Algebraic structures in combinatorial problems

Andrei A. Bulatov* Peter Jeavons

Abstract

The Constraint Satisfaction Problem provides a framework in which a wide variety of combinatorial problems can be expressed in a natural way. In [14, 17, 18, 5] Jeavons and co-authors develop an approach to study the constraint satisfaction problem which is based on characterising the complexity of problem classes via certain algebraic objects. In particular, each subclass of the constraint satisfaction problem with a finite set of values may be assigned a finite algebra such that the complexity of the subclass depends only on the algebra. In this paper we further develop this approach by showing how the complexity of problem classes is connected with properties of locally finite varieties, and survey, generalise, and organise results obtained to date.

2000 AMS Subject Classification 08A70

1 Introduction

Many combinatorial problems can be naturally expressed in the framework of the Constraint Satisfaction Problem (CSP). The aim in a constraint satisfaction problem is to assign values to a given set of variables such that certain constraints are satisfied on the values that can be assigned simultaneously to some subsets of variables [22, 25]. A standard satisfiability problem [10], where the variables must be assigned Boolean values, gives a typical example of such problem.

Investigation of the constraint satisfaction problem is one of the most rapidly growing areas of computer science, and developing constraint languages based on this research is recognised by the ACM as a strategic direction in computer science. The mathematical framework used to describe constraint satisfaction problems has strong links with many other areas of computer science such as database theory [36, 20], temporal and spatial

*When working on this paper the first author was supported by a grant of DAAD.

reasoning [32], machine vision [25], belief maintenance [7], technical design [27], natural language comprehension [1], programming language analysis [26], etc.

Throughout the paper we assume that P$\neq$NP. Recall that problems from P are said to be *tractable*. The general constraint satisfaction problem is known to be NP-complete, and hence, intractable [22, 25]. However, certain restrictions on the form of constraints may affect the complexity of the corresponding problem class and give rise to a tractable subclass of the general CSP. There is, therefore, a fundamental research direction aiming to recognise tractable subclasses of the constraint satisfaction problem. A progress in this direction may help in speeding up of general superpolynomial algorithms, and provide efficient algorithms for those applications, which fall in one of the known tractable classes. This problem is also important from a theoretical perspective, as it helps to clarify the boundary between tractability and intractability in a wide range of combinatorial search problems.

This problem was completely solved for the important special case of Boolean constraint satisfaction problems by Schaefer in 1978 [31]. Schaefer established that for Boolean constraint satisfaction problems (which he called "Generalised Satisfiability Problems") there are exactly six different families of tractable constraints, and any problem involving constraints not contained in one of these six families is NP-complete. This important result is known as Schaefer's Dichotomy Theorem for Boolean relations.

Schaefer [31] raised the question of how this result could be generalised to larger domains (that is, sets with more than 2 elements). Several approaches tackling this problem have appeared recently. In particular, Feder and Vardi [9] used techniques from logic programming and group theory to identify three broad families of tractable constraints which include all of Schaefer's six classes. In a series of papers Jeavons and co-authors have characterised a number of tractable constraint classes over finite domains using invariance properties of relations [15, 16, 17, 19].

In [18] and then in [5, 4] it have been shown that, for any subclass of CSP over a finite set of values, there is a finite algebra such that the complexity of the subclass depends only on the corresponding algebra. The first aim of this paper is to develop the algebraic approach to study CSP, and show how the complexity of a problem class can be determined by properties of a certain variety; the second aim is to attempt to represent the algebraic point of view on this subject rather than that of complexity theory. Therefore, we both prove theorems and survey previous results, often in more general form.

The paper is organised as follows. In Section 2 we introduce the constraint satisfaction problem, define some connected notions, and provide some examples. In Section 3 we, first, recall the technique developed in [18, 5, 4], second, show how this technique can be improved for a more general situation, and third, describe the connection between the complexity of subclasses of CSP and properties of varieties generated by finite algebras. Section 4 contains a short survey of algebras which give rise to tractable and NP-complete problem classes. There we also pose a conjecture on a possible characterisation of algebras corresponding to tractable subclasses of CSP. In Section 5 we exhibit an algorithm which, under the assumption of the conjecture from Section 4, recognises tractable algebras and sets of relations.

2 Constraint Satisfaction Problem

2.1 Definition and examples

The 'constraint satisfaction problem' was introduced by Montanari in 1974 [25] and has been widely studied [8, 9, 24, 21, 22, 23, 35].

For a class of sets, $\mathcal{A} = \{A_i \mid i \in I\}$, a subset ϱ of $A_{i_1} \times \ldots \times A_{i_k}$ together with the list $(i_1, \ldots, i_k)$ is called a k-ary *relation over* $\mathcal{A}$ with *signature* $(i_1, \ldots, i_k)$.

Definition 2.1 *The* constraint satisfaction problem (CSP) *is the combinatorial decision problem with*

INSTANCE: *a quadruple* $(V, \mathcal{A}, \delta, \mathcal{C})$ *where*

- V *is a set of* variables*;*
- $\mathcal{A} = \{A_i \mid i \in I\}$ *is a collection of sets [*domains*];*
- δ *is a mapping from* V *to* I*, called the* domain function*;*
- $\mathcal{C}$ *is a finite set of* constraints.
- *Each constraint* $C \in \mathcal{C}$ *is a pair* $\langle s, \varrho \rangle$*, where*
  - $s = (v_1, \ldots, v_{m_C})$ *is a tuple of variables of length* m_C*, called the* constraint scope*;*
  - ϱ *is an* m_C*-ary relation over* $\mathcal{A}$ *with signature* $(\delta(v_1), \ldots, \delta(v_{m_C}))$*, called the* constraint relation.

QUESTION: *does there exist 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* $\langle s, \varrho \rangle \in \mathcal{C}$*, with* $s = (v_1, \ldots, v_m)$*, the tuple* $(\varphi(v_1), \ldots, \varphi(v_m))$ *belongs to* ϱ*?*

The size of a problem instance is the length of the encoding of all tuples in all constraints.

Often all the variables can be supposed to have a common set of values. To distinguish this important particular case, we will refer to the general case as the *multi-sorted* CSP, and to the case with a common domain as the *one-sorted* CSP. A one-sorted problem instance is then written as $(V, A, \mathcal{C})$ where A is the common domain and V, $\mathcal{C}$ are as before.

The Constraint Satisfaction Problem provides a framework for a wide range of problems from logic, graph theory, linear algebra, planning, database theory, etc.

Example 2.2 An instance of the standard propositional SATISFIABILITY problem [10, 28] is specified by giving a formula in conjunctive normal form, that is, a conjunction of clauses, and asking whether there are values for the variables which make the formula true.

Suppose that $\Phi = X_1 \wedge \cdots \wedge X_n$ is such a formula, where the X_i are clauses. The satisfiability question for Φ can be expressed as the instance $(V, \{0,1\}, \mathcal{C})$ of the CSP, where V is the set of all variables appearing in the clauses X_i, and $\mathcal{C}$ is the set of constraints $\{\langle s_1, \varrho_1\rangle, \ldots, \langle s_n, \varrho_n\rangle\}$, where each constraint $\langle s_k, \varrho_k\rangle$ is constructed as follows:

- $s_k = (x_1^k, \ldots, x_{m_k}^k)$ where $x_1^k, \ldots, x_{m_k}^k$ are the variables appearing in the clause X_k;
- $\varrho_k = \{0,1\}^{m_k} \setminus \{(a_1, \ldots, a_{m_k})\}$ where $a_i = 1$ if x_i^k is negated in X_k and $a_i = 0$ otherwise (i.e., ϱ_k consists of all m_k-tuples that make X_k true).

The solutions of this CSP instance are exactly the assignments which make the formula Φ true. Hence, any instance of SATISFIABILITY can be expressed as an instance of the CSP. □

Example 2.3 An instance of GRAPH UNREACHABILITY consists of a graph $G = (V, E)$ and a pair of vertices, $v, w \in V$. The question is whether there is no path in G from v to w. This can be expressed as the CSP instance $(V, \{0,1\}, \mathcal{C})$ where

$$\mathcal{C} = \{\langle e, \{(0,0),(1,1)\}\rangle | e \in E\} \cup \{\langle (v), \{(0)\}\rangle, \langle (w), \{(1)\}\rangle\}.$$

□

Example 2.4 A system of linear equations over a field F can be expressed as the CSP instance $(V, F, \mathcal{C})$ where V is the set of variables of the system, and each constraint $\langle s, \varrho \rangle$ from $\mathcal{C}$ corresponds to an equation. Then s is the set of variables appearing in the equation, and ϱ is the set of solutions of the equation, that is, a hyperplane. $\square$

Example 2.5 A graph $G = (V, E)$ is said to be *interval* if its vertices can be represented by intervals of the real line such that two vertices are adjacent if and only if intervals which represent them have a common point. The question whether a given graph G is interval is expressed by the CSP instance $(V, J(\mathbb{R}), \{\langle \{v, w\}, \varrho \rangle \mid (v, w) \in E\})$ where $J(\mathbb{R})$ is the set of real intervals, and ϱ is defined to be the binary relation

$$\{((a, b), (c, d)) \mid (a, b),\ (c, d) \text{ have a common point}\}$$

$\square$

Temporal reasoning also uses the analogous but more complicated formalism known as *Allen's Interval algebra.* Allen's Interval algebra consists of 2^{13} disjunctions of subsets of 13 'primitive' basic relations : "before", "meets", "during", "overlaps", "starts", "finishes", their converses, and the 'trivial' basic relation "equal" on the set $J(\mathbb{R})$.

All the examples above are essentially one-sorted. However, sometimes the expressive power of one-sorted CSPs is not enough.

Example 2.6 Constraint satisfaction problems in which each variable has a distinct set of possible values frequently arise in the study of databases, although a rather different vocabulary is normally used, as in the following definition.

Definition 2.7 *A* relational database *is a finite collection of* tables*. A table consists of a* scheme *and an* instance*:*

A scheme is a finite set of attributes*, where each attribute has an associated set of possible values, referred to as* domain*.*

An instance is a finite set of rows*, where each row is a mapping that associates with each attribute of the scheme a value in its domain.*

As is easily seen, an instance of a constraint satisfaction problem corresponds to a relational database by a simple translation of terms: 'attributes' have to be replaced with 'variables', 'tables' with 'constraints', 'domains' with 'sets of values', 'scheme' with 'constraint scope', 'instance' with 'constraint relation', and 'rows' with 'tuples'. The saisfiability question of a constraint satisfaction problem corresponds to a *conjunctive query* to the database [36, 20]. □

The list of examples can easily be extended. The reader is encouraged to translate his own favourite combinatorial problem to the CSP language, although sometimes this is a nontrivial exercise[1].

2.2 Restricted classes of CSP

The Constraint Satisfaction Problem is NP-hard in general, as can be seen from the examples above. However, some restrictions on the form of constraints may ensure tractability of a problem class. There are two natural ways to restrict the CSP: first, restrictions on the form of the constraint relations; and second, restrictions on the way how the constraint scopes may interact. In this paper we explore the first way; some results and references concerning the second direction can be found in [11].

Let Γ be a set of relations over a class of sets, $\mathcal{A} = \{A_i \mid i \in I\}$.

Definition 2.8 *For a given set of relations* Γ *over* $\mathcal{A}$, $\mathrm{CSP}(\Gamma)$ *is defined to be the decision problem with*

INSTANCE: *An instance,* $P = (V; \mathcal{B}; \delta; \mathcal{C})$, *of a multi-sorted constraint satisfaction problem, in which*

- $\mathcal{B} \subseteq \mathcal{A}$;
- *for each constraint* $\langle s, \varrho \rangle \in \mathcal{C}$, *the relation* ϱ *is an element of* Γ.

