal{H} = (A; R_1^A, \ldots, R_n^A)$, $R_1^A, \ldots, R_n^A \in \Gamma$. Problems of the form $\mathsf{HOM}(\mathcal{H})$ are widely studied, see, e.g. [16; 23; 20; 26; 25; 24]. However, since the technique we use in this paper makes it more natural to deal with infinite constrain languages, we state our main results within the framework of the constraint satisfaction problem. However, many of the results can be stated in a stronger form for homorphism problem. 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, i.e., problems for which there exists a polynomial-time solution algorithm, and those which do not.

DEFINITION 3. *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 $\mathrm{CSP}(\Gamma')$ is tractable for each finite subset $\Gamma' \subseteq \Gamma$. It is said to be* NP-complete, *if $\mathrm{CSP}(\Gamma')$ is NP-complete for some finite subset $\Gamma' \subseteq \Gamma$.*

In [34], Schaefer has classified two-element structures or, equivalently, Boolean constraint languages with respect to the complexity. This result is known as Schaefer's Dichotomy theorem.

THEOREM 1. *A Boolean constraint language, Γ, is tractable if and only if one of the following 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 linear system over* $\mathrm{GF}(2)$.

Otherwise Γ is NP-complete.

More examples of both tractable and NP-complete relational structures will appear later in this paper and can also be found in [9; 12; 16; 22].

The classification problem for larger structures is still open and seems to be very interesting and hard [16].

PROBLEM 1 CLASSIFICATION PROBLEM. *Characterise 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 recognise tractable cases.

PROBLEM 2 ALGORITHMIC PROBLEM. *Find efficient (polynomial time) algorithms for* $\mathsf{HOM}(\mathcal{H})$ [CSP(Γ)], $\mathcal{H}$, [Γ] *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 Theorem 1 that every Boolean constraint language [correspondingly, 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. T.Feder and M.Vardi [16] conjectured that this is the case for all problems $\mathsf{HOM}(\mathcal{H})$,CSP($\Gamma$). A criterion distinguishing tractable problems has been suggested in [9].

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

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 our algebraic definitions we mainly follow [10; 28].

Every relational structure $\mathcal{H}$ has a collection of associated operations on the same universe. The unary associated operations 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.

DEFINITION 4. *An n-ary* polymorphism *of a relational structure $\mathcal{H}$ is a homomorphism from the n-th Cartesian power of $\mathcal{H}$, $\mathcal{H} \times \ldots \times \mathcal{H}$, 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.

DEFINITION 5. *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 $(a_{11}, \ldots, a_{m1}), \ldots, (a_{1n}, \ldots, a_{mn}) \in R$, the tuple $(f(a_{11}, \ldots, a_{1n}), \ldots, f(a_{m1}, \ldots, a_{mn}))$ 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* $\mathsf{Inv}\ F$.

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 $x_1 \oplus \ldots \oplus x_n$ or $x_1 \oplus \ldots \oplus x_n \oplus 1$, 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 e.g. [32]. 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.

THEOREM 2 [20]. *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.

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 2 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.

In [21; 22], Jeavons and co-authors have identified certain types of operations which give rise to tractable problem classes.

DEFINITION 6. *Let A be a finite set. An operation f on A is called*

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

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

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

—*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 [20; 21] the term *ACI operation* is used.

—*a* binary conservative commutative *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 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.*

PROPOSITION 1 [21; 22; 5; 3]. *If a relational structure [a constraint language] has a polymorphism which is constant, semilattice, binary conservative commutative, Mal'tsev, or majority, then it is tractable.*

PROPOSITION 2 [21; 22]. *If every polymorphism of a relational structure [a constraint language] is an essentially unary relation then it is NP-complete.*

The 2-element relational structures from Schaefer's six classes 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).

Similar to relational structures, sets endowed with some family of operations have been intensively studied in algebra.

DEFINITION 7. *An* algebra *is an ordered 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.

DEFINITION 8. *Given a relational structure,* $\mathcal{H}$, *with the universe* A, *the algebra* $(A, \mathsf{Pol}\ \mathcal{H})$ *is called the* algebra associated with $\mathcal{H}$, *and is denoted* $\mathbb{A}_{\mathcal{H}}$.

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

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 $\mathbb{A}$, the language $\mathsf{Inv}\ F$. Moreover, $\mathbb{A}$ has the associated problem class $\mathrm{CSP}(\mathbb{A}) = \mathrm{CSP}(\mathsf{Inv}\ F)$.

DEFINITION 9. *An algebra* $\mathbb{A} = (A; F)$ *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.

DEFINITION 10. *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, the 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 mean of superposition.

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 $x_1 \oplus x_2 \oplus \ldots \oplus x_n$ is a term operation of $\mathbb{A}$ because

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

□

PROPOSITION 3 [21; 22]. *If a finite algebra has a term operations which is constant, semilattice, binary conservative commutative, Mal'tsev, or majority, then it is tractable.*

An algebra is said to be a *G-set* if every its term operation is essentially unary, and the corresponding unary operation is a permutation.

PROPOSITION 4 [21; 22]. *A finite G-set is NP-complete.*

By combining those two results, and the classical result of E.Post [33], the algebraic version of Schaefer's theorem can be derived [9].

THEOREM 3 SCHAEFER. *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

Certain transformations of relational structures and constraint languages 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 to no 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. The 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 5 [9]. *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 $g(R) = \{(g(a_1), \ldots, g(a_n)) \mid (a_1, \ldots, a_n) \in R\}$, 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 [9]. *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 $f(x, \ldots, x) = x$ 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 Theorem 2 and Corollary 3 the study of the complexity of constraint languages is completely reduced to the study of properties of idempotent algebras.

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 satisfying the conditions above: g that maps 0 to 0 and $1, 2$ to 1, and h that 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 $x_1 \oplus \ldots \oplus x_n$, 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.

DEFINITION 11. *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 $b_1, \ldots, b_n \in B$, we have $f(b_1, \ldots, b_n) \in B$. Then the algebra $\mathbb{B} = (B, F\big|_B)$, where $F\big|_B$ 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 $\mathbb{B} \neq \mathbb{A}$ ($B \neq A$).*

DEFINITION 12. *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* factor-set*, and the algebra $\mathbb{A}/_\theta = (A/_\theta; F^\theta)$, $F^\theta = \{f^\theta \mid f \in F\}$ where $f^\theta(a_1^\theta, \ldots, a_n^\theta) = (f(a_1, \ldots, a_n))^\theta$, is said to be the* factor-algebra*.*

PROPOSITION 6 [9]. *Let Γ be a tractable constraint language on A, $B \subseteq A$ a subuniverse of $\mathbb{A}_\Gamma$, and θ a congruence of $\mathbb{A}_\Gamma$. Then*
(1) the subalgebra $\mathbb{B} = (B; (\mathsf{Pol}\,\Gamma)\big|_B)$ 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 factor-algebra of a tractable algebra is tractable. Furthermore, every *factor* of a tractable algebra, that is, a factor-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, make the following conjecture.

CONJECTURE 1 [9]. *A constraint language Γ on a finite set A is tractable if and only if $\mathbb{A}_{\Gamma^{\rm id}}$ satisfies* (NO-G-SET). *Otherwise it is NP-complete.*

By Theorem 3, the conjecture holds for constraint languages on a 2-element set. The main result of this paper is that the conjecture holds for constraint languages on a 3-element set.

THEOREM 4. *A constraint language Γ on a 3-element set is tractable if and only if the algebra $\mathbb{A}_{\Gamma^{\rm id}}$ satisfies* (NO-G-SET). *Otherwise Γ is NP-complete.*

3. TRACTABLE CONSTRAINT LANGUAGES ON A 3-ELEMENT SET

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 4,6. To show that the condition is sufficient, we have to exhibit, 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 exhibit a polynomial time algorithm that solves the constraint satisfaction problem rising from a set of relations satisfying this property.

