Constraint Satisfaction Problems: Complexity and Algorithms

Andrei A. Bulatov[1], Simon Fraser University

In this paper we briefly survey the history of the Dichotomy Conjecture for the Constraint Satisfaction problem, that was posed 25 years ago by Feder and Vardi. We outline some of the approaches to this conjecture, and then describe an algorithm that yields an answer to the conjecture.

1. THE COMPLEXITY DICHOTOMY PHENOMENON

It is well known that if P$\neq$NP, then there are infinitely many distinct complexity classes between them [Ladner 1975]. However, the only known examples of such classes are constructed through diagonalization, and therefore are somewhat artificial. It is thus an appealing idea to suggest that no 'natural' problem attains any of those intermediate complexity classes and either belongs to P or is NP-complete. Proving such a result for all 'natural' problems may be difficult, as it is not clear what a 'natural' problem may mean precisely. However, in a more practical perspective one can try to show that large classes of problems that are arguably 'natural' enjoy this dichotomy property. This approach has been suggested for several classes of problems, most notably by Valiant in the context of counting problems, the Holant problem and holographic algorithms [Valiant 2006; 2008], and by Feder and Vardi in the context of the Constraint Satisfaction Problem (CSP) [Feder and Vardi 1993; 1998]. Since this paper is about the CSP, we discuss the latter research direction in more details.

In the most concise way the CSP of the type studied by Feder and Vardi can be represented through homomorphisms of relational structures. A Constraint Satisfaction Problem, given relational structures $\mathcal{G}, \mathcal{H}$ with the same vocabulary, asks whether or not there is a homomorphism from $\mathcal{G}$ to $\mathcal{H}$. We will refer to this definition of the CSP as the homomorphism definition. As is easily seen, it allows for various kinds of restrictions and parametrizations, say, by restricting the classes of structures from which $\mathcal{G}$ and $\mathcal{H}$ can be taken from. The nonuniform CSP for a relational structure $\mathcal{H}$, denoted $\mathrm{CSP}(\mathcal{H})$, asks whether there exists a homomorphism from a given structure $\mathcal{G}$ to $\mathcal{H}$.

In [Feder and Vardi 1993; 1998] Feder and Vardi start off by a systematic study of problems representable through logic formulas that may exhibit the dichotomy phenomenon. Due to Fagin's theorem we know that every problem in NP can be represented as deciding if the input structure satisfies a second-order existential formula. So, Feder and Vardi tried to identify the largest class of formulas that does not represent the whole of NP, and therefore may have the dichotomy property. The candidate class they arrived at was MMSNP, the class of monotone monadic formulas without inequalities. In particular, they show that if any of these three properties is eliminated, any problem from NP is polynomial time equivalent to a problem in the resulting class, and therefore this class cannot have the dichotomy property. On the other hand, they show that for every problem from the class MMSNP itself there is a randomized polynomial time reduction to a certain nonuniform CSP[2]. This led them to the following CSP Dichotomy Conjecture

CONJECTURE 1.1 (CSP DICHOTOMY CONJECTURE, [FEDER AND VARDI 1993; 1998]). *For every finite relational structure $\mathcal{H}$, the nonuniform constraint satisfaction problem,* $\mathrm{CSP}(\mathcal{H})$*, either can be solved in polynomial time, or is NP-complete.*

In this paper we discuss how the CSP Dichotomy Conjecture has been resolved, starting from early dichotomy results of Schaefer [Schaefer 1978] on the Generalized Satisfiability Problem, and Hell and Nesetril [Hell and Nešetřil 1990] on Graph H-Colouring, to its final resolution by the author [Bulatov 2017b; 2017c] and independently and nearly simultaneously by Zhuk [Zhuk 2017a; 2018]. Most of the paper uses the links between the CSP and universal algebra — the approach that has been very effective in the study of the CSP. However, before turning to algebraic concepts we touch upon other approaches and provide a number of results and examples obtained there.

Complexity classification of the decision CSP and its variants has been moved beyond the Dichotomy conjecture. Dichotomy or other complexity classification results have been proved (or sometimes conjectured) for a number of variants of the CSP. We will mention some of these results towards the end of the paper.

2. THE CONSTRAINT SATISFACTION PROBLEM

We begin with defining the CSP in a way that historically has been used by researchers in Artificial Intelligence, and will be convenient in this paper. In the definition below tuples of elements are denoted in boldface, say, $\mathbf{a}$, and the ith component of $\mathbf{a}$ is referred to as $\mathbf{a}[i]$. The set $\{1, \dots, n\}$ will be denoted by $[n]$.

Definition 2.1. Let $A_1, \dots, A_\ell$ be finite sets. An instance $\mathcal{I} = (V, \mathcal{C})$ of the CSP over $A_1, \dots, A_\ell$ consists of a finite set of variables V such that each $v \in V$ is assigned a domain A_{i_v}, $i_v \in [\ell]$, and a finite set $\mathcal{C}$ of constraints. Each constraint is a pair $\langle \mathbf{s}, R \rangle$ where R is a relation over $A_1, \dots, A_\ell$ (say, k-ary), often called the *constraint relation*, and $\mathbf{s}$ is a k-tuple of variables from V, called the *constraint scope*. Let $\sigma : V \to A = A_1 \cup \dots \cup A_\ell$ be a mapping with $\sigma(v) \in A_{i_v}$; we write $\sigma(\mathbf{s})$, for $(\sigma(\mathbf{s}[1]), \dots, \sigma(\mathbf{s}[k]))$. A *solution* of $\mathcal{I}$ is a mapping $\sigma : V \to A$ such that for every constraint $\langle \mathbf{s}, R \rangle \in \mathcal{C}$ we have $\sigma(\mathbf{s}) \in R$. The objective in the CSP is to decide whether or not a solution of a given instance $\mathcal{I}$ exists.

Since its inception in the early 70s [Mackworth 1977], the CSP has become a very popular and powerful framework, widely used to model computational problems first in artificial intelligence, [Dechter 2003] and later in many other areas.

Restrictions of the general CSP similar to nonuniform CSPs from the previous section can be introduced through constraint languages. Let $A_1, \dots, A_\ell$ be finite sets and Γ a set (finite or infinite) of relations over $A_1, \dots, A_\ell$, called a *constraint language*. Then $\mathrm{CSP}(\Gamma)$ is the class of all instances $\mathcal{I}$ of the CSP such that $R \in \Gamma$ for every constraint $\langle \mathbf{s}, R \rangle$ from $\mathcal{I}$. The following examples are just a few of the problems representable as $\mathrm{CSP}(\Gamma)$.

Example 2.2. (k-COL) The standard k-Colouring problem has the form $\mathrm{CSP}(\Gamma_{k\mathrm{COL}})$, where $\Gamma_{k\mathrm{COL}} = \{\neq_k\}$ and $\neq_k$ is the disequality relation on a k-element set (of colours).

[2]These reductions have been derandomized by Kun [Kun 2013].

(3-SAT) An instance of the 3-SAT problem is a propositional logic formula in CNF each clause of which contains 3 literals, and the goal is to decide if it has a satisfying assignment. Thus, 3-SAT is equivalent to $\mathrm{CSP}(\Gamma_{3\mathrm{SAT}})$, where $\Gamma_{3\mathrm{SAT}}$ is the constraint language on $\{0,1\}$ that contains relations $R_1,\ldots,R_8$, which are the 8 ternary relations that can be expressed by a 3-clause.

(LIN) Let F be a finite field and let $\mathrm{LIN}(F)$ be the problem of deciding the consistency of a system of linear equations over F. Then $\mathrm{LIN}(F)$ is equivalent to $\mathrm{CSP}(\Gamma_{\mathrm{LIN}(F)})$, where $\Gamma_{\mathrm{LIN}(F)}$ is the constraint language over F whose relations are given by a linear equation.

As the examples above indicate, in most cases our CSPs start off as problems over a single domain. However, solution algorithms often modify the domains of variables in different ways.