QUESTION: *Does* P *have a solution?*

The set Γ is said to be *tractable* if, for each finite subset $\Gamma' \subseteq \Gamma$, the class $\mathrm{CSP}(\Gamma')$ is tractable, Γ is said to be *NP-complete* if $\mathrm{CSP}(\Gamma')$ is NP-complete for some finite subset $\Gamma' \subseteq \Gamma$. Some stronger versions of tractability will be discussed in the next section.

Many well-known combinatorial problems can be represented in the form $\mathrm{CSP}(\Gamma)$.

[1] Of course, we are seeking for simple and natural reductions only, at least for relatively simple and natural. Since acting of a nondeterministic Turing machine which stops in polynomial time can be expressed by an instance of SATISFIABILITY, any problem from NP is expressible in terms of any NP-complete problem. However, it is very unlikely, that for example, Gaussian elimination can be naturally expressed in terms of graph theory.

Example 2.9 An instance of GRAPH q-COLORABILITY consists of a graph G. The question is whether the vertices of G can be labelled with q colours so that adjacent vertices are assigned different colours.

This problem corresponds to CSP($\{\neq_A\}$) where A is a q-element set (of colours) and

$$\neq_A \;= \{(a,b) \in A^2 \mid a \neq b\}$$

is the disequality relation on A. □

Example 2.10 The NOT-ALL-EQUAL SATISFIABILITY (NAE) problem [10, 28] is a restricted version of the standard SATISFIABILITY problem which remains NP-complete. This problem corresponds to the problem CSP($\{N\}$) where N is the following ternary relation on $\{0,1\}$:

$$N = \{(a,b,c) \mid \{a,b,c\} = \{0,1\}\}.$$

□

While the examples above give NP-complete classes, the next problem class is tractable in spite of the infinite family of relations involved.

Example 2.11 The family of all linear systems over a field F corresponds to the class CSP(H) where H is the class of all hyperplanes of finite-dimensional F-vector spaces. □

3 Algebraic structure of the CSP

Throughout the rest of the paper we assume the domains to be finite.

3.1 From a set of relations to finite algebras

In this section we explore the following question: how a tractable set of relations can be extended such that the obtained set remains tractable. First, we deal with the one-sorted CSP.

We make use of the standard correspondence between relations and logical predicates: an n-ary relation ϱ consists of all tuples on which the corresponding n-ary predicate P_ϱ is true. Recall that an existential first order formula is said to be *primitive positive (pp-)* if its quantifier-free part is a conjunction of predicates. These formulas can be used as operations that allow one to derive new relations from given ones. A set of relations on a set A closed with respect to all pp-formulas is called a *relational clone*. The smallest relational clone, $\langle\Gamma\rangle$, containing a set of relations Γ is said to be *generated by* Γ.

Theorem 3.1 ([13]) *For any set of relations Γ over a finite set, and any finite set of relations $\Gamma' \subseteq \langle\Gamma\rangle$, there is a polynomial time reduction from* $\mathrm{CSP}(\Gamma')$ *to* $\mathrm{CSP}(\Gamma)$.

Corollary 3.2 *If a set of relations Γ is tractable, then the relational clone $\langle\Gamma\rangle$ is also tractable.*

An operation f on a set A *preserves* an n-ary relation ϱ (or ϱ is *invariant* under f) if ϱ is a subuniverse of the algebra $(A; f)^n$. For a given set of operations, C, the set of all relations invariant under operations from C is denoted by $\mathsf{Inv}\ C$. Conversely, for a set of relations, Γ, the set of all operations preserving relations from Γ is denoted by $\mathsf{Pol}\ \Gamma$. The universe of a finite algebra $\mathbb{A}$ will be denoted by A.

It is well known [29, 33] that, for any finite algebra $\mathbb{A} = (A; F)$, the set $\mathsf{Inv}\ F$, that is, the set of subuniverses of finite direct powers of $\mathbb{A}$, forms a relational clone. Moreover, for any relational clone Γ, there is a finite algebra $\mathbb{A} = (A; F)$ such that $\Gamma = \mathsf{Inv}\ F$. In fact, F can be chosen to be $\mathsf{Pol}\ \Gamma$.

Thus every finite algebra $\mathbb{A}$ determines a subclass $\mathrm{CSP}(\mathsf{Inv}\ F)$ of the constraint satisfaction problem. We say that the algebra $\mathbb{A}$ is tractable [NP-complete] if the set $\mathsf{Inv}\ F$ is tractable [NP-complete]. Conversely, for any problem class of the form $\mathrm{CSP}(\Gamma)$, there is a corresponding finite algebra, and the tractability [NP-completeness] of the problem class depends only on this algebra.

Next we do the analogous procedure for the multi-sorted CSP. However, we avoid the direct use of relational clones, and instead use the results on one-sorted problems.

Let $\mathfrak{A}$ be a class of similar finite algebras. As usual, the class of all subalgebras of algebras from $\mathfrak{A}$ is denoted by $\mathsf{S}\mathfrak{A}$, the class of all finite direct products of algebras from $\mathfrak{A}$ by $\mathsf{P}_{\mathrm{fin}}\mathfrak{A}$. Then the problem class $\mathrm{CSP}(\mathfrak{A})$ is defined to be $\mathrm{CSP}(\Gamma)$ where Γ is the class of universes of algebras from $\mathsf{SP}_{\mathrm{fin}}\mathfrak{A}$. If $\mathfrak{A}$ is finite, say $\mathfrak{A} = \{\mathbb{A}_1, \dots, \mathbb{A}_n\}$, we shall write $\mathrm{CSP}(\mathbb{A}_1, \dots, \mathbb{A}_n)$ instead of $\mathrm{CSP}(\mathfrak{A})$. The class $\mathfrak{A}$ is said to be tractable [NP-complete] if Γ is tractable [NP-complete]. If $\mathfrak{A} = \{\mathbb{A}_1, \dots, \mathbb{A}_n\}$ is tractable, then $\mathbb{A}_1, \dots, \mathbb{A}_n$ are said to be *mutually tractable*.

Proposition 3.3 *A class of similar finite algebras, $\mathfrak{A}$, is tractable if and only if every $\mathbb{A}_1, \dots, \mathbb{A}_n \in \mathfrak{A}$ are mutually tractable.*

In a practical situation we are usually given not algebras, but a set of relations, which is mostly finite. In this case one can find an appropriate algebraic structure on the domains in the following way.

Let $A_1, \dots, A_n$ be finite sets, and Γ a set of relations over $\{A_1, \dots, A_n\}$. Denote by A the Cartesian product of $A_1, \dots, A_n$ and by φ_i, $i \in \{1, \dots, n\}$, the projection mapping of A onto A_i. Then, for $\varrho \in \Gamma$ of signature $(i_1, \dots, i_k)$, set

$$\chi(\varrho) = \{(a_1, \dots, a_q) \in A^q \mid (\varphi_{i_1}(a_1), \dots, \varphi_{i_q}(a_q)) \in \varrho\}.$$

Further, let θ_i denote the kernel of φ_i, $\Phi = \{\theta_1, \dots, \theta_n\}$, and $\Gamma' = \{\chi(\varrho) \mid \varrho \in \Gamma\} \cup \Phi$. Finally, set $\mathbb{A} = (A; \mathsf{Pol}\,\Gamma')$, and let $\mathbb{A}_i$ denote the homomorphic image of $\mathbb{A}$ under φ_i, $i \in \{1, \dots, n\}$. Obviously, $\mathbb{A} = \mathbb{A}_1 \times \ldots \times \mathbb{A}_n$. What we have to prove is that $\mathbb{A}_1, \dots, \mathbb{A}_n$ give an appropriate algebraic structure for Γ.

Theorem 3.4 *Let $A_1, \dots, A_n$ be finite sets, Γ a set of relations over $\{A_1, \dots, A_n\}$, and $\mathbb{A}, \mathbb{A}_1, \dots, \mathbb{A}_n$ defined as above. Then the following are equivalent:*

(a) $\mathrm{CSP}(\Gamma)$ *is tractable;*

(b) $\mathrm{CSP}(\mathbb{A}_1, \dots, \mathbb{A}_n)$ *is tractable;*

(c) $\mathbb{A}$ *is tractable.*

Proof. We need an auxiliary statement.

Claim. For any finite set of relations Δ over $\{A_1, \dots, A_n\}$, $\mathrm{CSP}(\Delta)$ is polynomial-time equivalent to $\mathrm{CSP}(\chi(\Delta))$.

Let $P = (V; \{A_1, \dots, A_n\}; \delta; \mathcal{C})$ be an instance of $\mathrm{CSP}(\Delta)$ where $\mathcal{C} = \{C_1, C_2, \dots, C_q\}$ and $C_i = \langle s_i, \varrho_i \rangle$.

Consider the instance $P' = (V; A; \mathcal{C}')$ where $A = A_1 \times A_2 \times \cdots \times A_n$, $\mathcal{C}' = \{C'_1, C'_2, \dots, C'_q\}$ and $C'_i = \langle s_i, \chi(\varrho_i) \rangle$. Note that every solution to P can be used to construct a solution to P' by extending the value assigned to each variable (arbitrarily) to a tuple over A. Conversely, every solution to P' can be used to obtain a solution to P by projecting the tuple over A assigned to each variable v onto $A_{\delta(v)}$. Hence we have a reduction from $\mathrm{CSP}(\Delta)$ to $\mathrm{CSP}(\chi(\Delta))$.

Furthermore, every tuple in every constraint relation of P is replaced by a fixed number of tuples (depending only on the cardinalities of the sets $A_1, \dots, A_n$), to obtain the corresponding constraint relation in P'. Hence, the reduction can be carried out in linear time.

The same arguments can be applied in the reverse direction to obtain a polynomial-time reduction from $\mathrm{CSP}(\chi(\Delta))$ to $\mathrm{CSP}(\Delta)$.

Hence the two problems are polynomial-time equivalent.

We exhibit the following implications: (a)$\Longrightarrow$(c)$\Longrightarrow$(b)$\Longrightarrow$(a).

(a)$\Longrightarrow$(c). By Theorem 3.1, it is enough to prove that every problem instance from CSP(Γ') can be reduced to an instance from CSP(Γ). Notice first that $\theta_i = \chi(\underline{0}_i)$, for each $i \in \{1, \ldots, n\}$, where $\underline{0}_i$ is the equality relation on A_i. So, the required reduction will be done in two steps: the first one, CSP(Γ') to CSP($\Gamma \cup \{\underline{0}_1, \ldots, \underline{0}_n\}$) is done in the Claim; for the second step, CSP($\Gamma \cup \{\underline{0}_1, \ldots, \underline{0}_n\}$ to CSP(Γ), we just slightly change the proof of Proposition 4.1 from [13].

Let $P = (V, \{A_1, \ldots, A_n\}; \delta; \mathcal{C})$ be any instance of CSP($\Gamma \cup \{\underline{0}_1, \ldots, \underline{0}_n\}$). We will modify P to construct an instance P' of CSP(Γ), as follows. For each constraint $C_i \in \mathcal{C}$, if ϱ_i is $\underline{0}_j$ then

– if $s_i = (v_1, v_2)$ then replace all occurrences of v_1 in the constraint scopes of constraints from $\mathcal{C}$ with v_2;

– remove the constraint C_i.

Clearly, this construction can be carried out in polynomial time, and P' has a solution if and only if P has a solution.

(c)$\Longrightarrow$(b). Follows straightforwardly from the Claim, noticing that, for any $\Delta \subseteq \mathsf{SP}_{\text{fin}}(\mathbb{A}_1, \ldots, \mathbb{A}_n)$, $\chi(\Delta) \subseteq \mathsf{SP}_{\text{fin}}(\mathbb{A})$.