In this section we define the 10 properties and state the main algebraic theorem. The algorithms will be constructed in Section 4, while the algebraic part based on an elaborate case analysis is done in Appendix.

Throughout this section $\mathbb{A}$ satisfies (NO-G-SET). By $\underline{n}$, for a natural number n, we will denote the set $\{1, \ldots, n\}$. Let R be an n-ary relation, and $I = \{i_1, \ldots, i_k\} \subseteq \underline{n}$; then R_I denotes the k-ary relation $\{\mathbf{a}_I \mid \mathbf{a} \in R\}$ where $\mathbf{a}_I = (\mathbf{a}[i_1], \ldots, \mathbf{a}[i_k])$. We will often consider relations whose coordinate positions are indexed by not necessarily natural numbers, but elements of some arbitrary set. 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 generated by the set

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

and $W_1, \ldots, W_k$ the classes of this equivalence relation. Recall that the *graph* of a mapping $f\colon A \to B$ 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 13. *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* $R \in \mathsf{Inv}\ F$*,*

$R_W \cap B^{|W|} = (R_{W_1} \cap B^{|W_1|}) \times \ldots \times (R_{W_k} \cap B^{|W_k|})$*,*

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* $R \in \mathsf{Inv}\ F$*,*

—*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* $\mathbf{a}[k] = a$*;*

—*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* $\mathbf{a}[k] = a, \mathbf{a}[l] = b$*;*

—*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*

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

THEOREM 5. *If an idempotent 3-element algebra* $\mathbb{A} = (A; F)$ *satisfies* (NO-G-SET)*, then there is a set* F' *of its term operations such that the algebra* $\mathbb{A}' = (A; F')$ *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.*

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

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 resembles algorithms from linear algebra.

4.1 Partial solutions and bounded width

Many heuristic algorithms for constraint satisfaction involve "constraint propagation", which can be intuitevely 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.

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.

DEFINITION 14. *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 two properties of constraint satisfaction problems.

DEFINITION 15. *Let* $\mathcal{P} = (V; A; \mathcal{C})$ *be a constraint satisfaction problem.*

(1) It 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) The problem is said to be strong k-consistent *if it is* l*-consistent for all* $l \le k$*.*

(3) The problem $\mathcal{P}$ *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* $\mathcal{S}_W$*.*

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 Fig. 1. The output of this algorithm is said to be the k-minimal problem associated with $\mathcal{P}$.

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** $\mathcal{P}', \mathcal{P}'' := (V; A; \mathcal{C}')$ where
$\mathcal{C}' = \mathcal{C} \cup \{\langle W, A^k \rangle \mid W \subseteq V,\ |W| = k\}$;
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 $R' = \{\mathbf{a} \in R \mid \mathrm{pr}_{s \cap W} \mathbf{a} \in \mathrm{pr}_{s \cap W} \mathcal{S}'_W\}$.
endfor
Step 7 **until** $\mathcal{P}'' = \mathcal{P}'$.
Step 8 **output** $\mathcal{P}'$.

Fig. 1. Algorithm k-MINIMALITY

DEFINITION 16. *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 $\mathbf{C}$ *is of width* k *for a certain* k *then* $\mathbf{C}$ *is said to be of* bounded width.

DEFINITION 17. *A class* $\mathbf{C}$ *of constraint satisfaction problems is said to be of* relational width k *if any problem instance* $\mathcal{P}$ *from* $\mathbf{C}$ *has a solution if and only if the* k*-minimal problem associated with* $\mathcal{P}$ *contains no constraint with empty constraint relation.*

If $\mathbf{C}$ *is of relational width* k *for a certain* k *then* $\mathbf{C}$ *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 $\mathrm{CSP}(\Gamma)$, Γ finite, or, equivalently, problems of the form $\mathsf{HOM}(\mathcal{H})$. Indeed, let k be the maximal arity of relations from Γ, $\mathcal{P} \in \mathrm{CSP}(\Gamma)$ 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 prolem associated with $\mathcal{P}$.

Sometimes a problem may have bounded width just because of certain polymorphisms of the constraint language.

PROPOSITION 7 [21; 22]. *If a constraint language* Γ *has a semilattice polymorphism then* $\mathrm{CSP}(\Gamma)$ *is of relational width 1. If it has a majority polymorphism the* $\mathrm{CSP}(\Gamma)$ *is of relational width 3.*

4.2 Multi-sorted constraints satisfaction problem

In [6], it was shown that the generalised version of the constraint satisfaction problem, *multi-sorted constraint satisfaction problem*, in which every variable is allowed to have its own domain, can also be studied through algebraic approach. In this paper we need multi-sorted constraint satisfaction problem, and some results from [6] as an auxiliary tool. So, we cite the required definitions and results very briefly.

Definition 18. *For any collection of sets $\mathcal{A} = \{A_i \mid i \in I\}$, and any list of indices $(i_1, i_2, \ldots, i_m) \in I^m$, a subset ϱ of $A_{i_1} \times A_{i_2} \times \cdots \times A_{i_m}$, together with the list $(i_1, i_2, \ldots, i_m)$, will be called a* multi-sorted relation *over $\mathcal{A}$ with arity m and* signature $(i_1, i_2, \ldots, i_m)$. *For any such relation R, the signature of R will be denoted $\sigma(R)$.*

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

Definition 19. *Let Γ be a multi-sorted constraint language over a collection of sets $\mathcal{A} = \{A_i \mid i \in I\}$. The* multi-sorted constraint satisfaction problem over Γ, *denoted* $\mathrm{m}-\mathrm{CSP}(\Gamma)$, *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*

. *$s = (v_1, \ldots, v_{m_C})$ 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 $(\delta(v_1), \ldots, \delta(v_{m_C}))$, called the* constraint relation.

The question whether there exists a solution, *i.e., 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 $s = (v_1, \ldots, v_m)$, the tuple $(\varphi(v_1), \ldots, \varphi(v_m))$ belongs to R?*

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

It is possible to introduce the algebraic structure of the multi-sorted 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.

Definition 20. *Let $\mathcal{A}$ be a collection of sets. An n-ary* multi-sorted operation *t on $\mathcal{A}$ is defined by a collection of* interpretations *$\{t^A \mid A \in \mathcal{A}\}$, where each t^A is an n-ary operation on the corresponding set A. The multi-sorted operation t on $\mathcal{A}$ is said to be a* polymorphism *of a multi-sorted relation R over $\mathcal{A}$ with signature $(\delta(1), \ldots, \delta(m))$ if, for any $(a_{11}, \ldots, a_{m1}), \ldots, (a_{1n}, \ldots, a_{mn}) \in R$, we have*

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

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

The next theorem establishes the remarkable fact that many of the known one-sorted tractable classes listed in Proposition 1 can be combined in almost arbitrary ways to obtain new multi-sorted tractable classes.

Note that a multi-sorted operation, t, is said to be *idempotent* if every its interpretations t^A is an idempotent operation.