We illustrate the correspondence between the homomorphism definition of the CSP and Definition 2.1 with an example. Consider again the k-COLOURING problem, and let $\mathcal{H}_k$ denote the relational structure with universe $[k]$ over vocabulary $\{R_{\neq}\}$ and $R_{\neq}^{\mathcal{H}_k}$ is interpreted as the disequality relation. In other words, $\mathcal{H}_k = K_k$ is a complete graph with k vertices. Then a homomorphism from a given graph $G=(V,E)$ to K_k exists if and only if it is possible to assign vertices of K_k (colours) to vertices of G in such a way that for any $(u,v)\in E$ the vertices u and v are assigned different colours. The latter is just a proper k-colouring of G.

Using the homomorphism definition the k-COLOURING problem can be generalized to the H-COLOURING problem, where H is a graph or digraph: Given a (di)graph G, decide whether or not there is a homomorphism from G to H. Using the CSP notation the H-COLOURING problem is $\mathrm{CSP}(E_H)$, where E_H denotes the edge relation of H. The H-COLOURING problem has received much attention in graph theory, see, e.g. [Hell and Nešetřil 2004; Hell and Nešetřil 1990].

3. LOGIC AND CONSTRAINT PROPAGATION

3.1. Logic and Databases

The next step in the CSP research was motivated by its applications in the theory of relational databases. The QUERY EVALUATION problem can be thought of as deciding whether a first order sentence in the vocabulary of a database is true in that database (that is, whether or not the query has a non-empty answer). The QUERY CONTAINMENT problem asks, given two queries Φ and Ψ, whether the query $\Phi\to\Psi$ is true in all databases of the given vocabulary. The former problem is of course the main problem relational databases are needed for, while the latter is routinely used in various query optimization techniques. It turns out that both problems have intimate connections to the CSP, if the CSP is properly reformulated. We need some terminology from model theory.

A *vocabulary* is a finite set of relational symbols $R_1,\ldots,R_n$ each of which has a fixed arity $\mathrm{ar}(R_i)$. A *relational structure* over vocabulary $R_1,\ldots,R_n$ is a tuple $\mathcal{H} = (H; R_1^{\mathcal{H}},\ldots,R_n^{\mathcal{H}})$ such that H is a non-empty set, called the *universe* of $\mathcal{H}$, and each $R_i^{\mathcal{H}}$ is a relation over H having the same arity as the symbol R_i. A sentence is said to be a *conjunctive query* if it only uses existential quantifiers and its quantifier-free part is a conjunction of atomic formulas.

Definition 3.1. An instance of the CSP is a pair $(\Phi,\mathcal{H})$, where $\mathcal{H}$ is a relational structure in a certain vocabulary, and Φ is a conjunctive query in the same vocabulary. The objective is to decide whether Φ is true in $\mathcal{H}$.

To see that the definition above is equivalent to the previous two definitions of the CSP, we again consider its special case, k-COLOURING. The vocabulary corresponding to the problem contains just one binary predicate $R_{\neq}$. Let again $\mathcal{H}_k$ be the relational structure with universe $[k]$ in the vocabulary $\{R_{\neq}\}$, where $R_{\neq}^{\mathcal{H}_k}$ is interpreted as the disequality relation on the set $[k]$. Then an instance $G = (\{v_1, \dots, v_n\}, E)$ of k-COLOURING is equivalent to testing whether conjunctive query $\exists x_1, \dots, x_n \bigwedge_{(v_i,v_j)\in E} R_{\neq}(x_i, x_j)$ is true in $\mathcal{H}$.

Thus, the QUERY EVALUATION problem, when restricted to conjunctive queries, is just the CSP. A database is then considered as the input relational structure. The Chandra-Merlin Theorem [Chandra and Merlin 1977] shows that the QUERY CONTAINMENT problem is also equivalent to the CSP.

Relational database theory also massively contributed to the CSP research, most notably by techniques related to local propagation algorithms and the logic language Datalog that we will discuss next.

3.2. Local Propagation Algorithms

Constraint propagation algorithms are probably the most natural way to solve a CSP, and have been intensively studied and widely used in AI since the very beginning. There is a variety of such algorithms (see [Dechter 2003] to have some idea) differing in strength and running time, but we describe essentially one such algorithm, applicable whenever any other propagation algorithm solves the problem.

Let $R \subseteq A_1 \times \dots \times A_\ell$ be a relation, $\mathbf{a} \in A_1 \times \dots \times A_\ell$, and $J = \{i_1, \dots, i_k\} \subseteq [\ell]$. Let $\mathrm{pr}_J\mathbf{a} = (\mathbf{a}[i_1], \dots, \mathbf{a}[i_k])$ and $\mathrm{pr}_J R = \{\mathrm{pr}_J\mathbf{a} \colon \mathbf{a} \in R\}$, be the *projections* of $\mathbf{a}$ and R, respectively, on J. Often we will use sets of CSP variables to index entries of tuples and relations. Projections in this case are defined in a similar way. Let $\mathcal{I} = (V, \mathcal{C})$ be a CSP instance. For $W \subseteq V$ by $\mathcal{I}_W$ we denote the *restriction* of $\mathcal{I}$ onto W, that is, the instance $(W, \mathcal{C}_W)$, where for each $C = \langle \mathrm{s}, R\rangle \in \mathcal{C}$, the set $\mathcal{C}_W$ includes the constraint $C_W = \langle \mathrm{s} \cap W, \mathrm{pr}_{\mathbf{s}\cap W} R\rangle$. The set of solutions of $\mathcal{I}_W$ will be denoted by $\mathcal{S}_W$.

Unary solutions, that is, when $|W| = 1$ play a special role. As is easily seen, for $v \in V$ the set $\mathcal{S}_v$ is just the intersection of unary projections $\mathrm{pr}_v R$ of constraints whose scope contains v. Instance $\mathcal{I}$ is said to be *1-minimal* if for every $v \in V$ and every constraint $C = \langle \mathrm{s}, R\rangle \in \mathcal{C}$ such that $v \in \mathrm{s}$, it holds $\mathrm{pr}_v R = \mathcal{S}_v$. For a 1-minimal instance one may always assume that allowed values for a variable $v \in V$ is the set $\mathcal{S}_v$. We call this set the *domain* of v. The domain $\mathcal{S}_v$ may change as a result of transformations of the instance.

Instance $\mathcal{I}$ is said to be *(2,3)-minimal* if it satisfies the following condition:
– for every $X = \{u, v\} \subseteq V$, any $w \in V - X$, and any $(a, b) \in \mathcal{S}_X$, there is $c \in \mathcal{S}_w$ such that $(a, c) \in \mathcal{S}_{\{u,w\}}$ and $(b, c) \in \mathcal{S}_{\{v,w\}}$.

For $k \in \mathbb{N}$, $(k, k+1)$-minimality is defined in a similar way using $k, k+1$ instead of $2, 3$.

Instance $\mathcal{I}$ is said to be *minimal* (or *globally minimal*) if for every $C = \langle \mathrm{s}, R\rangle \in \mathcal{C}$ and every $\mathbf{a} \in R$ there is a solution φ such that $\varphi(\mathrm{s}) = \mathbf{a}$. Similarly, $\mathcal{I}$ is said to be *globally 1-minimal* if for every $v \in V$ and $a \in \mathcal{S}_v$ there is a solution φ with $\varphi(v) = a$. Clearly, establishing minimality amounts to solving the problem and so not always can be easily done.

Any instance can be transformed to a 1-minimal or (2,3)-minimal instance in polynomial time using the standard constraint propagation algorithms (see, e.g. [Dechter 2003]). These algorithms work by changing the constraint relations and the domains of the variables eliminating some tuples and elements from them.

If a constraint propagation algorithm solves a CSP, the problem is said to be of bounded width. More precisely, $\mathrm{CSP}(\Gamma)$ is said to have *bounded width* if for some k

every $(k, k+1)$-minimal instance from $\mathrm{CSP}(\Gamma)$ has a solution (we also say that $\mathrm{CSP}(\Gamma)$ has width k in this case).

Example 3.2. (1) The 2-SAT problem has bounded width, namely, width 2.

(2) The H-COLOURING problem has width 2 when graph H is bipartite, and is NP-complete otherwise.

(3) HORN-SAT is the SATISFIABILITY problem restricted to Horn clauses, i.e., clauses of the form $x_1 \wedge \cdots \wedge x_k \to y$. Let $\Gamma_{k\text{-HORN}}$ be the constraint language consisting of relations expressible by a Horn clause with at most k premises. The problem k-HORN-SAT is equivalent to $\mathrm{CSP}(\Gamma_{k\text{-HORN}})$ and has width k.