(b)$\Longrightarrow$(a). If $\varrho \in \Gamma$, then $\chi(\varrho) \in \mathsf{SP}_{\text{fin}}(\mathbb{A})$, and therefore,

$$\varrho = \{(\varphi_{i_1}(a_1), \ldots, \varphi_{i_m}(a_m)) \mid (a_1, \ldots, a_m) \in \chi(\varrho)\}$$

is a subalgebra of $\mathbb{A}_{i_1} \times \ldots \times \mathbb{A}_{i_m}$. So, $\Gamma \subseteq \mathsf{SP}_{\text{fin}}(\mathbb{A}_1, \ldots, \mathbb{A}_n)$. $\square$

3.2 From an algebra to a variety

In this subsection we study how the standard algebraic operators are connected with the complexity of algebras.

Theorem 3.5 *For any set of finite similar algebras, $\mathbb{A}_1, \ldots, \mathbb{A}_n$, the following are equivalent:*

1. CSP($\mathbb{A}_1, \ldots, \mathbb{A}_n$) *is tractable [NP-complete];*

2. CSP($\mathbb{B}, \mathbb{A}_1, \ldots, \mathbb{A}_n$) *where $\mathbb{B}$ is a subalgebra of $\mathbb{A}_1$ is tractable [NP-complete];*

3. CSP($\mathbb{C}, \mathbb{A}_1, \ldots, \mathbb{A}_n$) *where $\mathbb{C}$ is a homomorphic image of $\mathbb{A}_1$ is tractable [NP-complete];*

4. $\mathrm{CSP}(\mathbb{A}_1 \times \mathbb{A}_2, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ *is tractable [NP-complete].*

Proof. Note first that problems (2), (3), and (4) include problem (1), so there is a trivial polynomial time reduction from (1) to (2), (3) and (4). We show that, for any finite $\Gamma \subseteq \mathsf{SP}_{\mathrm{fin}}(\mathbb{B}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ ($\mathsf{SP}_{\mathrm{fin}}(\mathbb{C}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$, $\mathsf{SP}_{\mathrm{fin}}(\mathbb{A}_1 \times \mathbb{A}_2, \mathbb{A}_1, \ldots, \mathbb{A}_n)$), there is a polynomial time reduction from $\mathrm{CSP}(\Gamma)$ to $\mathrm{CSP}(\mathbb{A}_1, \ldots, \mathbb{A}_n)$.

(1) $\Rightarrow$ (2) Let $\mathbb{B}$ be a subalgebra of $\mathbb{A}_1$ and let B be the universe of $\mathbb{B}$. Every instance of $\mathrm{CSP}(\mathbb{B}, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ can be mapped to an equivalent instance of $\mathrm{CSP}(\mathbb{A}_1, \ldots, \mathbb{A}_n)$ by simply replacing all occurrences of B with A_1. Hence there is a polynomial-time reduction from problem (2) to problem (1).

(1) $\Rightarrow$ (3) Let φ be a homomorphism from $\mathbb{A}_1$ onto $\mathbb{A}_0 = \mathbb{C}$, and let A_0 be the universe of $\mathbb{A}_0$. For any relation ϱ over $\{A_0, A_1, \ldots, A_n\}$ with the signature $(i_1, \ldots, i_k)$, we set $\varphi^{-1}(\varrho) = \{\mathbf{b} = (b_1, \ldots, b_k) \mid$ there is $\mathbf{a} = (a_1, \ldots, a_k) \in \varrho$ such that $b_j = a_j$ if $i_j \neq 0$ and $b_j \in A_1, \varphi(b_j) = a_j$ otherwise$\}$. Obviously, $\varphi^{-1}(\varrho)$ is a relation over $\{A_1, \ldots, A_n\}$ of the same arity as ϱ. Moreover, it can be straightforwardly checked that if ϱ is a subuniverse of a direct product of some algebras from $\mathbb{A}_0, \mathbb{A}_1, \ldots, \mathbb{A}_n$, then $\varphi^{-1}(\varrho)$ is a subuniverse of a direct product of some algebras from $\mathbb{A}_1, \ldots, \mathbb{A}_n$.

Now take a finite set $\Gamma \subseteq \mathsf{SP}_{\mathrm{fin}}(\mathbb{A}_0, \mathbb{A}_1, \ldots, \mathbb{A}_n)$, then $\Gamma' = \{\varphi^{-1}(\varrho) \mid \varrho \in \Gamma\}$ is a finite subset of $\mathsf{SP}_{\mathrm{fin}}(\mathbb{A}_1, \ldots, \mathbb{A}_n)$. Take also a problem instance $P = (V; \{A_0, A_1, \ldots, A_n\}; \delta; \mathcal{C})$ from $\mathrm{CSP}(\mathbb{A}_0, \mathbb{A}_1, \ldots, \mathbb{A}_n)$ such that all the constraint relations belong to Γ (i.e. $P \in \mathrm{CSP}(\Gamma)$). We now show how to reduce P to an instance from $\mathrm{CSP}(\Gamma')$ in linear time.

The corresponding instance is $P' = (V; \{A_1, \ldots, A_n\}; \delta'; \mathcal{C}')$ where $\delta'(v) = \delta(v)$ if $\delta(v) \neq 0$ and $\delta'(v) = 1$ otherwise, and $C' = \langle s, \varphi^{-1}(\varrho)\rangle$ for each $C = \langle s, \varrho\rangle \in \mathcal{C}$.

If ψ is a solution of P' then ψ' where

$$\psi'(v) = \begin{cases} \psi(v), & \text{if } \delta'(v) \neq 0; \\ \varphi\psi(v), & \text{if } \delta'(v) = 0 \end{cases}$$

is a solution of P. Conversely, if ξ is a solution of P then any function $\psi : V \to A_1 \cup \ldots \cup A_n$ such that $\psi'(v) = \xi(v)$, for any $v \in V$, where ψ' is constructed as above, is a solution of P'.

(1) $\Rightarrow$ (4) For a given finite set $\Gamma \subseteq \mathsf{SP}_{\mathrm{fin}}(\mathbb{A}_1 \times \mathbb{A}_2, \mathbb{A}_1, \ldots, \mathbb{A}_n)$, we are going to show how to reduce an instance from $\mathrm{CSP}(\Gamma)$ to an instance from $\mathrm{CSP}(\Gamma')$ where Γ' is a certain finite subset of $\mathsf{SP}_{\mathrm{fin}}(\mathbb{A}_1, \ldots, \mathbb{A}_n)$, in linear time.

First we have to transform a relation over $\{\mathbb{A}_1 \times \mathbb{A}_2, \mathbb{A}_1, \ldots, \mathbb{A}_n\}$ to a relation over $\{\mathbb{A}_1, \ldots, \mathbb{A}_n\}$. Let $\mathbf{a}$ be an element of $D_1 \times \ldots \times D_k$ where

$D_1, \dots, D_k \in \{A_1 \times A_2, A_1, \dots, A_n\}$ and $D_i = A_1 \times A_2$. Then one may eliminate this occurrence of $A_1 \times A_2$ by the following transformation

$$\mathbf{a} = \left(\begin{array}{c} a_1 \\ \vdots \\ (a_i^1, a_i^2) \\ \vdots \\ a_k \end{array} \right) \quad \mapsto \quad \left(\begin{array}{c} a_1 \\ \vdots \\ a_i^1 \\ a_i^2 \\ \vdots \\ a_k \end{array} \right).$$

Repeating this transformation for each occurrence of $A_1 \times A_2$ gives a tuple $\mathbf{a}'$ over $A_1, \dots, A_n$. It is clear that the tuple $\mathbf{a}'$ is $k+s$-ary where s is the number of occurrences of $A_1 \times A_2$ among $D_1, \dots, D_k$. Then for a relation $\varrho \subseteq D_1 \times \dots \times D_k$, we set $\varrho' = \{\mathbf{a}' \mid \mathbf{a} \in \varrho\}$ and, for a set of relations Γ on $A_1 \times A_2, A_1, \dots, A_n$, Γ' denotes the set $\{\varrho' \mid \varrho \in \Gamma\}$. It can be straightforwardly checked that if ϱ is a subuniverse of a direct product of some algebras from $\mathbb{A}_1 \times \mathbb{A}_2, \mathbb{A}_1, \dots, \mathbb{A}_n$, then ϱ' is a subuniverse of a direct product of certain algebras from $\mathbb{A}_1, \dots, \mathbb{A}_n$.

We define the required reduction as follows. Let $\mathbb{A}_0 = \mathbb{A}_1 \times \mathbb{A}_2$, and $P = (V; \{A_0, A_1, \dots, A_n\}; \delta; \mathcal{C})$ be a problem instance from CSP(Γ). Then for each variable $v \in V$ such that $\delta(v) = 0$, we do

- replace v with two new variables v^1, v^2;
- set $\delta'(v^1) = 1$, $\delta'(v^2) = 2$;
- for every $C = \langle s, \varrho \rangle \in \mathcal{C}$ such that $v \in s$, set $s' = \langle s - \{v\} \rangle \cup \{v^1, v^2\}$ (inserting v_1, v_2 in the right places).

Finally, replace every constraint relation ϱ with ϱ', and for each $v \in V$ with $\delta(v) \neq 0$, set $\delta'(v) = \delta(v)$. Notice that ϱ' is a subset of $\prod_{v \in s'} A_{\delta'(v)}$. The obtained problem instance we denote by $P' = (V'; \{A_1, \dots, A_n\}; \delta'; \mathcal{C}')$.

Let ψ be a solution of P. Define the function $\psi' \colon V' \to A_1 \cup \dots \cup A_n$ as follows: $\psi'(v) = \psi(v)$ if $v \in V$ (and therefore $\mathbb{A}_{\delta(v)} \neq \mathbb{A}_1 \times \mathbb{A}_2$); and $\psi'(v^1) = a^1, \psi'(v^2) = a^2$ if v^1, v^2 was obtained from v and $\psi(v) = (a^1, a^2) \in \mathbb{A}_1 \times \mathbb{A}_2$. It can be easily checked that ψ' is a solution to P'. Conversely, let ξ' be a solution of P'. To get a solution of P we set $\xi(w) = \xi'(w)$ whenever w is not of the form v^1, v^2, for some $v \in V$, and $\xi(w) = (\xi'(w^1), \xi'(w^2))$ otherwise. $\Box$

Making use of Proposition 3.3, we may apply this theorem to infinite classes of algebras. For a class of algebras, $\mathfrak{A}$, by $\mathsf{pvar}\mathfrak{A}$, $\mathsf{var}\mathfrak{A}$ we denote the pseudo-variety and the variety generated by $\mathfrak{A}$, and by $\mathfrak{A}_{\mathrm{fin}}$ the class of finite algebras from $\mathfrak{A}$.

Corollary 3.6 *A class of finite algebras, $\mathfrak{A}$, is tractable if and only if* $\mathsf{pvar}\mathfrak{A}$ *is tractable.*

As is well known, if $\mathfrak{A}$ is a finite class of finite algebras, then the pseudo-variety generated by $\mathfrak{A}$ equals the class of finite algebras from the variety generated by $\mathfrak{A}$.

Corollary 3.7 *A finite class of finite algebras, $\mathfrak{A}$, is tractable if and only if* $(\mathsf{var}\mathfrak{A})_{\mathrm{fin}}$ *is tractable.*

Corollary 3.8 *If $\mathfrak{A}$ is a finite class of finite algebras, and* $\mathsf{var}\mathfrak{A}$ *contains a finite NP-complete algebra, then $\mathfrak{A}$ is NP-complete.*

Thus, the tractability of an algebra is a property which is determined by identities.