THEOREM 6 [4; 6]. *Let Γ be a multi-sorted constraint language over a collection of finite sets $\mathcal{A} = \{A_1, \ldots, A_n\}$.*

If, for each $A_i \in \mathcal{A}$, $\mathsf{m}-\mathsf{Pol}\,\Gamma$ contains a multi-sorted 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 $\mathrm{m}-\mathrm{CSP}(\Gamma)$ *is tractable.*

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

DEFINITION 21. *The 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 multi-sorted 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 is called the set of basic operations of the collection $\mathcal{A}$.*

DEFINITION 22. *For any collection of similar algebras $\mathcal{A} = \{(A_i; F^{A_i}) \mid i \in I\}$ with basic operations F, we define $\mathsf{m}-\mathsf{Inv}\,\mathcal{A}$ to be the set Γ of all multi-sorted relations over the sets $\{A_i \mid i \in I\}$ such that $\mathsf{m}-\mathsf{Pol}\,\Gamma = F$.*

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

PROPOSITION 8 [6]. *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 term operation, or a 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 9 [6]. *Let $\mathbb{A}_1, \ldots, \mathbb{A}_l$ be similar finite algebras and every $\mathbb{A}_i$ have either a constant term operation, or a semilattice or majority term operation. Then* $\mathrm{m}-\mathrm{CSP}(\{\mathbb{A}_1, \ldots, \mathbb{A}_l, \mathbb{B}_1, \ldots, \mathbb{B}_n\})$ *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 multi-sorted constraint satisfaction problems appear in this paper. We say that a problem instance $\mathcal{P} = (V; A; \mathcal{C}) \in \mathrm{CSP}(\mathbb{A})$ 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 multi-sorted problem over the family of all proper subalgebras of $\mathbb{A}$. If $\mathbb{A}$ satisfies (No-G-Set), then by the results of [33], every 2-element subalgebra of $\mathbb{A}$ has a constant, semilattice, majority or affine term operation. If $\mathbb{A}$ is idempotent then the collection of its subalgebras satisfies the conditions of Theorem 8. Therefore, $\mathcal{P}$ can be solved in polynomial time.

Corollary 4. *If a 3-element idempotent algebra satisfies* (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 5 allows us, first, to reduce an arbitrary problem instance to a 2-valid problem instance, and, second, to solve the obtained instance as a multi-sorted problem instance by making use of the algorithms from [6].

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.

4.3.1 *Relations invariant with respect to a special operation.* In (8), the tractability of $\mathbb{A}$ follows from Proposition 3. As is shown in [21], CSP($\mathbb{A}$) is of width 3 in this case. In (9), CSP($\mathbb{A}$) is of width 3, as is proved in [5]. The result of [3] states that any finite algebra with a Mal'tsev term operation is tractable, and the solution algorithm 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 13(1) and, therefore, $\mathbf{b} \in R$.

Thus, to solve $\mathcal{P}$ we let W to be the set of variables $v \in V$ such that $\mathcal{S}_v \in Z$ and 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 subalgebras 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 $\begin{pmatrix} \mathbf{a}[v] \\ \mathbf{a}[w] \end{pmatrix} \in \mathcal{S}_{\{v,w\}}$, for any $v, w \in W$.*

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, $\mathcal{S}_{\{v,w\}} \cap B^2 \neq \varnothing$. Let $W' \subseteq W$ be a maximal set such that there is $\mathbf{a} \in B^{|W'|}$ with $\begin{pmatrix} \mathbf{a}[v] \\ \mathbf{a}[w] \end{pmatrix} \in \mathcal{S}_{\{v,w\}}$, 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 $\begin{pmatrix} \mathbf{a}_{W'}[u] \\ a \end{pmatrix} \notin \mathcal{S}_{\{u,w\}}$, $\begin{pmatrix} \mathbf{a}_{W'}[v] \\ b \end{pmatrix} \notin \mathcal{S}_{\{v,w\}}$. 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 $\begin{pmatrix} \mathbf{a}_{W'}[u] \\ \mathbf{a}_{W'}[v] \end{pmatrix} \in R_{\{u,v\}}$, and by the B-extendibility property, for some $\mathbf{b} \in R$, $\mathbf{b}[u] = \mathbf{a}_{W'}[u], \mathbf{b}[v] = \mathbf{a}_{W'}[v]$, and $\mathbf{b}[w] \in B$, that contradicts the assumptions made. □

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 property.* 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 $\mathcal{P} = (V; A; \mathcal{C}) \in \mathrm{CSP}(\mathbb{A})$. 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 $\bigcup_{\langle s,R\rangle \in \mathcal{C}} \theta_B(R)$. 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 Fig 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 $\mathcal{P}^\eta$

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 8 **Output** $\mathcal{P}'$.

Fig. 2. Algorithm RECTANGULARITY

the *factor-problem*, that is the multi-sorted 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 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 [5]). 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

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

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 $i \in \{1, \ldots, k\}$ 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 factor-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 $\varphi(v) = c^\eta$, $\mathbf{b}[v] \in B$ when $\varphi(v) = B$. The semirectangularity of $\mathbb{A}$ implies that $(\psi(v_1), \ldots, \psi(v_m)) \in R$.

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

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 $\langle s', R_{s'}\rangle$
enddo
Step 5 **output**($\mathcal{P}$)

Fig. 3. Algorithm IRREDUCIBILITY

4.3.6 *The splitting and semisplitting property.* 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 $f\big|_B$ 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 representative. 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 Fig 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 $\mathcal{P}'_U$, that is $\mathcal{P}'$ restricted to U, and $\mathcal{P}'_{U'}$, 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 $v, w \in U$, $\mathcal{S}'_{\{v,w\}} = \mathcal{S}_{\{v,w\}} \cap B^2$ 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 $u \in W$ 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 7, it has a solution. $\Box$

4.4 Problems of bounded width

Problems of bounded width or of bounded relational width are of particular interest, because there exist simple standard algorithms solving such problems. Analising thoroughly the 10 properties we are able to characterise those constraint satisfaction problems over a 3-element domain which have bounded (relational) width.

THEOREM 7. *Let Γ be a tractable constraint language over a 3-element set A.*
(1) If Γ has a non-surjective unary polymorphism then $\mathrm{CSP}(\Gamma)$ *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 $g \in \mathsf{Inv}\ \Gamma$ such that $g\big|_{f(A)}$ is a semilattice or majority operation.*
(2) If every unary polymorphism of Γ is surjective then $\mathrm{CSP}(\Gamma)$ *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 operation.*
(3) If $\mathrm{CSP}(\Gamma)$ *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 non-surjective 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 $|f(A)| = 1$ or $|f(A)| = 2$ and there exist $g \in \mathsf{Inv}\ \mathcal{H}$ such that $g\big|_{f(A)}$ 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 $\mathbb{A}_\Gamma$ itself or every 2-element factor of $\mathbb{A}_\Gamma$ has a semilattice or majority operation.*

Proof. (1) If there is a polymorphism, f, of Γ with $|f(A)| = 1$ then f is a constant operation and every problem from $\mathrm{CSP}(\Gamma)$ 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 [20], any problem $\mathcal{P} = (V; A; \mathcal{C})$ is equivalent to the problem $\mathcal{P}' = (V; f(A); \mathcal{C}')$ 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. By Proposition 7, if $\mathcal{P}'$ has 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.

(2) Notice first that the algebra $\mathbb{A}'$ in Theorem 5 can be chosen to be $\mathbb{A}$ unless it satisfies conditions considered in Section 6.5. However, even in that case if $\mathbb{A}$ itself or every its 2-element factor has a semilattice or majority term operation then $\mathbb{A}'$ or every 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 $\mathrm{CSP}(\Gamma)$ is of relational width 3 by Proposition 7 and results of [5]. If $\mathbb{A}_\Gamma$ has a Mal'tsev term operation and every its 2-element factor has a semilattice or majority term operation, then by results of [2]

If $\mathbb{A}_\Gamma$ satisfies the partial zero property then $\mathrm{CSP}(\Gamma)$ 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 can be reduced to a 2-valid problem, which is also 3-minimal. By the assumption made for every domain B of this 2-valid problem there is a term operation f such that $f\big|_B$ is a semilattice or majority operation. By Proposition 7, the problem has a soluiton.