(4) The LIN problem provides an archetypical example of a CSP that does not have bounded width. Indeed, for any k it is not difficult to construct a system of linear equations that is inconsistent, but the $(k, k+1)$-minimality algorithm returns an instance with non-empty constraints.

Problems of bounded width are well studied, see the older survey [Bulatov et al. 2008] and more recent [Barto 2014a]. Barto and Kozik [Barto and Kozik 2014; Barto 2014a] and independently Bulatov [Bulatov 2016c] characterized languages Γ such that $\mathrm{CSP}(\Gamma)$ has bounded width. As this characterization involves some algebraic concepts, we return to it later.

3.3. Datalog and other equivalent characterizations of bounded width

Problems of bounded width have several equivalent characterizations.

Datalog. This simple language is related to the least fixed point operator in logic and uses the predicates of some relational structure as well as certain derived predicates. For example, let $G = (V, E)$ be a graph, and let us consider the following simple Datalog program

$$\begin{array}{lll} P(x,y) & \mid- & E(x,y) \\ P(x,y) & \mid- & P(x,z) \wedge E(z,t) \wedge E(t,y) \\ O & \mid- & P(x,x) \end{array}$$

Here, say, the second *rule* (line) means that predicate P has to be true on all pairs x, y, for which there are z, t satisfying the expression on the right hand side. It is easy to see that predicate O here becomes true if and only if G contains an odd cycle. In this sense the Datalog program above decides whether a given graph is NOT bipartite. For more basics of Datalog and its applications see [Kolaitis and Vardi 1995; Kolaitis 2007].

Homomorphism duality. Let $\mathcal{H}$ be a structure, a set $\mathcal{O}$ is said to be an *obstruction set* for $\mathcal{H}$ if for any structure $\mathcal{G}$, there is a homomorphism from $\mathcal{G}$ to $\mathcal{H}$ if and only if for no structure $\mathcal{G}' \in \mathcal{O}$ there is a homomorphism from $\mathcal{G}'$ to $\mathcal{G}$. For instance, the set of odd cycles is an obstruction set for any bipartite graph. Structure $\mathcal{H}$ is said to have *tree-width k duality* if there is an obstruction set $\mathcal{O}$ for $\mathcal{H}$ such that the tree-width of every structure of $\mathcal{O}$ is at most k. Thus, any bipartite graph has tree-width 2 duality. Homomorphism duality originates in graph theory [Hell and Nešetřil 2004], and has been used for CSP as well.

Pebble games. The *existential k-pebble game* has been introduced in [Kolaitis and Vardi 1995]. It is played on a pair of structures $\mathcal{G}$ and $\mathcal{H}$ by two players *Spoiler* and *Duplicator*. Spoiler has k pebbles and in every move she can place or remove a pebble on an element of $\mathcal{G}$. Duplicator has to respond by maintaining a partial homomorphism from $\mathcal{G}$ to $\mathcal{H}$. More precisely, if $I \subseteq \mathcal{G}$ is the current set of pebbled elements of $\mathcal{G}$ and

$\varphi : I \to \mathcal{H}$ is the current partial homomorphism, then, if Spoiler removes a pebble, Duplicator has to respond with the corresponding restriction of φ. If Spoiler adds a pebble, Duplicator has to respond with an extension of φ. Spoiler wins if at any point of time Duplicator is unable to respond. Duplicator wins if she has a strategy of keeping the game going forever.

It turns out that the three frameworks above lead to the same class(es) of the CSP.

THEOREM 3.3 ([KOLAITIS AND VARDI 1995; FEDER AND VARDI 1998]). *For a finite relational structure $\mathcal{H}$ and $k \in \mathbb{N}$ the following conditions are equivalent:*

(1) There is a Datalog program $\mathcal{P}$ whose rules use no more than k different variables that defines the class of structures not homomorphic to $\mathcal{H}$.
(2) $\mathcal{H}$ has tree-width k duality.
(3) A structure $\mathcal{G}$ is homomorphic to $\mathcal{H}$ if and only if Duplicator has a winning strategy in the k-pebble game on $\mathcal{G}, \mathcal{H}$.

Although CSPs of width k (as defined here) are not captured by the conditions of Theorem 3.3, those conditions capture bounded width.

THEOREM 3.4 ([BARTO AND KOZIK 2012; BARTO 2014A; BULATOV 2016C]). *For a finite relational structure $\mathcal{H}$ the following conditions are equivalent:*

(1) $\mathrm{CSP}(\mathcal{H})$ has width 2, that is, establishing (2,3)-minimality is a solution algorithm.
(2) $\mathrm{CSP}(\mathcal{H})$ has width ℓ for some $\ell \geq 2$, that is, $\mathrm{CSP}(\mathcal{H})$ has bounded width.
(3) there is k, for which the equivalent conditions of Theorem 3.3 hold.

From the practical perspective establishing (2,3)-minimality, and moreover $(k, k+1)$-minimality for $k > 2$ is very computationally demanding. It is therefore important to find the fastest propagation algorithm that nevertheless solves problems of bounded width. So far the fastest such algorithm was suggested by Kozik [Kozik 2016].

Note that if a relational structure $\mathcal{H}$ has infinite number of predicates or, equivalently, a constraint language is infinite, Theorems 3.3,3.4 are no longer true, and some of the conditions (such as definability by Datalog) are even not applicable. The concept of width and bounded width of a CSP nevertheless remains valid, as well as, the equivalence of conditions (1),(2) of Theorem 3.4.

4. ALGEBRAIC APPROACH

The most successful approach to tackling the Dichotomy Conjecture turned out to be the algebraic one. In this section we introduce the algebraic approach to the CSP and show how it can be used to determine the complexity of nonuniform CSPs. A keen reader can find more details on the algebraic approach, its applications, and the underlying algebraic facts from the following books [Grätzer 2008; Hobby and McKenzie 1988], surveys [Barto et al. 2017; Barto and Kozik 2017; Bulatov and Valeriote 2008; Bulatov et al. 2008], and research papers [Bulatov et al. 2005; Bulatov 2006b; 2011; 2016a; Bulatov and Dalmau 2006; Berman et al. 2010; Barto 2014a; Barto and Kozik 2014; 2012; Idziak et al. 2010].

4.1. Primitive Positive Definitions

Let Γ be a set of relations (predicates) over a finite set A. A relation R over A is said to be *primitive-positive (pp-) definable* in Γ if $R(\mathbf{x}) = \exists \mathbf{y}\ \Phi(\mathbf{x}, \mathbf{y})$, where Φ is a conjunction that involves predicates from Γ and equality relations. The formula above is then called a *pp-definition* of R in Γ. A constraint language Δ is pp-definable in Γ if so is every relation from Δ. In a similar way pp-definability can be introduced for relational structures.

Example 4.1. Let $K_3 = ([3], E)$ be a 3-element complete graph. Its edge relation is the binary disequality relation on $[3] = \{1, 2, 3\}$. Then the pp-formula

$$Q(x,y,z) \;=\; \exists t,u,v,w(E(t,x) \wedge E(t,y) \wedge E(t,z) \wedge E(u,v) \wedge E(v,w) \\ \wedge E(w,u) \wedge E(u,x) \wedge E(v,y) \wedge E(w,z))$$

defines the relation Q that consists of all triples containing exactly 2 different elements from $[3]$.

A link between pp-definitions and reducibility between nonuniform CSPs was first observed by Jeavons et al. in [Jeavons et al. 1997].

THEOREM 4.2 ([JEAVONS ET AL. 1997]). *Let Γ and Δ be constraint languages and Δ finite. If Δ is pp-definable in Γ then* $\mathrm{CSP}(\Delta)$ *is polynomial time reducible*[3] *to* $\mathrm{CSP}(\Gamma)$.

It was later shown that pp-definability in Theorem 4.2 can be replaced with a more general notion of *pp-constructability* [Barto et al. 2017; Barto et al. 2018].