We complete this subsection by showing that, in fact, a classification of algebras according to the complexity of corresponding problem classes can be derived from such a classification in the case of idempotent algebras.

Let $\mathbb{A} = (A; F)$ be an algebra, and $U \subseteq A$. The algebra $\mathbb{A}|_U = (U, F')$ where F' consists of the restrictions on U of all term operations of $\mathbb{A}$ preserving U is called the *term-induced* algebra.

Proposition 3.9 ([4]) *Let $\mathbb{A}$ be a finite algebra, f a unary idempotent term, and $U = f(A)$. Then $\mathbb{A}$ is tractable if and only if $\mathbb{A}|_U$ is tractable.*

By choosing an idempotent term with a minimal range, we may restrict ourselves to considering only surjective algebras. Recall that the *full idempotent reduct* of an algebra $\mathbb{A} = (A; F)$ is the algebra $\mathsf{Id}(\mathbb{A}) = (A, F')$ where F' consists of all idempotent term operations of $\mathbb{A}$.

Proposition 3.10 ([4]) *A surjective finite algebra $\mathbb{A}$ is tractable if and only if its full idempotent reduct is tractable.*

One can easily repeat this reduction to idempotent algebras when considering the multi-sorted CSP over a finite class of algebras. Let $\mathbb{A}_1, \ldots, \mathbb{A}_n$ be similar finite algebras, and $\mathbb{A} = \mathbb{A}_1 \times \ldots \times \mathbb{A}_n$. For a term f by f^A, f^{A_i} we denote the corresponding term operations of $\mathbb{A}$, $\mathbb{A}_i$.

Lemma 3.11 *(a) If f^A is a unary term of $\mathbb{A}$ with a minimal range, then $f^{A_1},\dots,f^{A_n}$ are terms with minimal ranges for $\mathbb{A}_1,\dots,\mathbb{A}_n$ respectively, and $f^A(A) = f^{A_1}(A_1)\times\ldots\times f^{A_n}(A_n)$.*
(b) The full idempotent reduct $\mathsf{Id}(\mathbb{A})$ is the direct product of $\mathbb{A}'_1,\dots,\mathbb{A}'_n$ where $\mathbb{A}'_i = (A_i;F')$, $1\le i\le n$, and F' is the set of all terms f such that all $f^{A_1},\dots,f^{A_n}$ are idempotent.

Further, suppose that $f_1,\dots,f_n$ are unary terms such that $f_i^{A_i}$ has a minimal range. Then $f = (f_1\dots f_n)^s$, for a certain s, is idempotent, and f^{A_i} has a minimal range for each $i\in\{1,\dots,n\}$.

Proposition 3.12 *(a) If $\mathbb{A}_1,\dots,\mathbb{A}_n$ are mutually tractable, then so are $\mathbb{A}_1\big|_{U_1},\dots,\mathbb{A}_n\big|_{U_n}$ where $U_i = f^{A_i}(A_i)$, $1\le i\le n$.*
(b) If $\mathbb{A}_1,\dots,\mathbb{A}_n$ are surjective and mutually tractable, then $\mathbb{A}'_1 = (A_1;F'),\dots,\mathbb{A}'_n = (A_n;F')$ where F' consists of all terms g, such that all $g^{A_1},\dots,g^{A_n}$ are idempotent, are also mutually tractable.

To prove the proposition we just have to notice that $\mathbb{A} = \mathbb{A}_1\times\ldots\times\mathbb{A}_n$ is tractable if and only if $\mathbb{A}\big|_{f^A(A)}$, in part (a), or $\mathsf{Id}(\mathbb{A})$, in part (b), is tractable, and then make use of Lemma 3.11.

3.3 Three levels of tractability

In Section 2 we defined the tractability of a set of relations, Γ, as the existence, for each finite subset $\Gamma'\subseteq\Gamma$, of a polynomial-time algorithm $A_{\Gamma'}$, which solves the problem $\mathrm{CSP}(\Gamma')$. This definition arose from the tradition of complexity theory, where combinatorial problems are defined mostly via finite sets of relations. However, the tractability of an infinite set Γ does not guarantee existence of a uniform algorithm that solves arbitrary instance from $\mathrm{CSP}(\Gamma)$ in polynomial time. The tractability of the problem class $\mathrm{CSP}(\Gamma)$ is, on the one hand, a natural generalisation of tractability of the set Γ, and on the other hand, is of interest in some areas of computer science.

Definition 3.13 *A set of relations Γ over sets $A_1,\dots,A_n$ is said to be* globally tractable *if there exists a polynomial time algorithm solving* $\mathrm{CSP}(\Gamma)$. *A set of finite similar algebras $\{\mathbb{A}_1,\dots,\mathbb{A}_n\}$ is said to be* globally tractable *if $\mathsf{SP}_{\mathrm{fin}}(\mathbb{A}_1,\dots,\mathbb{A}_n)$ is globally tractable. A class of finite algebras, $\mathfrak{A}$, is said to be* globally tractable *if any finite subclass $\{\mathbb{A}_1,\dots,\mathbb{A}_n\}\subseteq\mathfrak{A}$ is globally tractable.*

The most well known example of a globally tractable set of relation is linear systems over a field. As will be seen in the next section, almost all algebras, which have been proved to be tractable, can be shown to be globally tractable. Finite groups are the only exception. However, the global tractability has been disproved for no finite group, and is very unlikely to be disproved.

Problem 3.14 *Is it true that any tractable finite algebra (finite set of finite algebras) is globally tractable?*

We get an even stronger version of tractability if we allow the set of relations to be over an infinite family of domains. The authors know no example of an established combinatorial problem formulated via so large a class of relations. So, we define only the algebraic version of this notion.

Definition 3.15 *A class of finite algebras, $\mathfrak{A}$, is said to be* uniformly tractable *if there exists a polynomial time algorithm solving the problem class* $\mathrm{CSP}(\mathsf{SP}_{\mathrm{fin}}\mathfrak{A})$.

Though this strongest version of tractability is not widely used in computer science, we strongly believe that it might be very useful in theoretical, and probably practical perspective. As before, to date there is no example of a (globally) tractable class of algebras, which is proved not to be uniformly tractable.

4 Complexity of algebras of special types

In this section we give a small survey of algebras of known complexity. Most results are stated here in a stronger form than that was proved earlier. We also prove some natural generalisations of some results.

4.1 Tractable algebras

Algebras with a particular term. It was proved in [14, 15, 17, 18, 19] that a set of relations Γ over a finite set A is tractable whenever all relations from Γ are invariant under one of the following operations: constant operation, semilattice operation, affine operation (that is the operation $x-y+z$ where $+,-$ are the operations of an Abelian group), dual discriminator operation, near-unanimity operation. By careful inspection of the proof, one may state this result in a stronger form.

Proposition 4.1 *If a variety $\mathfrak{A}$ has one of the following terms: constant, semilattice, affine, dual discriminator, near-unanimity, then the class of finite algebras from $\mathfrak{A}$ is uniformly tractable.*

T.Feder and M.Vardi proved in [9] that an arbitrary finite group, and even an arbitrary finite algebra with a Mal'tsev group term, is tractable.

Proposition 4.2 *If a finite algebra $\mathbb{A}$ has a ternary term f which can be represented in the form $xy^{-1}z$ where $\cdot, ^{-1}$ are the operations of a certain group, then $\mathbb{A}$ is tractable.*

Para-primal algebras. Clearly, the more terms has an algebra or a variety, the simpler the corresponding problem class. From this point of view algebras close to primal algebras have to have an easy corresponding problem class. The widest known primal-like type of algebras is para-primal algebras.

Let A be a finite set, ϱ an n-ary relation on A, and $\varnothing \neq I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$. By the *projection* of ϱ onto I we mean the relation $\varrho|_I = \{(a_{i_1}, \dots, a_{i_k}) \mid (a_1, \dots, a_n) \in \varrho\}$. The set I is said to be *ϱ-reduced* if it is minimal with the property that the natural mapping $\varrho \mapsto \varrho|_I$ is one-to-one. An algebra $\mathbb{A}$ is called *para-primal* if it is finite, and for every $n \in \mathbb{N}$, every subuniverse ϱ of $\mathbb{A}^n$, and every ϱ-reduced set I, we have $\varrho|_I = \prod_{i\in I} \varrho|_{\{i\}}$.

The following algebras are examples of para-primal algebras: quasi-primal algebras, Abelian groups of prime order, algebras with the switching term

$$s(a,b,c) = \begin{cases} a & \text{if } b = c, \\ b & \text{if } a = c, \\ c & \text{otherwise.} \end{cases}$$

Para-primal algebras were proved to be tractable in [6]. As a matter of fact, [6] proves a stronger result.

Proposition 4.3 ([6]) *An arbitrary class of similar finite para-primal algebras is uniformly tractable.*

Algebras which guarantee special properties of a problem instance. Let $P = (V; \mathcal{A}; \delta; \mathcal{C})$ be a problem instance. For any constraint $\langle s, \varrho \rangle \in \mathcal{C}$, and any subsequence s' of the constraint scope s, that is $s' = (v_{i_1}, \dots, v_{i_k}) \subseteq (v_1, \dots, v_m) = s$, we denote $\varrho|_{s'}$ the relation $\varrho|_{\{i_1,\dots,i_k\}}$. A *partial solution* of P on $V' \subseteq V$ is a function $\varphi\colon V' \to \bigcup_{v\in V'} A_{\delta(v)}$ such that, for any $C = \langle s, \varrho \rangle \in \mathcal{C}$, $\varphi(s \cap V') \in \varrho|_{s\cap V'}$. Let us denote the set of all partial solutions on V' by $S_{V'}$; the set $S_{V'}$ can be viewed as a $|V'|$-ary relation. A problem instance P is said to be *k-minimal* if, for any at most k-element subset V' of V, and any $C = \langle s, \varrho \rangle \in \mathcal{C}$, we have $\varrho\big|_{s \cap V'} = S_{V'}\big|_{s \cap V'}$.

Any problem instance P can be transformed to an equivalent k-minimal instance P'. To do this we have just to repeat the following procedure until the obtained problem instance coincides with the previous one: for each k-element set $V' \subseteq V$, calculate the relation $S_{V'}$, and then for each constraint $C = \langle s, \varrho \rangle \in \mathcal{C}$, throw out from ϱ all tuples $\mathbf{a}$ such that $\mathbf{a}\big|_{s \cap V'} \notin S_{V'}\big|_{s \cap V'}$. The obtained instance P' is k-minimal and the time complexity of this procedure is $O(m^2 n^k)$ where n is the number of variables and m is the total number of tuples in the constraint relations. The obtained instance is referred to as *associated* with P. As is easily seen the solution set of the associated instance is equal to that of the original instance.

A problem class $\mathbf{C}$ is said to be of *width*[2] k if, for any $P \in \mathbf{C}$ such that, for each k-element set of variables, V', there is a constraint whose scope contains V', the instance P has a solution whenever the k-minimal instance associated with P has no empty constraint. So, for a problem class of width k, there exists a solving algorithm of time complexity $O(m^2 n^k)$. In particular, if $\mathbf{C}$ is of the form $\mathrm{CSP}(\Gamma)$, then Γ is globally tractable.

In [6] a sufficient condition for a problem class to be of width 1 was proved. A *set function* on a set A is defined to be a mapping from the power set $\mathcal{P}(A)$ to A. A set function f preserves an n-ary relation ϱ, if for any tuples $(a_{11}, \ldots, a_{1n}), \ldots, (a_{m1}, \ldots, a_{mn}) \in \varrho$ we have $(f(\{a_{11}, \ldots, a_{m1}\}), \ldots, f(\{a_{1n}, \ldots, a_{mn}\})) \in \varrho$.