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. Since there is a term operation f of $\mathbb{A}_\Gamma$ such that $f\big|_B$ is a semilattice or majority operation, if $\mathcal{P}$ is a 3-minimal problem instance then every problem of the form $\mathcal{P}'_{W_i}$ has a solution and, hence, the problem $\mathcal{P}'$, 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 factor-problem, $\mathcal{P}^\theta$, is also 3-minimal. Since every domain of the factor-problem is a factor of $\mathbb{A}_\Gamma$, by Proposition 7, the factor-problem has a solution. As we already observed, every problem of the form $\mathcal{P}_{W_i}$ has a solution. Therefore, the mapping ψ defined in Section 4.3.5 is a solution to $\mathcal{P}$.

5. RECOGNISING TRACTABLE CASES

In a practical perspective, we need a method that allows us to recognise if a given constraint language Γ is tractable. The following problem is, therefore, very tempting.

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

Schaefer's Dichotomy Theorem [34] does not solve this problem satisfactory. Indeed, it can be easily verified if a relation is of type (1) or (2), however, the way of recognising the types (3)–(6) is not obvious (see also [25]). Theorem 3, the algebraic version of Schaefer's result, fills this gap: to check the tractability of a Boolean constraint language one just have to check if all relations from the language are invariant under one of the 6 Boolean operations.

In the general case, such a method can hopefully be derived from a description of tractable algebras. For example, in [7], a polynomial time algorithm has been exhibited that checks if a finite algebra, whose basic operations are given explicitly by their operation tables, satisfies (NO-G-SET). Therefore, if Conjecture 1 holds then the tractability of an algebra can be tested in polynomial time. In particular, this algorithm sounds in the case of 3-element algebras.

However, the algorithm does not solve TRACTABLE-LANGUAGE even under the assumption of Conjecture 1, because in this problem we are given a constraint language, not an algebra. Actually, we need to solve the problem

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

By the results of [7], this problem is NP-complete. However, its restricted version remains tractable.

NO-G-SET-LANGUAGE(k). Given finite constraint language Γ on a finite set A, $|A| \leq k$, 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 8. *There is a polynomial time algorithm that given a constraint language Γ on a 3-element set determines if Γ is tractable.*

An example of such an algorithm is provided by the general algorithm from [7]. 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 a 3-element algebra $\mathbb{A}$ has a 2-element subuniverse or a nontrivial congruence, and there is a term operation f which is not a projection on the subalgebra or the factor-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 [36]. In particular, if a strictly simple algebra satisfies (NO-G-SET) then one of the following operations is 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 (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 4 Boolean operations: $\wedge$, $\vee$, the majority operation $(x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$, the affine operation $x - y + z$ (since $\mathbb{A}$ is idempotent, a constant cannot be its term operation).

As is well known [28], 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 3 steps, the algorithm shown in Fig.4 construct the language Γ^{id}.

This algorithm is polynomial time, because the hardest step, 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 0??

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

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** $S = C = \varnothing$ **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.

6.1 Prerequisites

In this section we state and prove some auxiliary statements. An element $a \in A$ is said to be a *right-zero* [*left-zero*] with respect to the 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.

Definition 23. *Let $A = \{a,b,c\}$. A binary operation f is said to be $(a-b)$-operation if b, c are left- (right-)zeroes, and $\{f(a,a), f(a,b), f(a,c)\} = \{a,b\}$ ($\{f(a,a), f(b,a), f(c,a)\} = \{a,b\}$).*

Definition 24. *Let $A = \{a,b,c\}$. A binary operation f is said to be* zero-operation *if one of the following conditions holds*

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

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

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

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

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

otherwise.

The tuple $\mathbf{a}_0$ can be set to be $\mathbf{a}$. Suppose that a tuple $\mathbf{a}_k$ is already found. Then set $\mathbf{a}_{k+1} = \mathbf{a}_k$ if $\mathbf{a}_k[k+1] \in \{b,c\}$, or $R_{k+1} = \{a,b\}$ and $f(a,b) = a$; 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 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 of $\mathbb{A}$, on which f 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

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

□

An important particular type of zero-operations rises from the following definition.

DEFINITION 25. *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 4 cases depending on the number of 2-element subalgebras and nontrivial homomorphic images of the algebra $\mathbb{A}$. Recall that an algebra is said to be *simple* if it has only two homomorphic images: one-element, and the algebra itself.

6.2 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 Á. Szendrei [36]. To formulate Á.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 $g \in G$, and $\mathrm{F}(G)$ denotes 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, $T(\overline{A})$ the group of translations $\{x+a \mid a \in A\}$, $\mathsf{End}\ {}_F\overline{A}$ 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

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

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.

THEOREM 9. *[36] 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 nonidentity 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;

(d) $(A,\mathrm{F}(G)\cap \mathrm{F}_k^0)$ 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 nonidentity member of G;

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

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

In [9], 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°), (d), (e); otherwise it is NP-complete. As is easily seen, exactly those algebras satisfy the condition of Theorem 0**??**. 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

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

and 0 is a zero-element with respect this operation; in the case (e) the algebra has a term semilattice operation. Since a set of relations closed under a majority operation is of width 3 ([22]), and any semilattice has a zero-element, Theorem 0**??** holds for 3-element strictly simple algebras.

6.3 Simple algebras with 1 or 2 subalgebras

In this subsection we assume that $\mathbb{A}$ has the subalgebra $\mathbb{B} = \{0,1\}$, but at least one of $\{0,2\},\{1,2\}$ is not.

We need several particular operations on the 3-element set:

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

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

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

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

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. Suppose that $\{1,2\}$ is not a subuniverse. In 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 $f(x_1,\ldots,x_n)$, and $a_1,\ldots,a_n \in \{1,2\}$ such that $f(a_1,\ldots,a_n) = 0$. Without loss of generality we may assume that $a_1 = \ldots = a_k = 1$, $a_{k+1} = \ldots = a_n = 2$. Then the operation $f(\underbrace{x,\ldots,x}_{k},y,\ldots,y)$ 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 get 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 $f(x,y)$ 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 the operation $g(x,y) = f(f(x,y),x)$, $g(x, g(x,y))$ 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, 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 $f\big|_B$ 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). □

If $\mathbb{A}$ has a zero-element with respect to a binary term operation, or a zero-operation, or a $(2-a)$-operation, then by Lemmas 4,3, $\mathbb{A}$ 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 properties listed in Theorem 0**??**.

Lemma 8. *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 g a term operation which is either a semilattice, or majority, or minority operation on B. Then*

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

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

– *if $g\big|_B$ is a minority operation then $\mathbb{A}$ is $\{0,1\}$-rectangular.*

Proof. We prove the lemma in the case when f is the operation (1); the other 2 cases are quite similar.

If a term operation g is such that $g\big|_{\{0,1\}}$ 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 $\begin{pmatrix} a \\ b \end{pmatrix} \in R_{\{k,l\}}$, and moreover, for any $\begin{pmatrix} a \\ b \end{pmatrix} \in R_{\{k,l\}}$ 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 $\mathbf{a}[1] = a$. Then, there is $\mathbf{b} \in R$ with $\mathbf{b}[2] \in B$; set $\mathbf{c} = f(\mathbf{b}, \mathbf{a}_1)$. As is easily seen, $\mathbf{c}[1] = \mathbf{a}[1] = a$, $\mathbf{c}[2] \in B$. Further, let $\begin{pmatrix} a \\ b \end{pmatrix} \in R_{\{1,2\}} \cap B^2$, and let $\mathbf{a}_2 \in R$ be such that $\mathbf{a}[1] = a, \mathbf{a}[2] = b$. Then, for any $k \le n$, there is $\mathbf{a}_k \in R$ such that $\mathbf{a}[1] = a, \mathbf{a}[2] = b$, and $\mathbf{a}[i] \in B$ 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 $R_k \ne A$. 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.