4.2. Polymorphisms and Invariants

Primitive positive definability can be concisely characterized using polymorphisms. An operation $f : A^k \to A$ is said to be a *polymorphism* of a relation $R \subseteq A^n$ if for any $\mathbf{a}_1, \ldots, \mathbf{a}_k \in R$ the tuple $f(\mathbf{a}_1, \ldots, \mathbf{a}_k)$ also belongs to R, where $f(\mathbf{a}_1, \ldots, \mathbf{a}_k)$ stands for $(f(\mathbf{a}_1[1], \ldots, \mathbf{a}_k[1]), \ldots, f(\mathbf{a}_1[n], \ldots, \mathbf{a}_k[n]))$. Operation f is a polymorphism of a constraint language Γ if it is a polymorphism of every relation from Γ. Similarly, operation f is a polymorphism of a relational structure $\mathcal{H}$ if it is a polymorphism of every relation of $\mathcal{H}$. The set of all polymorphisms of language Γ or relational structure $\mathcal{H}$ is denoted by $\mathsf{Pol}(\Gamma)$, $\mathsf{Pol}(\mathcal{H})$. If F is a set of operations, $\mathsf{Inv}(F)$ denotes the set of all relations R such that every operation from F is a polymorphism of R.

Example 4.3. Let R be an affine relation, that is, R is the solution space of a system of linear equations over a field F. Then the operation $f(x,y,z) = x - y + z$ is a polymorphism of R. Indeed, let $A \cdot \mathbf{x} = \mathbf{b}$ be a system defining R, and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in R$. Then

$$A \cdot f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = A \cdot (\mathbf{x} - \mathbf{y} + \mathbf{z}) = A \cdot \mathbf{x} - A \cdot \mathbf{y} + A \cdot \mathbf{z} = \mathbf{b} - \mathbf{b} + \mathbf{b} = \mathbf{b}.$$

In fact, the converse can also be shown: if R is invariant under f, where f is defined in a certain finite field F then R is the solution space of some system of linear equations over F.

Several other useful polymorphisms are the following

Example 4.4 (*[Jeavons et al. 1997; Jeavons et al. 1998; Bulatov and Dalmau 2006; Maróti and McKenzie 20*
(1) A binary operation $\cdot$ on a set A is said to be a *semilattice operation* if it is (a) *idempotent*, $x \cdot x = x$; (b) *commutative*, $x \cdot y = y \cdot x$; and (c) *associative*, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, for any $x, y, z \in A$.

(2) A k-ary operation g on A is called a *near-unanimity* operation, or NU if

$$g(y, x, \ldots, x) = g(x, y, x, \ldots, x) = \cdots = g(x, \ldots, x, y) = x$$

for any $x, y \in A$. A ternary NU is also referred to as a *majority* operation.

(3) A k-ary operation g on A is called a *weak near-unanimity* operation, or WNU if it satisfies all the equations of an NU except for the last one

$$g(y, x, \ldots, x) = g(x, y, x, \ldots, x) = \cdots = g(x, \ldots, x, y).$$

[3] In fact, due to the result of [Reingold 2008] this reduction can be made log-space.

(4) A ternary operation h on A is called *Maltsev* if

$$h(x, y, y) = h(y, y, x) = x$$

for any $x, y \in A$. As we saw in Example 4.3 any structure whose relations can be represented by linear equations has the Maltsev polymorphism $x - y + z$ where $+$ and $-$ are the operations of the underlying field.

A link between polymorphisms and pp-definability of relations is given by the *Galois connection*.

THEOREM 4.5 (GALOIS CONNECTION, [BODNARCHUK ET AL. 1969; GEIGER 1968]). *Let Γ be a constraint language on A, and let $R \subseteq A^n$ be a non-empty relation. Then R is preserved by all polymorphisms of Γ if and only if R is pp-definable in Γ.*

Theorems 4.2 and 4.5 together imply that the complexity of $\mathrm{CSP}(\Gamma)$ depends entirely on $\mathsf{Pol}(\Gamma)$. This was used by Jeavons and coauthors in the early papers on the algebraic approach to show the tractability and hardness of certain classes of the CSP.

Example 4.6. (A) If a constraint language Γ has a semilattice or NU polymorphism, then $\mathrm{CSP}(\Gamma)$ can be solved in polynomial time, [Jeavons et al. 1997; Jeavons et al. 1998].

(B) If Γ has a Mal'tsev polymorphism h, then $\mathrm{CSP}(\Gamma)$ can be solved in polynomial time, [Bulatov 2002b; Bulatov and Dalmau 2006]. In certain special cases — when h is an affine operation or the operation $xy^{-1}z$ of a finite group — this result can be traced back to [Feder and Vardi 1998; Jeavons et al. 1997]. However, the general result turns out to be much more difficult than those from the previous item.

(C) If every polymorphism f of a constraint language Γ on a set A is such that $f(x_1, \dots, x_n) = x_i$ for some i and all $x_1, \dots, x_n \in A$, then $\mathrm{CSP}(\Gamma)$ is NP-complete [Jeavons et al. 1997].

(D) Schaefer's Theorem [Schaefer 1978] can be stated in terms of polymorphisms. Let Γ be a constraint language on a 2-element set (we assume this set to be $\{0, 1\}$). The problem $\mathrm{CSP}(\Gamma)$ is solvable in polynomial time if and only if one of the following operations is a polymorphism of Γ: the constant operations 0 or 1, the semilattice operations of conjunction and disjunction, the majority operation on $\{0, 1\}$ (there is only one such operation), or the Maltsev operation $x - y + z$ where $+$ and $-$ are modulo 2. Otherwise $\mathrm{CSP}(\Gamma)$ is NP-complete.

4.3. Algebras and the CSP

Recall that a (*universal*) *algebra* is an ordered pair $\mathbb{A} = (A, F)$ where A is a non-empty set, called the *universe* of $\mathbb{A}$, and F is a family of finitary operations on A, called the *basic operations* of $\mathbb{A}$. Operations that can be obtained from F by means of composition are said to be *term* operations of the algebra. Every constraint language on a set A can be associated with an algebra $\mathsf{Alg}(\Gamma) = (A, \mathsf{Pol}(\Gamma))$. In a similar way any relational structure $\mathcal{A}$ (with universe A) can be paired up with the algebra $\mathsf{Alg}(\mathcal{A}) = (A, \mathsf{Pol}(\mathcal{A}))$. On the other hand, an algebra $\mathbb{A} = (A, F)$, can be associated with the constraint language $\mathsf{Inv}(F)$ or the class $\mathsf{Str}(\mathbb{A})$ of structures $\mathcal{A} = (A, R_1, \dots, R_k)$ such that $R_1, \dots, R_k \in \mathsf{Inv}(F)$.

This correspondence can be extended to CSPs: For an algebra $\mathbb{A}$ by $\mathrm{CSP}(\mathbb{A})$ we denote the class of problems $\mathrm{CSP}(\mathcal{A})$, $\mathcal{A} \in \mathsf{Str}(\mathbb{A})$. Equivalently, $\mathrm{CSP}(\mathbb{A})$ can be thought of as $\mathrm{CSP}(\mathsf{Inv}(F))$ for the infinite constraint language $\mathsf{Inv}(F)$. Note, however, that there is a subtle difference in the notion of polynomial time solvability in these two cases that we will address next.

We say that algebra $\mathbb{A}$ is *tractable* if every $\mathrm{CSP}(\mathcal{A})$, $\mathcal{A} \in \mathsf{Str}(\mathbb{A})$, is solvable in polynomial time. Observe that this does not guarantee that there is a single solution algorithm for all such problems, nor does it guarantee that there is any uniformity among those algorithms. In general, it may be possible that for a tractable algebra $\mathbb{A} = (A, F)$ the problem $\mathrm{CSP}(\mathsf{Inv}(F))$ is NP-hard. If the problem $\mathrm{CSP}(\mathsf{Inv}(F))$ is solvable in polynomial time, we call $\mathbb{A}$ *globally tractable*. Algebra $\mathbb{A}$ is called *NP-complete* if some $\mathrm{CSP}(\mathcal{A})$, $\mathcal{A} \in \mathsf{Str}(\mathbb{A})$, is NP-complete. Algebra $\mathbb{A}$ is *globally NP-complete* if $\mathrm{CSP}(\mathsf{Inv}(F))$ is NP-complete.

Using the algebraic terminology we can pose a stronger version of the Dichotomy Conjecture.