Proposition 4.4 ([6]) *Let Γ be a set of relations on a finite set A. If there is a set function preserving all relations from Γ, then $\mathrm{CSP}(\Gamma)$ is of width 1.*

A problem class with the width more than 1 is exhibited in [3]. A groupoid $\mathbb{G}$ is said to be *conservative* if, for any $a, b \in \mathbb{G}$, $ab \in \{a, b\}$.

Proposition 4.5 ([3]) *Let $\mathbb{G}$ be a finite conservative groupoid. If $\mathbb{G}$ is commutative, then $\mathrm{CSP}(\mathbb{G})$ is of width 3, but not of width 2. Otherwise $\mathbb{G}$ is NP-complete.*

Algebras with a zero-element. Proposition 4.4 allows one to prove the tractability of problem classes corresponding to algebras with certain properties. An element a of an algebra $\mathbb{A}$ is said to be a *zero-element* with respect to a term $t(x_1, \ldots, x_n)$ if t is non-unary and $t(a_1, \ldots, a_n) = a$ whenever $a \in \{a_1, \ldots, a_n\}$.

[2]There are several unequivalent notions of the width.

Proposition 4.6 *Let $\mathbb{A}$ be a finite algebra, and t a non-unary term. If, for any subalgebra $\mathbb{B}$ of $\mathbb{A}$, there exists a zero-element $z_{\mathbb{B}}$ of $\mathbb{B}$ with respect to t, then $\mathbb{A}$ is globally tractable.*

Proof. By Proposition 4.4, it is enough to prove that there exists a set function f on A preserving each relation from $\mathsf{SP}_{\text{fin}}\mathbb{A}$. Let us define $f(B)$, $\varnothing \neq B \subseteq A$ to be the zero-element of the subalgebra generated by B.

Now take n-ary $\varrho \in \mathsf{SP}_{\text{fin}}\mathbb{A}$ and $\mathbf{a}_1 = (a_{11}, \dots, a_{1n}), \dots, \mathbf{a}_m = (a_{m1}, \dots, a_{mn}) \in \varrho$. Denote the zero-element of the subalgebra generated by $a_{1i}, \dots, a_{mi}$ by z_i. Then there are m-ary terms $s_1, \dots, s_n$ such that $s_i(a_{1i}, \dots, a_{mi}) = z_i$. Set $\mathbf{c}_i = s_i(\mathbf{a}_1, \dots, \mathbf{a}_m)$, then the ith coordinate position of $\mathbf{c}_i$ is z_i. We have

$$t(\dots t(t(\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_2), \mathbf{c}_3, \dots, \mathbf{c}_3) \dots) = (z_1, \dots, z_n),$$

which means that f preserves ϱ. $\square$

Notice that every finite semilattice satisfies the conditions of Proposition 4.6.

Finite semigroups. A semigroup is called a *block-group* if none of its subsemigroups is a left- or right-zero semigroup.

Proposition 4.7 ([?]) *A finite semigroup is tractable if and only if it is a block-group. In this case it is globally tractable. Otherwise the semigroup is NP-complete.*

Proposition 4.7 can also be proved making use of Proposition 4.4. The set function which witnesses the tractability of a block-group $\mathbb{G}$ maps a set $\{a_1, \dots, a_k\} \subseteq \mathbb{G}$ to the element $(a_1^{n!} \dots a_k^{n!})^{n!}$ where $n = |\mathbb{G}|$.

4.2 NP-complete algebras and the Dichotomy Problem

Up to now only one reason for an algebra to be NP-complete is known. Suppose that a finite algebra $\mathbb{A}$ is term equivalent to a G-set. As is easily seen, the disequality relation is a subuniverse of $\mathbb{A}^2$. Moreover, if $|\mathbb{A}| = 2$, the relation N from Example 2.10 is a subuniverse of $\mathbb{A}^3$. Since GRAPH $|\mathbb{A}|$-COLORABILITY for $|\mathbb{A}| > 2$, and NAE, are NP-complete, this means that $\mathbb{A}$ is NP-complete.

Proposition 4.8 *If the pseudovariety generated by a finite algebra $\mathbb{A}$ contains an algebra term equivalent to a nontrivial G-set, then $\mathbb{A}$ is NP-complete.*

Combining Propositions 4.1,4.8 with Post's description of clones on a 2-element set [30] one may completely classify 2-element algebras with respect to complexity. The following result is slightly stronger, than that obtained by Schaefer [31], which was stated for the complexity of sets of relations.

Proposition 4.9 ([31]) *A pseudo-variety generated by a 2-element algebra $\mathbb{A}$ is uniformly tractable if $\mathbb{A}$ is not term equivalent to a G-set. Otherwise it is NP-complete.*

In [**?**], the result of [31] and Proposition 4.9 is extended to a classification of 3-element algebras with respect to complexity.

Proposition 4.10 ([?]) *An idempotent 3-element algebra $\mathbb{A}$ is globally tractable if none of its nontrivial subalgebras and homomorphic images is term equivalent to a G-set. Otherwise $\mathbb{A}$ is NP-complete.*

Another class of finite algebras for which a complete classification is known, is the class of strictly simple algebras. A finite algebra is said to be *strictly simple* if it is simple and has no subalgebras except for one-element. Strictly simple algebras are completely described in [34].

Proposition 4.11 ([5, 4]) *A strictly simple algebra $\mathbb{A}$ is tractable if and only if it is term equivalent to no G-set.*

These propositions and the examples above lead to a natural conjecture. An algebra whose basic operations are trivial is called a *set*.

Conjecture 4.12 *An idempotent finite algebra is (globally) tractable if and only if it generates a variety which contains no finite set.*

4.3 Tractability and omitting types

Recall that tame congruence theory allows one to assign to each prime quotient of the congruence lattice $\mathsf{Con}\ \mathbb{A}$ of a finite algebra $\mathbb{A}$ one of five types. The type reflects the local structure of the algebra, which can be one of the following: **1.** a finite set with a group action on it, **2.** a finite vector space over a finite field, **3.** a two-element Boolean algebra, **4.** a two-element lattice, **5.** a two-element semilattice. The type of a prime quotient $\alpha \prec \beta$ is denoted by $\mathsf{typ}(\alpha,\beta)$, while $\mathsf{typ}\mathbb{A}$ denotes the set $\{\mathsf{typ}(\alpha,\beta) \mid \alpha \prec \beta \in \mathsf{Con}\ \mathbb{A}\}$, and $\mathsf{typ}\mathfrak{A}$, where $\mathfrak{A}$ is a class of finite algebras, the set $\bigcup_{\mathbb{A}\in\mathfrak{A}} \mathsf{typ}\mathbb{A}$. If $\mathbf{i} \notin \mathsf{typ}\mathbb{A}$ [$\mathsf{typ}\mathfrak{A}$], then we say that $\mathbb{A}$ [$\mathfrak{A}$] omits type $\mathbf{i}$.

The types of an algebra strongly affect its complexity. For example, Proposition 4.8 can be stated in the following form.

Theorem 4.13 *Every tractable idempotent algebra omits type* **1**. *If an idempotent algebra does not omit type* **1** *then it is NP-complete.*

Proof. Suppose $\mathbf{1} \in \mathsf{typ}(\mathbb{A})$, that is there exist $\alpha, \beta \in \mathsf{Con}\ \mathbb{A}$ such that $\mathsf{typ}(\alpha, \beta) = \mathbf{1}$. We may assume that α is the equality relation $\underline{0}$, because if $\mathbb{A}/_{\alpha}$ is NP-complete, then $\mathbb{A}$ is NP-complete, and therefore $\mathbb{A}$ can be replaced with $\mathbb{A}/_{\alpha}$. Let e be a unary idempotent polynomial such that $U = e(A)$ is a $(\underline{0}, \beta)$-minimal set. Denote by $\mathbb{A}'$ the algebra obtained by adding e to the set of basic operations of $\mathbb{A}$. Since $\mathsf{SP}_{\mathrm{fin}}\mathbb{A}' \subseteq \mathsf{SP}_{\mathrm{fin}}\mathbb{A}$, NP-completeness of $\mathbb{A}'$ ensures NP-completeness of $\mathbb{A}$.

By Theorem 3.9, $\mathbb{A}'$ is NP-complete if and if only if $\mathbb{B} = e(\mathbb{A}') = \mathbb{A}'|_U$ is NP-complete. As is easily seen, since $\mathbb{A}$ is idempotent, $\mathbb{B}$ is also idempotent. Let $M \subseteq U$ be a $(\underline{0}, \beta)$-trace of $\mathbb{A}$. The trace M is a β-block of $\mathbb{B}$ and therefore is a subuniverse of $\mathbb{B}$. Since $\mathbb{B}|_M$ is term equivalent to a G-set, $\mathbb{B}|_M$ is NP-complete, that implies NP-completeness of $\mathbb{A}$. $\square$

Theorem 4.13 does not give any new examples of NP-complete algebras, because inside the class of idempotent algebras, the algebras omitting type **1** are, in fact, the algebras mentioned in Proposition 4.8. For a class of algebras, $\mathfrak{A}$, $\mathsf{H}\mathfrak{A}$ denotes the class of all homomorphic images of algebras from $\mathfrak{A}$.

Proposition 4.14 *Let $\mathbb{A}$ be an idempotent algebra. Then* $\mathsf{pvar}\mathbb{A}$ *contains a set if and only if* $\mathsf{HS}(\mathbb{A})$ *does not omit type* **1**.

Proof. Making use of Theorem 9.6 from [12], it is not hard to see that if $\mathbf{1} \in \mathsf{typ}(\mathsf{HS}(\mathbb{A})) \subseteq \mathsf{typ}(\mathsf{var}\mathbb{A})$, then $\mathsf{var}\mathbb{A}$, and therefore $\mathsf{pvar}\mathbb{A}$ contains a finite set. If $\mathsf{HS}(\mathbb{A})$ contains a set $\mathbb{B}$, then $\mathbf{1} \in \mathsf{typ}\mathbb{B} \subseteq \mathsf{HS}(\mathbb{A})$. So we just have to prove that if $\mathsf{pvar}\mathbb{A}$ contains a set then so does $\mathsf{HS}\mathbb{A}$.

Let $\mathbb{B}$ be a subalgebra of $\mathbb{A}^n$, and θ a congruence of $\mathbb{B}$ such that $\mathbb{C} = \mathbb{B}/_{\theta}$ is a set. For a set $I \subseteq \{1, \ldots, n\}$ and a_i, $i \in \overline{I} = \{1, \ldots, n\} - I$, denote by $\mathbb{B}_I[a_i, i \in \overline{I}]$ the set of all $\mathbf{b} \in \mathbb{B}$ with $b_i = a_i$ whenever $i \in \overline{I}$. Then, let I be a minimal set such that there are $a_i \in A$, $i \in \overline{I}$, such that $\mathbb{B}_I[a_i, i \in \overline{I}]$ is not contained in a class of θ. Without loss of generality, we may assume $I = \{1, \ldots, k\}$. Let also $a_{k+1}, \ldots, a_n \in A$ be the elements which witness the condition above. Since $\mathbb{A}$ is idempotent, $\mathbb{B}' = \mathbb{B}_I[a_{k+1}, \ldots, a_n]$ is a subalgebra of $\mathbb{B}$, and $\mathbb{C}' = \mathbb{B}'/_{\theta'}$ where $\theta' = \theta \cap \mathbb{B}'^2$ is a set.