Furthermore, denote $W = \{i \in \underline{n} \mid R_i = A\}$, and take $\mathbf{a} \in B^{|W|}$ such that $\begin{pmatrix} \mathbf{a}[i] \\ \mathbf{a}[j] \end{pmatrix} \in R_{\{i,j\}}$, 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 [21], there is $\mathbf{b} \in R$ such that $\mathbf{b}\big|_W = \mathbf{a}$.

Let $R = \{\mathbf{c}_1, \dots, \mathbf{c}_m\}$, and

$$\mathbf{c} = f(\dots f(\mathbf{c}_1, \mathbf{c}_2), \dots, \mathbf{c}_m).$$

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

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

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 $i \in \underline{k}$, and let us suppose 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 $J = \{1, \dots, l\}$ and $U = W_1 \cup \dots \cup W_l$. There exists a tuple $\mathbf{b} \in R$ such that $\mathbf{b}_{W_1} \in \{0,1\}^{|W_1|}, \mathbf{b}_{W_l} = (2, \dots, 2)$, or vice versa. It will not be loss of generality if we suppose that $\mathbf{b}_{W_l} = (2, \dots, 2)$, and there is $m < l$ such that, for any $i \le l$, $\mathbf{b}_{W_i} = (2, \dots, 2)$ 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 4 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 $W_1 \cup \dots \cup W_{m-1}$ and is nonempty; the second part U_2 equals to $(W_1 \cup \dots \cup W_{m-1}) - U_1$, and consists of those positions in which $\mathbf{a}, \mathbf{b}$ have $0, 1$ while $\mathbf{c}$ has 2; the third part, $U_3 \subseteq W_m \cup \dots \cup W_l$, 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 $\mathbf{a}, \mathbf{b}$ 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

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

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

LEMMA 9. *Let $\mathbb{A}$ have a term operation f which is one of the operations (4),(5),(6),(9). Then*

– $\mathbb{A}$ *satisfies* $(2-0)$*-replacement property if* f *is (4);*
– $\mathbb{A}$ *satisfies* $(2-1)$*-replacement property if* f *is the operation (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.
(4) Let f is the operation (4). Take an (n-ary) relation $R \in \mathsf{Inv}\, F$. We prove that, for any $\mathbf{a} \in R$, and any $k \leq n$, there is $\mathbf{a}_k \in R$ such that

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

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 $\mathbf{b}[k+1] = 1$.

(6) Let f denote 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); and the required result follows from Lemma 8.

The arguments for the operations (5) and (9) are quite similar. □

Recall that we denote θ the equivalence relation whose classes are $\{0,1\}$ and $\{2\}$.

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

(b) If $\mathbb{A}$ *is simple and has a term operation which is one of the operations (8),(11),(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* $\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0 & 2 \\ 1 & 2 & 1 \end{array}$*;*

. *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 is an operation $g(x_1,\dots,x_n)$ and $\mathbf{a},\mathbf{b}\in A^n$ such $(\mathbf{a}[i],\mathbf{b}[i])\in\theta$, but $(g(\mathbf{a}[1],\dots,\mathbf{a}[n]),g(\mathbf{b}[1],\dots,\mathbf{b}[n]))\notin\theta$. It is not hard to see that g, $\mathbf{a},\mathbf{b}$ can be chosen such that $\mathbf{a},\mathbf{b}$ differ only in one coordinate position. Without loss of generality, let $\mathbf{a}[1]\ne\mathbf{b}[1]$, $\mathbf{a}[2]=\mathbf{b}[2]=\dots=\mathbf{a}[p]=\mathbf{b}[p]=0$, $\mathbf{a}[p{+}1]=\mathbf{b}[p{+}1]=\dots=\mathbf{a}[r]=\mathbf{b}[r]=1$, $\mathbf{a}[r{+}1]=\mathbf{b}[r{+}1]=\dots=\mathbf{a}[n]=\mathbf{b}[n]=2$. 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),(16) then $f(0,2)\in\{0,1\}$, $f(1,2)=0$, and, for the operation $h(x,y)=g(f(x,y),x,y)$, we have $h(1,2)=g(0,1,2)=2$, and $h(0,2)=g(0/1,0,2)\in\{0,1\}$, i.e. h destroys θ. If f is one of (7),(13),(15) then $f(0,2)=f(1,2)=1$, for the operation $h(x,y)=f(x,f(x,y),y)$, we have $h(0,2)=g(0,1,2)=2$, and $h(1,2)=g(1,1,2)\in\{0,1\}$, i.e. 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 $f(0,2)=f(1,2)=0$, and, for the operation

$h(x,y) = g(f(x,y),x,y)$, we have $h(0,2) = g(0,0,2) = 2$, $h(1,2) = g(0,1,2) \in \{0,1\}$, and h destroys θ. Analogously, if f is one of (7),(13),(15), and so $f(0,2) = f(1,2) = 1$, then for h obtained in the same way we have $h(0,2) \in \{0,1\}$, $h(1,2) = 2$, i.e. 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), (14), then the operation $h(x,y) = g(f(x,y),x,y)$ satisfies the conditions $h(0,2) = g(0,0,2) = 0$, $h(1,2) = g(0,1,2) = 2$, and therefore, destroys θ. If $f(0,2) = f(1,2) = 1$, that is f is one of the operations (7),(13),(15), then h satisfies the conditions $h(0,2) = g(1,0,2) = 2$, $h(1,2) = g(1,1,2) = 1$.

Finally, if $f(0,2) = 1$, $f(1,2) = 0$, that is f is one of the operations (8),(11),(16), then each of the operations $g(2,y,z), g(x,2,z)$ either satisfies the same conditions as $g(x,y,2)$, 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. □

Lemma 11. *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),(16) then either 2 is a zero-element with respect to some binary term operation of $\mathbb{A}$, or $\mathbb{A}$ has a binary conservative commutative term 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, $g(0,2) = 0, g(1,2) = 2$, and consider 4 cases.

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

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

Subcase A.3. $g(2,0) = 2$, $g(2,1) = 1$.

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

Subcase A.4. All other cases.

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

CASE B.

g	0	1	2
0	0	0	0
1	1	1	2
2			2

.

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

SUBCASE B.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 B.4.

SUBCASE B.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),(15), then $g(f(x, y), y)$ is a zero-operation. Otherwise, the operation $g(x, f(y, x))$ satisfies the conditions of Subcase B.2.

SUBCASE B.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),(15) is a term operation of $\mathbb{A}$. Therefore, either, the conditions of Lemma 9 or Lemma 8 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, or one of the operations (7),(8),(11),(13),(15),(16) is a term operation of $\mathbb{A}$. Denote this term operation by h. If h is one of (7),(13),(15), i.e. $h(0, 2) = h(1, 2) = 1$, then 2 is a zero-element with respect to the operation $g(h(x, y), y)$. If h is one of (8),(11),(16) then $g(h(x, y), y)$ satisfies the conditions of one of Cases E–H.

CASE C.

g	0	1	2
0	0	0	0
1	0	1	2
2			2

.

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 8, 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 B.4, either the conditions of Lemma 9 or Lemma 8 hold, or one of (7),(8),(11),(13),(15),(16) is a term operation of $\mathbb{A}$; denote this operation by h. If h is one of (7),(13),(15) then 2 is a zero-element with respect to the operation $g(h(x, y), y)$. If f is one of (8),(11),(16) then $g(h(x, y), y)$ satisfies the conditions of one of Cases E–H.

CASE D.

g	0	1	2
0	0	1	0
1	1	1	2
2			2

.

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

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

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

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

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

Subcase F.3. All other cases.

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

Subcase G.1. $g(2, 0) = 0$. The operation g is a zero-operation.

Subcase G.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 G.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 to the previous cases.