CONJECTURE 4.7 (DICHOTOMY CONJECTURE+). *Every finite algebra either is globally tractable or is NP-complete (in the local sense).*

The Dichotomy Conjecture+ can be made more precise by making use of weak near-unanimity terms. An operation f on a set A is said to be *idempotent* if the equality $f(x, \dots, x) = x$ holds for all $x \in A$. An algebra all of whose term operations are idempotent is said to be *idempotent*.

THEOREM 4.8 ([BULATOV ET AL. 2005]). *For any finite algebra $\mathbb{A}$ there is an idempotent finite algebra $\mathbb{B}$ such that:*

– *$\mathbb{A}$ is globally tractable if and only if $\mathbb{B}$ is globally tractable;*

– *$\mathbb{A}$ is NP-complete if and only if $\mathbb{B}$ is NP-complete.*

Theorem 4.8 reduces the Dichotomy Conjecture+ to idempotent algebras.

CONJECTURE 4.9. *If a relational structure $\mathcal{A}$ is such that $\mathsf{Alg}(\mathcal{A})$ is idempotent, then $\mathrm{CSP}(\mathcal{A})$ is solvable in polynomial time if and only if $\mathcal{A}$ admits a weak near-unanimity polymorphism. Otherwise it is NP-complete.*

Or in the stronger algebraic version

CONJECTURE 4.10 (DICHOTOMY CONJECTURE ++). *An idempotent algebra $\mathbb{A}$ is globally tractable if and only if it has a weak near-unanimity term operation. Otherwise it is NP-complete.*

By the results of [Maróti and McKenzie 2008] the Dichotomy Conjecture ++ is equivalent to the conjecture stated in [Bulatov et al. 2005].

5. THE PURSUIT OF THE DICHOTOMY CONJECTURE

The complexity of the CSP including the dichotomy conjecture has been intensively studied for 40 years. Here we outline the history of this area that is related to the dichotomy conjecture.

Early dichotomies. Schaefer [Schaefer 1978] obtained the first dichotomy theorem on the CSP, long before the Dichotomy Conjecture was proposed. His classification of constraint languages on a 2-element set can be easily extended to 2-element algebras. Then it claims that an idempotent 2-element algebra is globally tractable if and only if it has one of the following term operations: a semilattice operation, a majority operation, or the affine operation $x - y + z$. By [Post 1941] this is equivalent to having a weak near-unanimity term operation. Another early dichotomy result by Hell and Nesetril [Hell and Nešetřil 1990] gives a classification of (undirected) graphs H with respect to the complexity of the H-COLOURING problem: such a problem is polynomial time solvable if H is bipartite or contains a loop, and NP-complete otherwise. Let H be a graph, $\mathbb{A} = \mathsf{Alg}(H)$, and let $\mathbb{B}$ be the idempotent algebra constructed from $\mathbb{A}$ as in Theorem 4.8. If H is bipartite then $\mathbb{B}$ is 2-element and has a majority term operation.

Otherwise $\mathbb{B}$ does not have a WNU [Bulatov 2005]. Thus the classification from [Hell and Nešetřil 1990] matches the Dichotomy Conjecture++.

The two algorithms. Apart from posing the Dichotomy Conjecture Feder and Vardi [Feder and Vardi 1998] made several important observations. One of them is a clear distinction between two types of CSP algorithms. One might expect that problems as diverse as nonuniform CSPs would have a variety of different solution algorithms. This, however, is not the case, and all known (at that point) algorithms are variations of just two types of algorithms. Algorithms of the first type include all local propagation algorithms as described in Section 3.2. Feder and Vardi conjectured that the solvability of $\mathrm{CSP}(\mathcal{H})$ by a local propagation algorithm is related to a property they called the 'ability to count'. More precisely, they conjectured that a propagation algorithm solves $\mathrm{CSP}(\mathcal{H})$ if and only if $\mathcal{H}$ does not have the ability to count. This conjecture has been confirmed much later in [Larose et al. 2009] (if combined with the characterization of CSPs of bounded width [Barto and Kozik 2014; Bulatov 2016c]). Algorithms of the second type use variations of Gaussian Elimination or group theoretic approaches such as Furst's algorithm for coset generation [Furst et al. 1980] which is used in [Feder and Vardi 1998].

Algebraic approach, polymorphisms. The discovery by Jeavons et al. [Jeavons et al. 1997; Jeavons et al. 1998] of the connection between polymorphisms and the complexity of the CSP allowed, firstly, to greater unify approaches to different constraint languages. It turned out, many of them have polymorphisms possessing similar properties and so could be handled in similar ways. In particular, the fact that all the assorted constraint languages known to have bounded width actually have this property can be explained by polymorphisms of just two types [Jeavons et al. 1998]: NU and semilattice operations. In the former case bounded width follows from the decomposition theorem by Pixley [Pixley 1979], while in the latter case bounded width is an easy implication of the structure of semilattices. Similarly, in nearly all other cases the tractability of a CSP could be explained by the existence of a group Mal'tsev polymorphism $xy^{-1}z$ or $x - y + z$. For a recent survey of the algebraic approach see [Barto et al. 2017].

Algebraic approach: algebras and varieties. Extending the algebraic approach from polymorphisms to algebras and varieties of algebras [Bulatov et al. 2005; Bulatov and Jeavons 2001; 2003] contributed to the study of the CSP in two ways. Firstly, it demonstrated that what is important for the complexity of CSPs is not particular polymorphisms, but the identities or equations they satisfy (cf. Example 4.4). Secondly, it allowed to employ various structural theories of universal algebras. In particular, it made possible dichotomy results on arbitrary CSPs on small domains, they all agree with the Dichotomy Conjecture++, [Bulatov 2002a; 2006b] (3-element domains), [Marković 2011] (4-element domains), [Zhuk 2016b; 2016a] (5- and 7-element domains). Although still somewhat ad-hoc and based on case analysis, such results had been inaccessible by previous methods for more than 20 years. This structural approach also made it possible to design algorithms for several types of polymorphisms: Mal'tsev polymorphisms, see Example 4.4(4) ([Bulatov 2002b; Bulatov and Dalmau 2006] subsequently generalized in [Dalmau 2006]), and 2-semilattice polymorphisms satisfying the identities $xx = x$, $xy = yx$, $x(yx) = (xy)x$, [Bulatov 2006a]. Another direction that became possible due to the algebraic approach is the finer classification of complexity of the CSP. Initiated in [Allender et al. 2009] it was later developed in [Larose and Tesson 2009], where a conjecture was posed on such a classification using the language of omitting types in the sense of tame congruence theory [Hobby and McKenzie 1988], and then in [Larose et al. 2007; Kazda 2018].

Absorption and bounded width. Interaction with the CSP research catalyzed the development of universal algebra as well, since new tools were needed. One of the major developments was the concept of absorption and related techniques introduced by Barto and Kozik [Barto and Kozik 2012; Barto 2014b], see also [Barto and Kozik 2017] for a recent survey. Fundamental technical lemmas obtained in those papers, such as the Loop Lemma and the Rectangularity Lemma, allowed them to prove a number of major results in universal algebra. For CSP they led to a dichotomy theorem for digraphs without sources and sinks [Barto et al. 2009] and some other classes of digraphs [Barto et al. 2009; Barto and Bulin 2013]. The most important result obtained using this technique is the characterization of problems of bounded width.