Further, let $\mathbb{A}' = \{a \mid (a_1, \ldots, a_{k-1}, a, a_{k+1}, \ldots, a_n) \in \mathbb{B}'$ for some $a_1, \ldots, a_{k-1} \in A\}$, and η the image of θ' under the natural homomorphism $\mathbb{B}' \to \mathbb{A}'$. The relation η is not the total congruence of $\mathbb{A}'$. Indeed, otherwise there are $a \in A'$, and $a_1, \ldots, a_{k-1}, b_1, \ldots, b_{k-1} \in A$ such that $(a_1, \ldots, a_{k-1}, a, a_{k+1}, \ldots, a_n), (b_1, \ldots, b_{k-1}, a, a_{k+1}, \ldots, a_n)$ are in different θ'-classes, that contradicts the minimality of I. Therefore, $\mathbb{A}'/_{\eta} \in \mathsf{HS}(\mathbb{A})$ is a set. $\square$

Therefore Conjecture 4.12 can be re-stated as follows.

Conjecture 4.15 *An idempotent algebra $\mathbb{A}$ is (globally) tractable if and only if $\mathbf{1} \notin \mathsf{typ}(\mathsf{HS}(\mathbb{A}))$, otherwise $\mathbb{A}$ is NP-complete.*

5 Test for tractability

In this section we assume that Conjecture 4.15 holds. We exhibit an algorithm which under this assumption recognise tractability of a finite class of algebras and a set of relations. By Theorem 3.4, it is enough to consider the one-sorted case. The algorithms are, in fact, adapted versions of the algorithm presented in [2] for finding the type set of a finite algebra. In the last subsection we prove that the problem of recognising the tractability of a set of relations is NP-complete.

5.1 Subtraces and type set of an algebra

The straightforward algorithm computing the type set of a finite algebra includes the calculation of all congruences, unary and binary polynomials, and tends to be quite slow. To avoid inspecting all congruences and polynomials we need the notion of subtrace which was introduced in [2]. The algorithm from [2] is still of exponential time when being applied for calculating $\mathsf{typ}(\mathsf{HS}(\mathbb{A}))$. However, we do not need to calculate $\mathsf{typ}(\mathsf{HS}(\mathbb{A}))$, but only to check if $\mathbf{1} \in \mathsf{typ}(\mathsf{HS}(\mathbb{A}))$. Hence, we reproduce that part of the algorithm which is required for checking the condition $\mathbf{1} \in \mathsf{typ}(\mathsf{HS}(\mathbb{A}))$, and show how the algorithm can be changed so that it runs in polynomial time.

Definition 5.1 *Let $\mathbb{A}$ be a finite algebra with $\alpha \prec \beta$ in $\mathsf{Con}\,\mathbb{A}$. A subset S of A is called an (α, β)-*subtrace *of $\mathbb{A}$ if $S^2 \not\subseteq \alpha$ and S is a subset of an (α, β)-trace. The set S is a* subtrace of $\mathbb{A}$ *if there exist $\alpha \prec \beta$ in $\mathsf{Con}\,\mathbb{A}$ for which S is an (α, β)-subtrace.*

Let $\mathbb{A}$ be an algebra, and $(a, b) \in A^2$ with $a \neq b$. The ordered pair (a, b) is called a *1-snag* if there exists a binary polynomial f of $\mathbb{A}$ such that

$f(b,b) = b$ and $f(a,b) = f(b,a) = a$. By Theorem 7.2 from [12], if $\alpha \prec \beta$ in $\mathsf{Con}\ \mathbb{A}$ then $\mathsf{typ}(\alpha,\beta) = \mathbf{1}$ if and only if $\beta - \alpha$ contains no 1-snags.

Therefore to check whether $\mathbf{1} \in \mathsf{typ}(\mathbb{A})$ or not we have to inspect all the two-element subtraces looking for that which is not a 1-snag, and the two problems are

1. How to decide if a two-element set is a subtrace?

2. How to decide if a two-element set is a 1-snag?

The set of all unary polynomials of $\mathbb{A}$ is denoted by $\mathsf{Pol}_1\mathbb{A}$. Let $\mathbb{A}$ be an algebra and let B and C be non-void subsets of A. The sets B, C are called *polynomially isomorphic*, if there exist $f, g \in \mathsf{Pol}_1\mathbb{A}$ such that (i) $f(B) = C$, (ii) $g(C) = B$, (iii) $gf|_B = \mathrm{id}_B$, and (iv) $fg|_C = \mathrm{id}_C$. If B and C are polynomially isomorphic, then this is denoted $B \cong C$.

For a non-void subset S of an algebra $\mathbb{A}$, let $\sigma_{\mathbb{A}}(S)$ (or simply $\sigma(S)$ if $\mathbb{A}$ is clear from the context) denote the transitive closure of $\{f(S^2) \mid f \in \mathsf{Pol}_1\mathbb{A}, f(S) \ncong S\}$.

The following lemma gives us a tool for recognising subtraces.

Lemma 5.2 ([2]) *Let $\mathbb{A}$ be an algebra and $a, b \in A$. Then $\{a,b\}$ is a subtrace if and only if $(a,b) \notin \sigma(a,b)$.*

By the definition, $(c,d) \in \sigma(a,b)$ if and only if there exist $n \geq 1$ and $z_0, z_1, \ldots, z_n \in A$ with $c = z_0$, $d = z_n$ and there are $f_1, \ldots, f_n \in \mathsf{Pol}_1\mathbb{A}$ such that for each i, $1 \leq i \leq n$, $\{z_i, z_{i-1}\} = \{f_i(a), f_i(b)\}$ and $\{a,b\} \ncong \{f_i(a), f_i(b)\}$. Therefore the first problem can be solved in the following way.

Let $|A| = k$ and let $G(\mathbb{A})$ denote the directed graph whose vertices are all $\binom{k}{2}$ two-element subsets of $\mathbb{A}$ and $(\{a,b\},\{c,d\})$ is an edge in $G(\mathbb{A})$ if and only if there exists a $f \in \mathsf{Pol}_1\mathbb{A}$ such that $\{c,d\} = \{f(a), f(b)\}$. As is easily seen, graph $G(\mathbb{A})$ is reflexive and transitive. Since $\{a,b\}$ and $\{c,d\}$ are polynomially isomorphic if and only if $(\{a,b\},\{c,d\})$, $(\{c,d\},\{a,b\})$ are both edges of $G(\mathbb{A})$, the strongly connected components of $G(\mathbb{A})$ are exactly the classes of the equivalence relation $\cong$ on the set of two-element subsets of A. Further, for $\{a,b\} \subseteq A$, let $G_{a,b}(\mathbb{A})$ denote the graph with vertex set A, and such that (c,d) is an edge of $G_{a,b}$ if and only if $(\{a,b\},\{c,d\})$ is an edge of $G(\mathbb{A})$ and $\{a,b\} \ncong \{c,d\}$. Then $\{a,b\}$ is a subtrace in $\mathbb{A}$ if and only if there is no path from a to b in the graph $G_{a,b}$.

In order to determine if there exists a unary polynomial f such that $\{c,d\} = \{f(a), f(b)\}$, it suffices to consider the subalgebra of $\mathbb{A}^2$ generated by (a, b) and all pairs of the form (u, u), $u \in A$. The required polynomial exists if and only if (c, d) or (d, c) belongs to the subalgebra.

Analogously, by the definition of 1-snag, a subset $\{a, b\}$ is a 1-snag if and only if the subalgebra of $\mathbb{A}^3$ generated by $(a, b, b), (b, a, b)$ and all triples of the form (u, u, u), $u \in A$, contains (a, a, b).

Clearly, if $\mathbb{B}$ is a homomorphic image of $\mathbb{A}$ then $\mathsf{typ}(\mathbb{B}) \subseteq \mathsf{typ}(\mathbb{A})$. So, to determine if $\mathbf{1} \in \mathsf{typ}(\mathsf{HS}(\mathbb{A}))$ we just should check if $\mathbf{1} \in \mathsf{typ}(\mathsf{S}(\mathbb{A}))$. The straightforward algorithm that runs along the set of all subalgebras is still exponential for some hard cases. Indeed, a k-element algebra may have up to 2^k subalgebras. It turns out, to verify if $\mathbf{1} \in \mathsf{typ}(\mathsf{S}\mathbb{A})$ it is enough to check the subalgebras generated by two-element sets. Let $\mathsf{S}_2\mathbb{A}$ denote the set of all subalgebras of a finite algebra $\mathbb{A}$ generated by a two-element set.

Proposition 5.3 $\mathbf{1} \in \mathsf{typ}(\mathsf{S}\mathbb{A})$ *if and only if* $\mathbf{1} \in \mathsf{typ}(\mathsf{S}_2\mathbb{A})$.

Proof. If $\mathbf{1} \in \mathsf{typ}(\mathsf{S}\mathbb{A})$ then there is a subalgebra $\mathbb{B}$ of $\mathbb{A}$ and a two-element subtrace $\{a, b\}$ such that neither (a, b), nor (b, a) is a 1-snag in $\mathbb{B}$. Denote by $\mathbb{C}$ the subalgebra generated by $\{a, b\}$. Since $\mathsf{Pol}_1\mathbb{C} \subseteq \mathsf{Pol}_1\mathbb{B}|_C$, we have $\sigma_{\mathbb{C}}(a, b) \subseteq \sigma_{\mathbb{B}}(a, b) \cap C^2$. Therefore $(a, b) \notin \sigma_{\mathbb{C}}(a, b)$ and, by Lemma 5.2, $\{a, b\}$ is a subtrace of $\mathbb{C}$.

By the assumption, $\mathbb{B}$ has no polynomial operation that turns (a, b) or (b, a) to a 1-snag. Since each polynomial operation of $\mathbb{C}$ is the restriction of a polynomial operation of $\mathbb{B}$, $(a, b), (b, a)$ are not 1-snag in $\mathbb{C}$. $\square$

5.2 Algorithms

5.2.1 Test for algebras.

ALGORITHM.
Input: Operation tables of the basic operations of an idempotent algebra $\mathbb{A}$.
Output: 'YES' if $\mathbf{1}$ is not in the set $\mathsf{typ}(\mathsf{S}(\mathbb{A}))$, 'NO' otherwise.

- For each $a \neq b$ in A do
  - Let $\mathbb{C}$ be the subalgebra generated by $\{a, b\}$.
  - Check if $\{a, b\}$ is a subtrace. For this do
    * Build the graph $G(\mathbb{C})$. The vertices of $G(\mathbb{C})$ are all 2-element subsets of C and $(\{c, d\}, \{e, f\})$ is an edge of $G(\mathbb{C})$ if and only if (e, f) or (f, e) belongs to the subalgebra of $\mathbb{C}^2$ generated by (c, d),(c, c),(d, d).

    * Build the graph $G_{a,b}(\mathbb{C})$. Its vertices are elements of C and (c,d) is an edge if and only if $(\{a,b\},\{c,d\})$ is an edge of $G(\mathbb{C})$, but $(\{c,d\},\{a,b\})$ is not.
    * Check if there is a path from a to b in $G_{a,b}(\mathbb{C})$. If there is no such a path then $\{a,b\}$ is a subtrace, otherwise it is not.
  - If $\{a,b\}$ is a subtrace, determine if (a,b) or (b,a) is a 1-snag. For this do
    * Generate a subalgebra $\mathbb{B}$ of $\mathbb{C}^3$ by triples (a,b,b), (b,a,b), (a,a,a), (b,b,b).
    * Check if the triple (a,a,b) belongs to $\mathbb{B}$. If yes then (a,b) is a 1-snag, otherwise it is not.
  - If $\{a,b\}$ is a subtrace, and neither (a,b) nor (b,a) is a 1-snag then OUTPUT('NO') and STOP.
- OUTPUT('YES') and STOP.