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

If $g(2, 0) = 2, g(2, 1) = 0$ then g is the operation (6), and, by Lemma 9, $\mathbb{A}$ satisfies the $\{1, 2\}$-extendibility property. If $g(2, 0) = g(2, 1) = 2$ then g is a zero-operation, and if $g(2, 1) = 2, g(2, 0) \in \{0, 1\}$ then $g(g(x, y), y)$ is a zero-operation. Further, 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. □

Lemma 12. *If $\mathbb{A}$ satisfies the conditions of Lemma 10 and has no binary opera-*

tion destroying θ, then either a zero-operation, or $(2-0)$- or $(2-1)$-operation, or the operation (1) is a term operation of $\mathbb{A}$.

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' on $\{0,1\}$ 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, $g'(y, g'(y,x))$ is a zero operation. In the case when $g'|_{\{0,1\}}$ is a semilattice operation, g' is a zero-operation. Further, 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 θ, a contradiction with the conditions of Lemma 12. 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 12, 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 $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{array}$, and therefore, h is a zero-operation.

□

6.4 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, and 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.

6.4.1 $\mathbb{A}$ 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 $f(x, f(y,x))$, the projections may be assumed to be first projections. We consider 3 subcases.

CASE 1. f is a semilattice operation on all three 2-element subuniverses. In this case, f is a commutative conservative binary operation, and, by [5], $\mathbb{A}$ is of width 3.

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. $f\big|_{\{0,2\}}$ 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, and therefore, by Lemma 4, $\mathbb{A}$ satisfies the partial zero property. In the second case, f is the operation (2), hence, by Lemma 8, $\mathbb{A}$ satisfies the $\{1,2\}$-rectangularity property.

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

Then the Cayley table of f is one of the following $\begin{array}{c|ccc} f & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 2 \end{array}$, $\begin{array}{c|ccc} . & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 \end{array}$ In the first case, 1 is a zero-element with respect to f. In 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 $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$, and 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 subalgebras 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, the affine operation $x - y + z$.

LEMMA 13. *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 $h\big|_B$ is the majority operation if there is a term operation f of $\mathbb{A}$ such that $f\big|_B$ is the majority operation, and $h\big|_B$ is the minority operation otherwise.*

Proof. Let $B \subseteq A$, $|B| = 2$, $f' = f\big|_B$, 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 done.

CLAIM 1. If $f_i(x,y) = y$ for some i and f' is not a projection, then there is a term operation g of $\mathbb{A}$ such that $g\big|_B$ 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) $f_2 = f_3 = x$. In this case set $g(x,y,z) = f(f(x,y,z),y,z)$; we have

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

Denote by B_1, B_2, B_3 the sets $\{0,1\}, \{0,2\}, \{1,2\}$ respectively. By Claim 1, there are ternary term operations g_1, g_2, g_3 of $\mathbb{A}$ such that $g_i\big|_{B_i}$ is the majority operation if there are a term operation f of $\mathbb{A}$ such that $f\big|_{B_i}$ is the majority operation, $g_i\big|_{B_i}$ is the minority otherwise, and $g_i\big|_{B_j}$ 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 $g_{ij}\big|_{B_i} = g_i\big|_{B_i}$. 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 $g_1\big|_{B_1}, g_2\big|_{B_2}$ are majority operations, but $g_1\big|_{B_2}$ is the minority operation. Then for $g'(x,y) = g_1(x,x,y)$, we have $g'\big|_{B_1}(x,y) = x, g'\big|_{B_2}(x,y) = y$; 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 $g''\big|_{B_3} \neq g_3\big|_{B_3}$ we repeat this procedure for B_1, B_3. $\square$

6.4.2 All three 2-element subalgebras have the majority term operation, but no semilattice term operation.. By Lemma 13, there is a term operation f which is the majority operation on each 2-element subset. This means that f is a majority operation on A, and therefore, $\mathbb{A}$ has a majority term operation.

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

6.4.4 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 semilattice nor majority operation.. By Lemma 13, there is a term operation f of $\mathbb{A}$ such that $f\big|_{B_1}$ is the minority operation, and $f\big|_{B_2}, f\big|_{B_3}$ 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 14. *The algebra $\mathbb{A}$ satisfies the B_2-semisplitting property.*

Proof. For $I, J \subseteq \underline{n}$ $I \cap J = \varnothing$, and $\mathbf{a} \in R_I, \mathbf{b} \in R_J$, we write $(\mathbf{a},\mathbf{b})$ for the $|I|+|J|$-tuple $\mathbf{c} = (c_I)_{i \in I \cup J}$ with $c_i = a_i$ if $i \in I$, and $c_i = b_i$ 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

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

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 $\begin{pmatrix}\mathbf{a}\\2\end{pmatrix} = g\left(\begin{pmatrix}\mathbf{a}\\b\end{pmatrix}, \begin{pmatrix}\mathbf{c}\\2\end{pmatrix}\right)$ belongs to $R_{W_1 \cup \{i\}}$. If $a \in \{0,1\}$ then $\begin{pmatrix}\mathbf{a}\\a\end{pmatrix} = g\left(\begin{pmatrix}\mathbf{a}\\2\end{pmatrix}, \begin{pmatrix}\mathbf{b}\\a\end{pmatrix}\right)$.

Claim 2. For any $i, j \in W$, any $(a, \mathbf{a}) \in R_{W_1 \cup \{i\}}$ 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,

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

and 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 $i \in I \cap W$, belongs to $R_{W_1 \cup I \cup \{i_1,i_2\}}$.

The base case of induction, $I = \varnothing$, 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 $I \neq \varnothing$. 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 $\{\mathbf{c}[i], \mathbf{d}[i]\} \neq \{0, 1\}$, and 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 $g\left(\begin{pmatrix}\mathbf{d}\\\mathbf{a}\end{pmatrix},\begin{pmatrix}\mathbf{b}\\\mathbf{c}\end{pmatrix}\right)=\begin{pmatrix}\mathbf{b}\\\mathbf{a}\end{pmatrix}$.

To finish the proof of Lemma 14, we just should put $I = W \cup W_2 \cup W_3$ in Claim 4. □

By the assumptions made, f is the majority operation on B_2, and therefore, in this case, $\mathbb{A}$ satisfies the conditions of Theorem 0**??**(5).

6.4.5 One of the 2-element subalgebras has the majority term operation, but no semilattice operation; and the two others have the minority term operation, but no semilattice or majority operation.. By Lemma 13, $\mathbb{A}$ 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.

Let θ denote the equivalence relation whose classes are $\overline{0} = \{0, 1\}$ and $\overline{2} = \{2\}$. The class containing an element $a \in A$ will be denoted by a^θ, and for an n-ary relation R, we set $R^\theta = \{(a_1^\theta, \ldots, a_n^\theta) \mid (a_1, \ldots, a_n) \in R\}$.

LEMMA 15. *Let $R \in \mathsf{Inv}$ 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 non-identity bijection with the fixed point 2, or $R^\theta \in \{\mu_1, \mu_2\}$ where*

$$\mu_1 = \begin{pmatrix} \overline{0} & \overline{2} \\ \overline{0} & \overline{2} \end{pmatrix}, \quad \mu_1 = \begin{pmatrix} \overline{0} & \overline{0} \\ \overline{2} & \overline{2} \end{pmatrix}.$$

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 $g\left(\begin{pmatrix} b \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}\right) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \in R$ 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 $\begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, and either the tuple $\begin{pmatrix} 2 \\ 2 \end{pmatrix}$ or a tuple of the form $\begin{pmatrix} a \\ b \end{pmatrix}$ where $a, b \in \{0, 1\}$.

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

CASE 1. $\begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \in R.$

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

CASE 2. $\begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix} \in R.$

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

We consider two cases.