THEOREM 5.1 ([BARTO 2014A; BULATOV 2016C; 2004; KOZIK ET AL. 2015]). *For an idempotent algebra $\mathbb{A}$ the following are equivalent:*

(1) $\mathrm{CSP}(\mathbb{A})$ *has bounded width;*

(2) every (2,3)-minimal instance from $\mathrm{CSP}(\mathbb{A})$ *has a solution;*

(3) $\mathbb{A}$ *has a weak near-unanimity term operation of arity k for every $k \geq 3$;*

(4) every quotient algebra of a subalgebra of $\mathbb{A}$ *has a nontrivial operation, and none of them is equivalent to a module (in a certain precise sense).*

Few subpowers algorithm. The second type of algorithms identified in [Feder and Vardi 1998] is based on group theoretic tools and has been generalized to problems with a Mal'tsev polymorphism [Bulatov 2002b; Bulatov and Dalmau 2006], and then to problems with a Generalized Majority-Minority polymorphism [Dalmau 2006]. The common feature of all those algorithms was that similar to Gaussian Elimination they construct some sort of a compact basis of the set of solutions. Such a basis may not exist in the general case

It is thought that the property of relations to have a compact representation, where compactness is understood as having size polynomial in the arity of the relation, is the right generalization of linear algebra problems where Gaussian Elimination can be used. Let $\mathbb{A} = (A, F)$ be an algebra. It is said to be an *algebra with few subpowers* if every relation over A invariant under F admits a compact representation [Berman et al. 2010; Idziak et al. 2010]. The term few subpowers comes from the observation that every relation invariant under F is a subalgebra of a direct power of $\mathbb{A}$, and if the size of compact representation is bounded by a polynomial $p(n)$ then at most $2^{p(n)}$ n-ary relations can be represented, while the total number of such relations can be as large as $2^{|A|^n}$. Algebras with few subpowers are completely characterized by Idziak et al. [Berman et al. 2010; Idziak et al. 2010]. A minor generalization of the algorithm from [Dalmau 2006] solves $\mathrm{CSP}(\mathbb{A})$, where $\mathbb{A}$ has few subpowers.

Conservative CSPs and graphs of algebras. An important general class of constraint languages where a dichotomy theorem could be obtained by more or less ad-hoc methods is the class of conservative languages: A language Γ over a set A is said to be *conservative* if every subset of A is a (unary) relation in Γ. In terms of the CSP it means that in an instance the set of possible values of each variable can be arbitrarily restricted. Similar problems have been studied within the graph homomorphism community, where they are called LIST HOMOMORPHISM PROBLEMS (as every vertex of the source graph is equipped with a list of possible images), see [Hell and Nešetřil 2004; Feder and Hell 1998; Feder et al. 1999]. As is shown in [Bulatov 2011; 2003] the Dichotomy Conjecture++ holds for such constraint languages. This result was later greatly simplified in [Barto 2011; Bulatov 2016a]. The main approach used in these proofs (except [Barto 2011] that is based on absorption) has later proved to be the key to resolving the dichotomy conjecture. For any polynomial time solvable conservative

Γ on a set A the problem $\mathrm{CSP}(\Gamma)$ restricted to a 2-element subset of A is equivalent to one of Schaefer's cases and therefore has one of the good polymorphisms: semilattice, majority, or affine. Thus within this approach 2-element subsets of A are assigned types that depend on which good polymorphism occurs in the restricted problem, converting A into an edge-coloured graph. This approach has been further generalized for arbitrary constraint languages (algebras) in [Bulatov 2004; 2016b; 2016c].

Hybrid algorithms. CSPs solvable by 'pure' constraint propagation and few subpowers algorithms are fully characterized, see above, and fall way short of the CSPs that are conjectured to be polynomial time solvable according to the dichotomy conjecture. Designing 'hybrid' algorithms has therefore been the crucial problem in resolving the conjecture. Apart from the ad-hoc hybrid algorithms used for CSPs on small domains [Bulatov 2006b; Marković 2011; Zhuk 2016b; 2016a] and conservative CSPs [Bulatov 2011; 2003; Barto 2011; Bulatov 2016a], Maroti was the first who explicitly posed the problem of combining different algorithmic techniques for the CSP. Recall that a *congruence* of an algebra $\mathbb{A}$ is an equivalence relation θ invariant with respect to the operations of $\mathbb{A}$. This way one may consider the *quotient algebra* $\mathbb{A}/_\theta$, whose elements are the equivalence classes of θ. Maroti attempted to prove the tractability of $\mathrm{CSP}(\mathbb{A})$, in which algebra $\mathbb{A}$ has a congruence θ such that $\mathbb{A}/_\theta$ is an algebra with a Mal'tsev term, and every block of θ satisfies the condition of Theorem 5.1, that is, gives rise to a CSP of bounded width; or the other way round $\mathbb{A}/_\theta$ has bounded width, while every θ-block is Mal'tsev. He managed to design a hybrid algorithm for the former case [Maróti 2011a] and to make significant progress towards resolving the latter case [Maróti 2011b]. This latter case however turned out to be the crux in proving the dichotomy conjecture [Bulatov 2017a].

Dichotomy theorems. The dichotomy conjecture was settled independently and almost at the same time by the author [Bulatov 2017c; 2017b] (the Dichotomy Conjecture++), and Zhuk [Zhuk 2017a; 2017b] (Conjecture 4.9) and [Zhuk 2018] (the Dichotomy Conjecture++). The first algorithm, [Bulatov 2017c; 2017b] is based on the local structure of algebras as introduced in [Bulatov 2004; 2016b; 2016c] and a new notion of minimality. The second algorithm uses a totally different approach which involves an intricate combination of constraint propagation techniques and then approximating a solution using systems of linear equations. In this paper we give an outline of the first algorithm [Bulatov 2017c; 2017b].

6. THE ALGORITHM

We now outline the algorithm resolving the Dichotomy Conjecture. The main approach will be to introduce a more general minimality notion (not local anymore) that allows us to solve problems beyond bounded width, and then to reduce the general CSP to such instances. A more detailed description along with some simple examples can be found in [Bulatov 2017b; 2018].

6.1. Algorithm ingredients

Gaussian Elimination and Few Subpowers. The main routine of the algorithm is removing semilattice edges. Let $\mathbb{A} = (A, F)$ be an idempotent algebra. A pair of elements $a, b \in \mathbb{A}$ is said to be a *semilattice edge* if there is a binary term operation f of $\mathbb{A}$ such that $f(a, a) = a$ and $f(a, b) = f(b, a) = f(b, b) = b$, that is, f is a semilattice operation on $\{a, b\}$. We say that algebra $\mathbb{A}$ is *semilattice free* if it does not contain semilattice edges. Removing semilattice edges is useful because of the following

PROPOSITION 6.1 ([BULATOV 2016C]). *If an idempotent algebra $\mathbb{A}$ is semilattice free, then it has few subpowers, and therefore $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time.*

Quasi-Centralizers. Quasi-centralizer is an operator on the congruences of an algebra. It is similar to the centralizer as it is defined in commutator theory [Freese and McKenzie 1987], albeit the exact relationship between the two concepts is not quite clear, and so we name it differently for safety.

The set of all congruences of an algebra $\mathbb{A}$ is denoted by $\mathsf{Con}(\mathbb{A})$. For an algebra $\mathbb{A}$, a term operation $f(x, y_1, \dots, y_k)$, and $\mathbf{a} \in \mathbb{A}^k$, let $f^{\mathbf{a}}(x) = f(x, \mathbf{a})$; it is a unary *polynomial* of $\mathbb{A}$. Let $\alpha, \beta \in \mathsf{Con}(\mathbb{A})$, and let $\zeta(\alpha, \beta) \subseteq \mathbb{A}^2$ denote the following binary relation: $(a, b) \in \zeta(\alpha, \beta)$ if and only if, for any term operation $f(x, y_1, \dots, y_k)$, any $i \in [k]$, and any $\mathbf{a}, \mathbf{b} \in \mathbb{A}^k$ such that $\mathbf{a}[i] = a$, $\mathbf{b}[i] = b$, and $\mathbf{a}[j] = \mathbf{b}[j]$ for $j \neq i$, it holds $f^{\mathbf{a}}(\beta) \subseteq \alpha$ if and only if $f^{\mathbf{b}}(\beta) \subseteq \alpha$. (Polynomials of the form $f^{\mathbf{a}}, f^{\mathbf{b}}$ are sometimes called *twin* polynomials.) The relation $\zeta(\alpha, \beta)$ is always a congruence of $\mathbb{A}$.

Propagation and algebras. Let $\mathbb{A} = (A, F)$ be an algebra. As is mentioned in Section 3.2 applying propagation and other algorithms to an instance $\mathcal{I}$ of $\mathrm{CSP}(\mathbb{A})$ may change the domain of the variables of $\mathcal{I}$. However, it is well known that these new domains remain *subalgebras* of $\mathbb{A}$, that is, operations from F act on the new domains as well. The domain of a variable v will be denoted $\mathbb{A}_v$. Transformations used in our algorithm can also change domain $\mathbb{A}_v$ to a quotient algebra $\mathbb{A}_v/_\theta$. Thus, we will consider constraint relations as subsets of the product of different algebras: $R \subseteq \mathbb{A}_{v_1} \times \dots \times \mathbb{A}_{v_k}$; however, the operations of $\mathbb{A}$ can still be used on these algebras.