To estimate the time complexity of the algorithm, we note that the crucial subroutine is generating of a subalgebra of $\mathbb{A}^3$. Such a subalgebra consists of at most k^3 elements where $k = |A|$. If g is a basic operation of arity m, then testing all elements of the subalgebra as arguments of g contributes time complexity $O(k^{3m})$. If $\mathbb{A}$ has l basic operations and if m is the largest arity of a basic operation, the worst case time complexity for generating a subalgebra of $\mathbb{A}^3$ is $l \cdot k^{3m}$. Assuming that the basic operations are given by their operation tables, the input is of size $O(k^m)$. Therefore, the time complexity is bounded by a cubic function on the length of the input.

5.2.2 Test for a set of relations.

Originally the theory and practice of the constraint satisfaction problem dealt with sets of relations rather than algebras. So, the input of the algorithm above, which is very natural from the algebraic point of view, seems to be quite exotic for the CSP. On the other hand, the problem: given set of relations, Γ, construct a set of basic operations of the algebra $(A, \mathsf{Pol}\,\Gamma)$, is believed to be very hard, and in some cases undecidable. Thus we need a method that allows us to recognise whether $(A; \mathsf{Pol}\,\Gamma)$ omits type **1** or not from the set Γ.

The main tool in the algorithm above is generating subalgebras of powers of $\mathbb{A}$. So, to adapt this algorithm to manage with relations we are to exhibit a way of constructing the subalgebra of $(A; \mathsf{Pol}\ \Gamma)^n$ generated by a subset from A^n using only relations from Γ.

To this end we just have to note that, for any $S = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\} \subseteq A^n$, an element $\mathbf{b}$ belongs to the subalgebra $\mathbb{B}$ generated by S if and only if there exists $f(x_1, \ldots, x_k) \in \mathsf{Pol}\,\Gamma$ such that $\mathbf{b} = f(\mathbf{a}_1, \ldots, \mathbf{a}_k)$.

The following algorithm gives the subalgebra from $(A; \mathsf{Pol}\ \Gamma)^n$ generated by a set $S \subseteq A^n$.

Algorithm.
Input: A finite set of relations, Γ, on A, a subset $S = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\} \subseteq A^n$.
Output: The subuniverse C of $(A; \mathsf{Pol}\ \Gamma)^n$ generated by S.

- $C := S$.
- For each $\mathbf{b} \in A^n$ do
  - Check if there exists a k-ary operation f preserving Γ and such that
    $f(\mathbf{a}_1, \ldots, \mathbf{a}_k) = \mathbf{b}$.
  - If yes then $C := C \cup \{\mathbf{b}\}$.
- OUTPUT(C) and STOP.

This algorithm can be used as a subroutine generating subalgebras for the algorithm from Section 5.1. Unfortunately it requires exponential time in general, because checking if there exists an operation with the required properties makes up to $|A|^{|A|^k}$ attempts.

Another problem which appears when testing a set of relations Γ: it is not guaranteed that $(A; \mathsf{Pol}\ \Gamma)$ is an idempotent algebra. This means that we have to make use of Propositions 3.9 and 3.10 before employing the algorithm from page 23.

Algorithm.
Input. A set A, and a set of relations, Γ, on A.
Output. 'YES' if Γ is tractable, 'NO' otherwise[3].

- Find an idempotent unary term e with a minimal range.
- Set $\Gamma' = \{e(\varrho) \mid \varrho \in \Gamma\} \cup \{\{a\} \mid a \in e(A)\}$.
- Invoke the algorithm from page 23 for $e(A)$ and Γ'.

[3]Recall, we are assuming that Conjecture 4.15 holds.

5.3 Complexity of the test

As is shown in the previous subsection, if we are given the operation tables of the basic operations of an algebra, then the tractability of the associated problem class can be recognised in polynomial time. The situation is quite different if we are given a set Γ of relations on a set A. We are going to prove that if the number $|A|$ is not bounded, then recognising tractability of Γ is NP-complete.

We consider three decision problems.

CSP-TRACTABILITY-OF-ALGEBRA
Instance. A finite set A and operation tables of idempotent operations $f_1, \dots, f_n$ on A.
Question. Does $\mathsf{HS}(\mathbb{A})$ where $\mathbb{A} = (A; \{f_1, \dots, f_n\})$ omit type $\mathbf{1}$?

CSP-TRACTABILITY
Instance. A finite set A and a finite set of relations, Γ, on A.
Question. Does $\mathsf{HS}(\mathbb{A})$ where $\mathbb{A} = (A; \mathsf{Pol}\,\Gamma)$ omit type $\mathbf{1}$?

CSP-TRACTABILITY(k)
Instance. A finite set A, $|A| \le k$, and a finite set of relations, Γ, on A.
Question. Does $\mathsf{HS}(\mathbb{A})$ where $\mathbb{A} = (A; \mathsf{Pol}\,\Gamma)$ omit type $\mathbf{1}$?

Theorem 5.4 *1) The problem* CSP-TRACTABILITY-OF-ALGEBRA *is tractable. 2) The problem* CSP-TRACTABILITY(k) *is tractable. 3) The problem* CSP-TRACTABILITY *is NP-complete.*

Proof. 1) The algorithm from page 23 witnesses the tractability of CSP-TRACTABILITY-OF-ALGEBRA.

2) Notice that all operations involved in the algorithm from page 23 are at most 4-ary. It can be easily verified that the time complexity of that algorithm is $O(k^{k^4} \cdot n \cdot m)$ where n is the maximal arity of the relations, and m the is total number of tuples in all the relations.

3) We reduce NAE to CSP-TRACTABILITY. Let $\mathcal{C} = C_1 \wedge \ldots \wedge C_q$ be an instance of NAE. Let V denote the set of variables occurring in $\mathcal{C}$. We construct a set of relations Γ.

Set $A_0 = \{a, b\}$, $A_v = W_v \cup A_v^0 \cup A_v^1$, $v \in V$, where $W_v = \{c_v^1, c_v^2, c_v^3, c_v^4, c_v^5, c_v^6\}$, $A_v^0 = \{0_v, 0'_v\}$, $A_v^1 = \{1_v, 1'_v\}$; and $A = A_0 \cup \bigcup_{v \in V} A_v$. For each $1 \le i \le q$, let $\{u, v, w\}$ be the variables occurring in C_i. Define a 6-ary relation ϱ_i in the following way

$$\begin{pmatrix} a & b & a & b & b & a & a & a & a & a & a & a & a & a & a & a & a & a \\ b & a & a & b & a & b & a & a & a & a & a & a & a & a & a & a & a & a \\ b & b & a & b & a & a & b & b & b & b & b & b & b & b & b & b & b & b \\ c_u^1 & c_u^2 & c_u^3 & c_u^4 & c_u^5 & c_u^6 & 0_u & 0'_u & 0_u & 0'_u & 0_u & 0'_u & 1_u & 1'_u & 1_u & 1'_u & 1_u & 1'_u \\ c_v^1 & c_v^2 & c_v^3 & c_v^4 & c_v^5 & c_v^6 & 0_v & 0'_v & 1_v & 1'_v & 1_v & 1'_v & 0_v & 0'_v & 0_v & 0'_v & 1_v & 1'_v \\ c_w^1 & c_w^2 & c_w^3 & c_w^4 & c_w^5 & c_w^6 & 1_w & 1'_w & 1_w & 1'_w & 0_w & 0'_w & 1_w & 1'_w & 0_w & 0'_w & 0_w & 0'_w \end{pmatrix}$$

Finally, set $\Gamma = \{(a) \mid a \in A\} \cup \{\theta\} \cup \varrho_1 \cup \ldots \cup \varrho_q$ where θ is the equivalence relation whose blocks are $\{a\}$ and $A' = A - \{a\}$. Denote by $\mathbb{A}$ the algebra $(A; \mathsf{Pol}\,\Gamma)$.

For $c, d \in A$ we shall write $c \equiv d$ if $c = 0_u, d = 0_v$, or $c = 0'_u, d = 0'_v$, or $c = 1_u, d = 1_v$, or $c = 1'_u, d = 1'_v$ for some $u, v \in V$. Further, for $c, d \in A_v^0 \cup A_v^1$ we shall write $c \cong d$ if c, d both lie either in A_v^0 or in A_v^1 for some $v \in V$, and $c \sim d$ if $c, d \in \{0_v, 1_v\}$ or $c, d \in \{0'_v, 1'_v\}$. Denote the tuples $(a,b,b), (b,a,b), (a,a,a), (b,b,b), (b,a,a), (a,b,a), (a,a,b)$ by $\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3, \mathbf{c}^4, \mathbf{c}^5, \mathbf{c}^6, \mathbf{c}^7$ respectively.

We prove two assertions:

Claim 1. $\mathsf{HS}(\mathbb{A})$ omits type $\mathbf{1}$ if and only if there is an operation f such that

$$f\begin{pmatrix} a & b & a & b \\ b & a & a & b \\ b & b & a & b \end{pmatrix} = \begin{pmatrix} a \\ a \\ b \end{pmatrix}.$$

Claim 2. $\mathcal{C}$ has a solution if and only if f exists.

Proof first Claim 1. As is easily seen $\{a, b\}$ is a subalgebra. Post's description of clones on a 2-element set [30] implies that if this subalgebra is not of type $\mathbf{1}$, then it has either a semilattice, or affine, or majority term, and therefore, an operation f with

$$f\begin{pmatrix} a & b & a & b \\ b & a & a & b \\ b & b & a & b \end{pmatrix} = \begin{pmatrix} a \\ a \\ b \end{pmatrix} \quad \text{or} \quad f\begin{pmatrix} b & a & a & b \\ a & b & a & b \\ a & a & a & b \end{pmatrix} = \begin{pmatrix} b \\ b \\ a \end{pmatrix}.$$

However, in the latter case f does not preserve relations ϱ_i. Thus, if there is no such operation then $\{a, b\}$ is a subalgebra of type $\mathbf{1}$, and we are to prove that if an operation with required properties exists then $\mathsf{HS}(\mathbb{A})$ omits type $\mathbf{1}$.

Claim 3. If a subset of A is a subuniverse of $\mathbb{A}$, then it is either one-element, or is one of A, A', $\{a, b\}$, C, $C \cup A_v^0 \cup A_v^1$, $C \cup \{d\}$ where $d \in A_v^0 \cup A_v^1$, $C \subseteq W_v$, $v \in V$.

Obviously, $A', \{a,b\}, W_v \cup A_v^0 \cup A_v^1$ are subalgebras, because they are projections of certain relations from Γ, or a class of θ. Take a subset $B \subseteq A$. Suppose first $B \neq A'$, $B \not\subseteq \{a,b\}$, $B \not\subseteq W_v \cup A_v^0 \cup A_v^1$, for any $v \in V$, say $B = \{a_1, \ldots, a_k\}$ $(a_1 \neq a)$. Then let k-ary operation $g_B(x_1, \ldots, x_k)$ be such that

$$\begin{aligned} g_B(a_1, \ldots, a_k) &= c \in A' - B, \\ g_B(x_1, \ldots, x_k) &= x_1, \text{ otherwise.} \end{aligned}$$

We have to prove that g_B preserves relations from Γ. By (a), (b), (c) we mark the parts of the proof corresponding to the parts of Γ.

(a) g_B obviously preserves unary one-element relations, but destroys B.