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

In this case θ does not belong to $\mathsf{Inv}\ F$, and 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 not 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

LEMMA 16. *Let $R \in \mathsf{Inv}$ be an (n-ary) relation such that $R_1 = \ldots = R_n = A$, 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 15, and therefore, is either a graph of a bijection with the fixed point 2, or A^2. (Notice that in both cases $\begin{pmatrix} 2 \\ 2 \end{pmatrix} \in R_a$.) 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 15, $R_a = A^2$. 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 $R \in \mathsf{Inv}$ is an ($(n + 1)$-ary) relation. As before, let $R_a = \{(a_2, \ldots, a_{n+1}) \mid (a, a_2, \ldots, a_{n+1}) \in R\}$. By the induction hypothesis, $R_a = A^n$ for any $a \in A$, and therefore, $R = A^{n+1}$. $\square$

LEMMA 17. *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 an even stronger condition.

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 16, $R_W = A^{|W|}$, 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 $v = 1$, $W_2 \cup W_3 =$

$\{2,\ldots,k\}$. Notice first that if $\begin{pmatrix}1\\ \mathbf{a}\end{pmatrix} \in R'$ then $\begin{pmatrix}0\\ \mathbf{a}\end{pmatrix} \in R'$, and vice versa. Indeed, there is $\mathbf{b} \in R'_{W_2\cup W_3}$ such that $\begin{pmatrix}0\\ \mathbf{b}\end{pmatrix} \in R'$; and $\begin{pmatrix}0\\ \mathbf{a}\end{pmatrix} = g\left(\begin{pmatrix}0\\ \mathbf{b}\end{pmatrix}, \begin{pmatrix}1\\ \mathbf{a}\end{pmatrix}\right) \in R'$.

Furthermore, if $(2,\mathbf{a}), (0,\mathbf{a}) \in R'$, for 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

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

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

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

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

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 16, $R_{\mathbf{a}} = A^{|W|}$; therefore, $R_{W\cup W_2\cup W_3} = R_W \times R_{W_2\cup W_3}$. The lemma is proved. □

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

LEMMA 18. *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 $\delta(R)$. It will be convenient for us to denote $\overline{2}$ the tuple consisting of 2s; the length of this tuple will 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}, $W_J = W' \cup W_{i_1} \cup \ldots \cup W_{i_l}$. The base case of induction is obvious, $\mathbf{c}_\varnothing = \mathbf{b}$. The next case is $|J| = 1$. It will not be loss of generality if we assume $J = \{1\}$. There is $\mathbf{c}_0 \in R_{W'}$ 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 the we 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 $\delta(R)$, 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 $(\mathbf{c}_0, \mathbf{c}_1, \overline{2}) \in R_{W' \cup W_1 \cup W_2}$ $((\mathbf{c}_0, \overline{2}, \mathbf{c}_2) \in R_{W' \cup W_1 \cup W_2})$, and $\mathbf{c}_1 \in \{0,1\}^{|W_1|}$ ($\mathbf{c}_2 \in \{0,1\}^{|W_2|}$). 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 $K \subseteq \{1, \dots, k\}$ is such that $|K| = m$. Again, without loss of generality, assume that $K = \{1, \dots, m\}$.

By the induction hypothesis, there are tuples of the form $\mathbf{b}^1 = (\mathbf{b}_0, \mathbf{a}_1, \dots, \mathbf{a}_{m-1}, \mathbf{d}^1_m)$, $\mathbf{b}^2 = (\mathbf{b}_0, \mathbf{a}_1, \dots, \mathbf{a}_{m-2}, \mathbf{d}^2_{m-1}, \mathbf{a}_m)$, $\mathbf{b}^3 = (\mathbf{b}_0, \mathbf{a}_1, \dots, \mathbf{a}_{m-3}, \mathbf{d}^3_{m-2}, \mathbf{a}_{m-1}, \mathbf{a}_m)$ 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 $i \in \{1, \dots, m-1\}$, $g(\mathbf{b}^1, \mathbf{b})$ is a tuple 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. □

6.5 Non-simple algebras

In this subsection we assume that $\mathbb{A}$ has a proper congruence. Without loss of generality, let us suppose that the equivalence relation θ, whose classes are $\{0,1\}, \{2\}$, is a congruence of $\mathbb{A}$. Then, 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, $f(\mathbf{a}[1], \ldots, \mathbf{a}[n]) \in \{0,1\}$, that is $B = \{0,1\}$ is a subuniverse. Denote $\mathbb{B}$ 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.

6.5.1 $f|_B$ 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 certain its reduct satisfies $(0-1)$-replacement property.

Since f preserve θ, the operation table of $f'(x,y) = f(f(x,y),y)$ is $\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & 0 & 1 & 0/1 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{array}$,

and 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 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.

6.5.2 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 $g^\theta(2^\theta, 0^\theta) = g^\theta(0^\theta, 2^\theta) = 0^\theta$.

Case 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 3,8, $\mathbb{A}$ satisfies one on the conditions listed in Theorem 0**??**.

Case 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 3 cases we assume that there are term operations f, g such that $f|_{\{0,1\}}, g^\theta$ are minority or majority operations.

6.5.3 $\mathbb{A}$ is conservative.. In this case we are in the conditions of Subsection 5.2.

6.5.4 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 9,8, some of them lead to a good property 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.

6.5.5 $\{0,1\}$ is the only subuniverse of $\mathbb{A}$.. As above, by making use of Lemmas 7,9,8, we have to consider only the cases when (16), or both (14), (15) are a term operation of $\mathbb{A}$.

Case 1. (14),(15) are term operations of $\mathbb{A}$.
Let $r(x,y)$ denote the operation (14). For any term operation $h'(x_1,\dots,x_n)$ of $\mathbb{A}$, the operation

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

satisfies the conditions: $\hat{h}(x_1,\dots,x_n) = 2$ if and only if $h(x_1,\dots,x_n) = 2$, 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 $\hat{f}\big|_{\{0,1\}}$ $(\hat{g}^\theta)$ is a minority or a majority operation if $f\big|_{\{0,1\}}$ (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 (No-G-Set), and $\{0,2\}$ is a subuniverse of $\mathbb{A}'$; We are in the conditions of Case 5.5.4.

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

Lemma 19. *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 has a term operation g' which is minority on $\mathbb{A}/_\theta$, $g'\big|_{\{0,1\}} = g\big|_{\{0,1\}}$, and g' preserves $\{0,2\},\{1,2\}$.*

Proof. Since, for any $x,y \in \{0,1\}$, $g(2,x,y) = g(x,2,y) = g(x,y,2) = 2$, we just are 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 $g'(x,y,2) = g(x,y,2)$, $g'\big|_{\{0,1\}} = g\big|_{\{0,1\}}$, and $g'(2,2,x) = x$. Repeating the same procedure for all three variables we get the required operation. $\square$

Lemma 20. *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 has a term operation g' which is majority on $\mathbb{A}/_\theta$, $g'\big|_{\{0,1\}} = g\big|_{\{0,1\}}$, and g' preserves $\{0,2\},\{1,2\}$.*

Proof. The proof is quite similar to that of Lemma 19, 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 $g(x,y,z) = 2$, that is g'^{θ} is a majority operation; $g'|_{\{0,1\}} = g|_{\{0,1\}}$, $g'(x,2,x) = g(x,2,x)$, $g'(x,x,2) = g(x,x,2)$, and $g'(2,x,x) = x$. Repeating the same procedure for all pairs of variables we get the required operation. □