Decomposition of CSPs. Let R be a binary relation, a subset of the product of $\mathbb{A} \times \mathbb{B}$, and $\alpha \in \mathsf{Con}(\mathbb{A})$, $\gamma \in \mathsf{Con}(\mathbb{B})$. Relation R is said to be *$\alpha\gamma$-aligned* if, for any $(a, c), (b, d) \in R$, $(a, b) \in \alpha$ if and only if $(c, d) \in \gamma$. This means that if $A_1, \dots, A_k$ are the α-blocks of $\mathbb{A}$, then there are also k γ-blocks of $\mathbb{B}$ and they can be labeled $B_1, \dots, B_k$ in such a way that

$$R = (R \cap (A_1 \times B_1)) \cup \dots \cup (R \cap (A_k \times B_k)).$$

Let $\mathcal{I} = (V, \mathcal{C})$ be a (2,3)-minimal instance from $\mathrm{CSP}(\mathbb{A})$. We will always assume that a (2,3)-minimal instance has a constraint $C^X = \langle X, R^X \rangle$ for every $X \subseteq V$, $|X| = 2$, where $R^X = \mathcal{S}_X$. Recall that $\mathbb{A}_v$ denotes the domain of $v \in V$. A set $W \subseteq V$ is said to be a *strand* if it is maximal (under inclusion) among the sets with the following property: There are $\alpha_v \in \mathsf{Con}(\mathbb{A}_v)$, $v \in W$, such that $R^{\{v,w\}}$ is $\alpha_v\alpha_w$-aligned. Thus for a strand W there is a one-to-one correspondence between α_v- and α_w-blocks of $\mathbb{A}_v$ and $\mathbb{A}_w$, $v, w \in W$. Moreover, by (2,3)-minimality these correspondences are consistent, that is, if $u, v, w \in W$ and B_u, B_v, B_w are α_u-, α_v- and α_w-blocks, respectively, such that $R^{\{u,v\}} \cap (B_u \times B_v) \neq \varnothing$ and $R^{\{v,w\}} \cap (B_v \times B_w) \neq \varnothing$, then $R^{\{u,w\}} \cap (B_u \times B_w) \neq \varnothing$. This means that $\mathcal{I}_W$ can be split into several instances, whose domains are α_v-blocks.

LEMMA 6.2. *Let $\mathcal{I}, W, \alpha_v$ for each $v \in W$, be as above. Then $\mathcal{I}_W$ can be decomposed into a collection of instances $\mathcal{I}_1, \dots, \mathcal{I}_k$, k constant, $\mathcal{I}_i = (W, \mathcal{C}_i)$ such that every solution of $\mathcal{I}_W$ is a solution of one of the $\mathcal{I}_i$ and for every $v \in W$ its domain in $\mathcal{I}_i$ is an α_v-block.*

Subdirectly irreducible algebras. In order to formulate the algorithm properly we need one more transformation of algebras. An algebra $\mathbb{A}$ is said to be *subdirectly irreducible* if the intersection of all its nontrivial (different from the equality relation) congruences is nontrivial. This smallest nontrivial congruence $\mu_{\mathbb{A}}$ is called the *monolith* of $\mathbb{A}$. It is a folklore observation that any CSP instance can be transformed in polynomial time to an instance, in which the domain of every variable is a subdirectly irreducible algebra. We will assume this property of all the instances we consider.

6.2. Block-Minimality

The notion of alignment allows for a new type of minimality of a CSP instance, block-minimality, which is key for our algorithm. In a certain sense it is similar to the standard local minimality, as it is also defined through a family of relations that have to be consistent in a certain way. However, block-minimality is not local, and is more difficult to establish, as it involves solving smaller CSP instances recursively. The definitions below are designed to allow for an efficient procedure to establish block-minimality.

Let $\mathcal{I} = (V, \mathcal{C}) \in \mathrm{CSP}(\mathbb{A})$ and α_v be a congruence of $\mathbb{A}_v$ for $v \in V$. By $\mathcal{I}/_{\overline{\alpha}}$ we denote the instance $(V, \mathcal{C}_{\overline{\alpha}})$ constructed as follows: the domain of $v \in V$ is $\mathbb{A}_v/_{\alpha_v}$; for every constraint $C = \langle \mathrm{s}, R\rangle \in \mathcal{C}$, $\mathrm{s} = (v_1, \dots, v_k)$, the set $\mathcal{C}_{\overline{\alpha}}$ includes the constraint $\langle \mathrm{s}, R/_{\overline{\alpha}}\rangle$, where $R/_{\overline{\alpha}} = \{(\mathbf{a}[v_1]^{\alpha_{v_1}}, \dots, \mathbf{a}[v_k]^{\alpha_{v_k}}) \mid \mathbf{a} \in R\}$.

Let $\mathsf{size}(\mathcal{I})$ denote the maximal size of domains of $\mathcal{I}$ that are not semilattice free and $\mathsf{MAX}(\mathcal{I})$ be the set of variables $v \in V$ with $|\mathbb{A}_v| = \mathsf{size}(\mathcal{I})$ and $\mathbb{A}_v$ is not semilattice free. For instances $\mathcal{I}, \mathcal{I}'$ we say that $\mathcal{I}'$ is *strictly smaller* than $\mathcal{I}$ if $\mathsf{size}(\mathcal{I}') < \mathsf{size}(\mathcal{I})$. For $Y \subseteq V$ let $\mu_v^Y = \mu_v$ if $v \in Y$ and $\mu_v^Y = \Delta_v$ otherwise, where Δ_v denotes the equality relation on $\mathbb{A}_v$.

Instance $\mathcal{I}$ is said to be *block-minimal* if for every strand W the following condition hold:

(BM) the problem $\mathcal{I}/_{\overline{\mu}^Y}$, where $Y = \mathsf{MAX}(\mathcal{I}) - W$, is minimal.

Next we observe that establishing block-minimality can be efficiently reduced to solving a polynomial number of strictly smaller instances. Let W be a strand and $Y = \mathsf{MAX}(\mathcal{I}) - W$. Then there are congruences α_v, $v \in W$, such that for any $v, w \in W$ the relation $R^{\{v,w\}}$ is $\alpha_v\alpha_w$-aligned. This means that every $\mathbb{A}_v$, $v \in W$, has the same number of α_v-blocks, let them be $B_v^1, \dots, B_v^k$, and $R^{\{v,w\}} \subseteq \bigcup_{i=1}^k B_v^i \times B_w^i$. By Lemma 6.2 the problem $\mathcal{I}_W$ can be decomposed into a collection $\mathcal{I}_1, \dots, \mathcal{I}_k$ of instances such that the domain of $v \in W$ in $\mathcal{I}_i$ is B_v^i. Now we restrict the domains of $v \in W$ in $\mathcal{I}/_{\overline{\mu}^Y}$ in the same way, that is, let $\mathcal{I}^i/_{\overline{\mu}^Y}$ is obtained from $\mathcal{I}/_{\overline{\mu}^Y}$ by setting the domain of $v \in W$ to be B_v^i and keeping the domains of the remaining variables. Note that every solution of $\mathcal{I}/_{\overline{\mu}^Y}$ is a solution of one of $\mathcal{I}^i/_{\overline{\mu}^Y}$. It is also not hard to see that $\mathsf{size}(\mathcal{I}^i/_{\overline{\mu}^Y}) < \mathsf{size}(\mathcal{I})$. Indeed, for every $v \in V$ either $|\mathbb{A}_v| < \mathsf{size}(\mathcal{I})$ if $v \notin \mathsf{MAX}(\mathcal{I})$, or $|\mathbb{A}_v/_{\mu_v}| < \mathsf{size}(\mathcal{I})$ if $v \in Y$, or $|B_v^i| < \mathsf{size}(\mathcal{I})$ if $v \in W$. Establishing the minimality of $\mathcal{I}/_{\overline{\mu}^Y}$ can be reduced to solving strictly smaller problems.