(b) Take $(x_1, y_1), \ldots, (x_k, y_k) \in \theta$. Then $g_B\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \cdots \begin{pmatrix} x_k \\ y_k \end{pmatrix}\right) = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ whenever neither $(x_1, \ldots, x_k)$ nor $(y_1, \ldots, y_k)$ equals $(a_1, \ldots, a_k)$. If $(x_1, \ldots, x_k) = (a_1, \ldots, a_k)$ then $x_1 \in A'$, and hence, $y_1 \in A'$. Thus $g_B(x_1, \ldots, x_k), g_B(y_1, \ldots, y_k) \in A'$.

(c) For any $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \varrho_i$ where $\mathbf{x}_j = (x_{1j}, \ldots, x_{6j})$, none of $(x_{l1}, \ldots, x_{lk})$ equals $(a_1, \ldots, a_k)$. Hence, $g_B(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \mathbf{x}_1$ which means g_B preserves ϱ_i.

Further, suppose that $|(A_v^0 \cup A_v^1) \cap B| \geq 2$, but $A_v^0 \cup A_v^1 \not\subseteq B$. Let $c, d \in (A_v^0 \cup A_v^1) \cap B$ and $e \in (A_v^0 \cup A_v^1) - B$. There are 12 possibilities of what c, d, e are. We consider one of them, namely, when $c = 0_v, d = 0'_v, e = 1_v$. The other 11 cases are quite analogous. Define the operation $g_{c,d,e}(x,y)$ as follows

$$g_{c,d,e}(x,y) = \begin{cases} e, & \text{if } x = c,\ y = d, \\ 0_u, & \text{if } x \in A_u^1,\ y \in A_u^0 \cup A_u^1,\ u \in V,\ x \not\sim y; \\ 1_u, & \text{if } x \in A_u^0,\ y \in A_u^0 \cup A_u^1,\ u \in V,\ x \not\sim y; \\ x & \text{otherwise.} \end{cases}$$

a) is obvious.

b) Let $\begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = g_{c,d,e}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right)$ where (x_1, x_2), $(y_1, y_2) \in \theta$. If $x_1, x_2 \neq a$ then $(g_1, g_2) \in A'^2 \subseteq \theta$. If $x_1 = a$ then $x_2 = a$ and $(g_1, g_2) = (a, a) \in \theta$.

c) Let $\mathbf{g} = g_{c,d,e}(\mathbf{x}, \mathbf{y})$, $\mathbf{x}, \mathbf{y} \in \varrho_i$. If (x_1, x_2, x_3) or (y_1, y_2, y_3) does not equal (a, a, b), then $\mathbf{g} = \mathbf{x}$. If $(x_1, x_2, x_3) = (y_1, y_2, y_3) = (a, a, b)$, then $x_4 \equiv x_5 \equiv x_6$, $y_4 \equiv y_5 \equiv y_6$. Therefore $(g_1, g_2, g_3) = (a, a, b)$; $g_4 \equiv g_5 \equiv g_6$;

$(g_4, g_5, g_6) = (x_4, x_5, x_6)$ when $x_i \sim y_i$, $i \in \{4,5,6\}$, and $g_i = 0_v$ if $x_i \in A_v^1$, $g_i = 1_v$ if $x_i \in A_v^0$ when $x_i \not\sim y_i$, $i \in \{4,5,6\}$. In both cases $\mathbf{g} \in \varrho_i$.

Claim 4. For each subalgebra $\mathbb{B}$ of $\mathbb{A}$, and a congruence η of $\mathbb{B}$, $\mathbb{B}/_\eta$ is not a set.

The algebra $(\{a,b\}^3, f)$ has either a semilattice, or a majority, or a minority operation. Therefore the algebra $(\{c_v^1, c_v^2, c_v^3, c_v^4, c_v^5, c_v^6\} \cup A_v^0 \cup A_v^1; f)$ being factorised modulo η_v, whose blocks are $\{c_v^1\}, \{c_v^2\}, \{c_v^3\}, \{c_v^4\}, \{c_v^5\}, \{c_v^6\}, A_v^0 \cup A_v^1$, also has one of those operations. So, the only subalgebras we have to check are $A, A', A_v^0 \cup A_v^1$.

1) $A_v^0 \cup A_v^1$ has no nontrivial subalgebras. Since it is idempotent, this implies simplicity. Making use of the description of idempotent strictly simple algebras [34] it is not a set.

2) $(A/_\theta; f)$ is isomorphic to $(\{a,b\}; f)$, and therefore, is not a set. Furthermore, any congruence of $\mathbb{A}$ which differs from θ and the total relation is not total on A'.

3) $\mathbb{A}' = \mathbb{A}|_{A'}$ is simple.

Since each congruence block is a subalgebra, any nontrivial congruence η is a subset of the equivalence relation η' whose blocks are $\{b\}$, $W_v \cup A_v^0 \cup A_v^1$, $v \in V$. For each $c, d \in W_v \cup A_v^0 \cup A_v^1$ define an operation $h_{c,d}$ as follows

$$h_{c,d}(x,y) = \begin{cases} c, & \text{if } x = c, y = b, \\ b, & \text{if } x = d, y = b, \\ x, & \text{otherwise.} \end{cases}$$

(a) $h_{c,d}$ is idempotent, consequently, it preserves all unary relations from Γ.

(b) $\begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = h_{c,d}\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ whenever $a \in \{x_1, x_2, y_1, y_2\}$, and $h_1, h_2 \in A'$ otherwise.

(c) Set $\mathbf{h} = h_{c,d}(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}, \mathbf{y} \in \varrho_i$. Since $h_{c,d}(x,y) = x$ if $x, y \in \{a,b\}$, we have $(h_1, h_2, h_3) = (x_1, x_2, x_3)$. Further, since $y_4, y_5, y_6 \neq b$, $(h_4, h_5, h_6) = (x_4, x_5, x_6)$.

This means that if $(c,d) \in \eta$, then $(c,b) \in \eta$, but $(c,b) \notin \eta'$, a contradiction with $\eta \subseteq \eta'$. Since $\mathbb{A}'$ is not a set, Claim 1 is proved.

Let us prove Claim 2. Note that if f is as required, then for each C_i with $V_i = \{u, v, w\}$ we have

$$\begin{aligned} f(c_u^1, c_u^2, c_u^3, c_u^4) &\in A_u^a \\ f(c_v^1, c_v^2, c_v^3, c_v^4) &\in A_v^b \\ f(c_w^1, c_w^2, c_w^3, c_w^4) &\in A_w^c \end{aligned}$$

where (a,b,c) is a solution for C_i. Therefore, $\mathcal{C}$ has a solution.

Conversely, suppose $\mathcal{C}$ has a solution $\varphi\colon V \to \{0,1\}$. Set

$$f(x,y,z,t) = g(x,y) = \begin{cases} x, & \text{if } x = y, \\ a, & \text{if } x = a \text{ or } y = a, \\ \varphi(v)_v, & \text{if } \{x,y\} = \{c_v^1, c_v^2\}, \\ & \text{or } x \in \{c_v^1, c_v^2, c_v^4\},\ y \in A_v^0 \cup A_v^1, \\ & \text{or } y \in \{c_v^1, c_v^2, c_v^4\},\ x \in A_v^0 \cup A_v^1, \\ c_v^3, & \text{if } x = c_v^3,\ y \in W_v \cup A_v^0 \cup A_v^1, \\ & \text{or } y = c_v^3,\ x \in W_v \cup A_v^0 \cup A_v^1, \\ & \text{or } \{x,y\} = \{c_v^1, c_v^5\}, \{x,y\} = \{c_v^2, c_v^6\}, \\ & \text{or } \{x,y\} \cap (A_v^0 \cup A_v^1) \neq \varnothing \\ & \text{and } \{x,y\} \cap \{c_v^5, c_v^6\} \neq \varnothing, \\ c_v^1, & \text{if } \{x,y\} = \{c_v^1, c_v^4\}, \\ c_v^2, & \text{if } \{x,y\} = \{c_v^2, c_v^4\}, \\ c_v^5, & \text{if } \{x,y\} = \{c_v^2, c_v^5\}, \{c_v^4, c_v^5\}, \\ c_v^6, & \text{if } \{x,y\} = \{c_v^1, c_v^6\}, \{c_v^4, c_v^6\}, \\ x, & \text{otherwise.} \end{cases}$$

(a) Since f is idempotent, it preserves all unary relations from Γ.

(b) As is easily seen, f preserves A'. Take $(x_1,x_2),(y_1,y_2) \in \theta$. If $a \notin \{x_1,x_2,y_1,y_2\}$ then $g\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right) \in A'^2 \subseteq \theta$. Otherwise, if $x_1 = a$ $(y_1 = a)$ then $x_2 = a$ $(y_2 = a)$; therefore $g\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right) = \begin{pmatrix} a \\ a \end{pmatrix} \in \theta$.

(c) Denote the set of tuples $\{\mathbf{b} \mid \begin{pmatrix} \mathbf{c}^i \\ \mathbf{b} \end{pmatrix} \in \varrho_t\}$ by D_{it}. We just have to prove that $g(\mathbf{c}^i, \mathbf{c}^j) \in \{\mathbf{c}^1, \dots, \mathbf{c}^6\}$, and if $g(\mathbf{c}^i, \mathbf{c}^j) = \mathbf{c}^k$ then $g(D_{it}, D_{jt}) \subseteq D_{kt}$. However, this follows straightforwardly from the definition of g. $\square$

References

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

[2] J.D. Berman, E.W. Kiss, P. Proehle, and Á. Szendrei. The set of types of a finitely generated variety. *Discrete Math.*, 112(1-3):1–20, 1993.

[3] 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.

[4] A.A. Bulatov, A.A. Krokhin, and P.G. Jeavons. Classifying complexity of constraints using finite algebras. Submitted to SIAM Journal of Computing.

[5] 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.

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

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

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

[9] 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.

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

[11] M. Gyssens, P.G. Jeavons, and D.A. Cohen. Decomposing constraint satisfaction problems using database techniques. *Artificial Intelligence*, 66(1):57–89, 1994.

[12] D. Hobby and R.N. McKenzie. *The Structure of Finite Algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, R.I., 1988.

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

[14] P.G. Jeavons and D.A. Cohen. An algebraic characterization of tractable constraints. In *Computing and Combinatorics. First International Conference COCOON'95 (Xi'an,China,August 1995)*, volume 959 of *Lecture Notes in Computer Science*, pages 633–642. Springer-Verlag, 1995.

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

[16] P.G. Jeavons, D.A. Cohen, and M. Gyssens. A unifying framework for tractable constraints. In *Proceedings 1st International Conference on Constraint Programming—CP'95 (Cassis, France, September 1995)*, volume 976 of *Lecture Notes in Computer Science*, pages 276–291. Springer-Verlag, 1995.

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

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

[19] P.G. Jeavons and M.C. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327–339, 1995.

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

[21] P.B. Ladkin and R.D. Maddux. On binary constraint problems. *Journal of the ACM*, 41:435–469, 1994.

[22] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.

[23] A.K. Mackworth. Constraint satisfaction. In S.C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1, pages 285–293. Wiley Interscience, 1992.

[24] A.K. Mackworth and E.C. Freuder. The complexity of constraint satisfaction revisited. *Artificial Intelligence*, 59:57–62, 1993.

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

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

[27] 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.

[28] C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

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

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

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

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

[33] A. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminaires de Mathematiques Superieures*. University of Montreal, 1986.

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

[35] E. Tsang. *Foundations of Constraint Satisfaction*. Academic Press, London, 1993.

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

Andrei Bulatov
Department of Algebra and Discrete Mathematics
Ural State University,
Ekaterinburg, Russia
e-mail: Andrei.Bulatov@usu.ru

Peter Jeavons
Computing Laboratory
University of Oxford, UK
e-mail: Peter.Jeavons@comlab.ox.ac.uk