Since $g|_{\{0,1\}}$ is an idempotent operation, and the subalgebra of $\mathbb{A}$ with the universe $\{0,1\}$ has no semilattice term operation, $g|_{\{0,1\}}(x,x,y), g|_{\{0,1\}}(x,y,x), g|_{\{0,1\}}(y,x,x) \in \{x,y\}$. If g^{θ} and $g|_{\{0,1\}}$ are minority (majority) operations, then applying one Lemmas 19 (Lemma 20) to g, we get a minority (majority) operation on A. Noticing that a minority operation is Mal'tsev operation, the algebra $\mathbb{A}$ has 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 21. *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 $f(x,y,z), g(x,y,z) \in \{0,1\}$, for any $x,y,z \in \{0,1\}$, we have $g'|_{\{0,1\}}(x,y,z) = f|_{\{0,1\}}(x,y,z)$. Further, the equality $h^{\theta}(x,y) = x$, implies $g'^{\theta}(x,y,z) = g^{\theta}(x,y,z)$. The lemma is proved. □

Finally, applying Lemmas 19,20 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 a minority operation. If either both of these operations are minority, or both are majority, then $\mathbb{A}$ has Mal'tsev or a majority term operation. Otherwise, the algebra $\mathbb{A}' = (A; g'')$ is a reduct of $\mathbb{A}$, and is conservative. 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 the 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 $\begin{pmatrix} h'(0,1) \\ h'(2,1) \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ witnesses that h' destroys the equivalence relation with classes

$\{0,2\},\{1\}$; while the pair $\begin{pmatrix} h'(1,0) \\ h'(2,0) \end{pmatrix}$ 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 5.5.3.

7. CONCLUSION

In fact, Theorem 0**??** implies a stronger result than that claimed in Theorem 4. The difference appears when considering infinite constraint languages satisfying the conditions of Conjecture 1. Theorem 4 claims that, for any finite subset Γ of such a language, there is its own polynomial time algorithm $\mathrm{Alg}(\Gamma)$ solving $\mathrm{CSP}(\Gamma)$, and for different subsets the corresponding algorithms can be quite different. Theorem 0**??** yields a uniform polynomial time algorithm that solves any problem from the class risen from the constraint language. Moreover, from the proof of Theorem 0**??** a general algorithm can be derived, which solves any problem instance $\mathcal{P}$ on a 3-element set provided that $\mathcal{P} \in \mathrm{CSP}(\Gamma)$, for some tractable Γ.

As a matter of fact, Theorem 0**??** is proved by the 'rough force' method, that is by analysing a large number of operations which provide the condition (No-G-Set). We believe that development of algebraic tools and more subtle usage of results from universal algebra will make it possible to obtain dichotomy results for larger domains, and eventually, for an arbitrary finite domain.

REFERENCES

J.F. Allen. *Natural Language Understanding.* Benjamin Cummihgs, 1994.

A. Bulatov. Three-element Mal'tsev algebras. *Acta Sci. Math (Szeged).* To appear.

A.A. Bulatov. Mal'tsev constraints are tractable. Technical Report PRG-RR-02-05, Computing Laboratory, University of Oxford, Oxford, UK, 2002.

A.A. Bulatov and P.G. Jeavons. Algebraic approach to multi-sorted constraits. To appear at CP'03.

A.A. Bulatov and P.G. Jeavons. Tractable constraints closed under a binary operation. Technical Report PRG-TR-12-00, Computing Laboratory, University of Oxford, Oxford, UK, 2000.

A.A. Bulatov and P.G. Jeavons. Algebraic approach to multi-sorted constraints. Technical Report PRG-RR-01-18, Computing Laboratory, University of Oxford, Oxford, UK, 2001.

A.A. Bulatov and P.G. Jeavons. Algebraic structures in combinatorial problems. Technical Report MATH-AL-4-2001, Technische universität Dresden, Dresden, Germany, 2001.

A.A. Bulatov, P.G. Jeavons, and A.A. Krokhin. The complexity of maximal constraint languages. In *Proceedings of the 33rd Annual ACM Simposium on Theory of Computing*, pages 667–674, Hersonissos, Crete, Greece, July 2001. ACM Press.

A.A. Bulatov, A.A. Krokhin, and P.G. Jeavons. Constraint satisfaction problems and finite algebras. In *Proceedings of 27th International Colloquium on Automata, Languages and Programming—ICALP'00*, volume 1853 of *Lecture Notes in Computer Science*, pages 272–282. Springer-Verlag, 2000.

S. Burris and H.P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics.* Springer-Verlag, New York-Berlin, 1981.

N. Creignou, S. Khanna, and M. Sudan. *Complexity Classifications of Boolean Constraint Satisfaction Problems*, volume 7 of *SIAM Monographs on Discrete Mathematics and Applications.* SIAM, 2001.

V. Dalmau. *Computational Complexity of Problems over Generalised Formulas.* PhD thesis, Department LSI of the Universitat Politecnica de Catalunya (UPC), Barcelona., March, 2000.

R. Dechter and A. Dechter. Structure-driven algorithms for truth maintenance. *Artificial Intelligence*, 82(1-2):1–20, 1996.

R. Dechter and I. Meiri. Experimental evaluation of preprocessing algorithms for constraint satisfaction problems. *Artificial Intelligence*, 68:211–241, 1994.

R. Dechter and J. Pearl. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence*, 34(1):1–38, 1988.

T. Feder and M.Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM Journal of Computing*, 28:57–104, 1998.

M. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA., 1979.

G. Gottlob, L. Leone, and F. Scarcello. A comparison of structural CSP decomposition methods. *Artificial Intelligence*, 124(2):243–282, 2000.

P.G. Jeavons. Constructing constraints. In *Proceedings 4th International Conference on Constraint Programming—CP'98 (Pisa, October 1998)*, volume 1520 of *Lecture Notes in Computer Science*, pages 2–16. Springer-Verlag, 1998.

P.G. Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.

P.G. Jeavons, D.A. Cohen, and M.C. Cooper. Constraints, consistency and closure. *Artificial Intelligence*, 101(1-2):251–265, 1998.

P.G. Jeavons, D.A. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.

P.G. Jeavons, D.A. Cohen, and J.K. Pearson. Constraints and universal algebra. *Annals of Mathematics and Artificial Intelligence*, 24:51–67, 1998.

Ph.G. Kolaitis. Constraint satisfaction, databases, and logic. In *Proceedings 17th International Joint Conference on Artificial Intellignece, IJCAI'03*, 2003.

Ph.G. Kolaitis and M.Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61:302–332, 2000.

Ph.G. Kolaitis and M.Y. Vardi. A game-theoretic approach to constraint satisfaction. In *Proceedings 17th National (US) Conference on Artificial Intellignece, AAAI'00*, pages 175–181, 2000.

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

R.N. McKenzie, G.F. McNulty, and W.F. Taylor. *Algebras, Lattices and Varieties*, volume I. Wadsworth and Brooks, California, 1987.

U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

B.A. Nadel. Constraint satisfaction in Prolog: Complexity and theory-based heuristics. *Information Sciences*, 83(3-4):113–131, 1995.

B.A. Nadel and J. Lin. Automobile transmission design as a constraint satisfaction problem: Modeling the kinematik level. *Artificial Intelligence for Engineering Design, Anaysis and Manufacturing (AI EDAM)*, 5(3):137–171, 1991.

R. Pöschel and L.A. Kalužnin. *Funktionen- und Relationenalgebren*. DVW, Berlin, 1979.

E.L. Post. *The two-valued iterative systems of mathematical logic*, volume 5 of *Annals Mathematical Studies*. Princeton University Press, 1941.

T.J. Schaefer. The complexity of satisfiability problems. In *Proceedings 10th ACM Symposium on Theory of Computing (STOC'78)*, pages 216–226, 1978.

E. Schwalb and L. Vila. Temporal constraints: a survey. *Constraints*, 3(2-3):129–149, 1998.

A. Szendrei. Simple surjective algebras having no proper subalgebras. *Journal of the Australian Mathematical Society (Series A)*, 48:434–454, 1990.

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