LEMMA 6.3. *Let $\mathcal{I} = (V, \mathcal{C})$ be a (2,3)-minimal instance. Then by solving a linear number of strictly smaller CSPs, $\mathcal{I}$ can be transformed to an equivalent block-minimal instance $\mathcal{I}'$.*

6.3. The Algorithm

In the algorithm we distinguish three cases depending on the presence of semilattice edges and quasi-centralizers of the domains of variables. In each case we employ different methods of solving or reducing the instance to a strictly smaller one.

Let $\mathcal{I} = (V, \mathcal{C})$ be a subdirectly irreducible, (2,3)-minimal instance. Let $\mathsf{Center}(\mathcal{I})$ denote the set of variables $v \in V$ such that $\zeta(\Delta_v, \mu_v)$ is the full relation. Let $\mu_v^* = \mu_v$ if $v \in \mathsf{MAX}(\mathcal{I}) \cap \mathsf{Center}(\mathcal{I})$ and $\mu_v^* = \Delta_v$ otherwise.

Semilattice Free Domains. If no domain of $\mathcal{I}$ contains a semilattice edge then by Proposition 6.1 $\mathcal{I}$ can be solved in polynomial time, using the few subpowers algorithm, as shown in [Idziak et al. 2010; Bulatov 2016c].

Small Centralizers. If $\mu_v^* = \Delta_v$ for all $v \in V$, block-minimality guarantees the existence of a solution, as Theorem 6.4 shows, and we can use Lemma 6.3 to solve the instance.

THEOREM 6.4. *If $\mathcal{I}$ is subdirectly irreducible, (2,3)-minimal, block-minimal, and $\mathsf{MAX}(\mathcal{I}) \cap \mathsf{Center}(\mathcal{I}) = \varnothing$, then $\mathcal{I}$ has a solution.*

Large Centralizers. Suppose that $\mathsf{MAX}(\mathcal{I}) \cap \mathsf{Center}(\mathcal{I}) \neq \emptyset$. In this case the algorithm proceeds in three steps.

Step 1. Consider the problem $\mathcal{I}/_{\overline{\mu}^*}$. We establish the global 1-minimality of this problem. If it is changed in the process, we start solving the new problem from scratch. Checking global 1-minimality can be reduced using standard techniques to solving a linear number of problems that are either strictly smaller, or have small centralizers (see above).

Step 2. For every $v \in \mathsf{Center}(\mathcal{I})$ we find a solution φ of $\mathcal{I}/_{\overline{\mu}^*}$ satisfying the following condition: there is $a \in \mathbb{A}_v$ such that $\{a, \varphi(v)\}$ is a semilattice edge if $\mu_v^* = \Delta_v$, or, if $\mu_v^* = \mu_v$, there is $b \in \varphi(v)$ such that $\{a, b\}$ is a semilattice edge. Such a solution exists since $\mathcal{I}/_{\overline{\mu}^*}$ is globally 1-minimal.

Step 3. Using the solutions found in Step 2 apply the transformation of $\mathcal{I}$ suggested by Maroti in [Maróti 2011b]. This transformation results in an equivalent instance, but eliminates from the respective domains the 'lower' ends a of semilattice edges chosen in Step 2. Thus the resulting instance is strictly smaller.

Using Lemma 6.3 and Theorems 6.4 it is not difficult to see that the algorithm runs in polynomial time. Indeed, every time it makes a recursive call it calls on a problem whose non-semilattice free domains of maximal cardinality have strictly smaller size, and therefore the depth of recursion is bounded by $|\mathbb{A}|$ if we are dealing with $\mathrm{CSP}(\mathbb{A})$. More precisely,

THEOREM 6.5. *If $\mathbb{A}$ is a finite idempotent algebra with a weak near-unanimity term. Then $\mathrm{CSP}(\mathbb{A})$ can be solved in time $O(Nmn^{|\mathbb{A}|+k})$, where n is the number of variables in an instance, m the number of constraints, N is the total number of tuples in all the constraint relations, and k is a constant such that the CSP over algebras with few subpowers derived from $\mathbb{A}$ can be solved in time $O(mn^k)$.*

7. FUTURE DIRECTIONS

We conclude the column with a short review of open questions related to the dichotomy conjecture, related areas and potential future directions.

Polymorphism oblivious algorithms. There is a peculiar asymmetry between the two main types of CSP algorithms, constraint propagation and the few subpowers algorithm. While constraint propagation can be run on any given instance without any prior knowledge about the underlying constraint language or algebra (although also without any guarantees to solve the problem), the few subpowers algorithm explicitly uses the polymorphisms associated with the problem. Both general algorithms for the CSP also use the knowledge of the algebraic structure of the problem. It is therefore an important question whether or not there exists an algorithm that solves, say, few subpowers CSPs without knowing any polymorphisms of the constraint language, but only certain local properties of the relations involved.

This question has a connection to the problem of recognizing, given a relational structure $\mathcal{H}$ or an algebra $\mathbb{A}$, if the problem $\mathrm{CSP}(\mathcal{H})$ or $\mathrm{CSP}(\mathbb{A})$ can be solved in polynomial time or has bounded width, or is within some other complexity class. This problem is known as the *metaproblem*, see [Chen and Larose 2017; Freese and Valeriote 2009]. Chen and Larose in [Chen and Larose 2017] observed that if a class of CSPs has such a polymorphism oblivious algorithm, then the metaproblem for this class can be solved in polynomial time (assuming the structures involved are cores and algebras are idempotent). In particular, the metaproblem for the class of structures of bounded width is polynomial time, while for the class of structures with tractable CSP the complexity of the metaproblem is unknown.

Other complexity classes. There is strong evidence that nonuniform CSPs can be complete in very few complexity classes. In [Allender et al. 2009] Allender et al. showed that for constraint language Γ on a 2-element set, the problem $\mathrm{CSP}(\Gamma)$ can be complete in only a handful of complexity classes: NP, P, $\oplus$P, NL, L, and AC^0. A similar classification has been conjectured in the general case by Larose and Tesson [Larose and Tesson 2009] with the class $\mathrm{mod}_p L$ for prime p instead of $\oplus$P. The best way to express this collection of conjectures is through omitting types of the local structure of algebras in the sense of tame congruence theory [Hobby and McKenzie 1988]. Assuming all the complexity classes involved are different, for a structure $\mathcal{H}$ the problem $\mathrm{CSP}(\mathcal{H})$

(a) is NP-complete unless $\mathsf{Alg}(\mathcal{H})$ omits the unary type,
(b) is $\mathrm{mod}_m L$-complete for some m if and only if $\mathsf{Alg}(\mathcal{H})$ omits the unary and semilattice types but does not omit the affine type ([Larose and Tesson 2009] shows that in this case $\mathrm{CSP}(\mathcal{H})$ is $\mathrm{mod}_p L$-hard for some prime p),
(c) is P-complete if and only if $\mathsf{Alg}(\mathcal{H})$ omits the unary type but not the semilattice type,
(d) is of bounded width if and only if $\mathsf{Alg}(\mathcal{H})$ omits the unary and affine types,
(e) is NL-complete if and only $\mathsf{Alg}(\mathcal{H})$ omits the unary, affine and semilattice types but not the lattice type,
(f) is L-complete if and only if $\mathsf{Alg}(\mathcal{H})$ omits the unary, affine, semilattice, and lattice types, but $\mathrm{CSP}(\mathcal{H})$ is not FO-expressible.

The hardness parts of all these conjectures are confirmed in [Larose and Tesson 2009]. Items (a) and (d) are the dichotomy theorem and the characterization of CSPs of bounded width which are also established. FO-expressible problems have been characterized in [Larose et al. 2007]. Kazda [Kazda 2018] proved that (e) implies (f). Finally, Dalmau and Krokhin [Dalmau and Krokhin 2008], and Barto et al. [Barto et al. 2012] made significant progress towards resolving (e). The rest of the problems above remain wide open.

Infinite CSPs. The majority of work on the CSP has been done under the assumption that the domain is finite. Allowing infinite domains expands the CSP framework so that it includes an enormous range of problems from GRAPH-SAT [Bodirsky and Pinsker 2015] to problems of scheduling and temporal reasoning [Allen 1983; Jonsson and Krokhin 2004; Bodirsky et al. 2018]. Problems representable by infinite CSPs such as temporal and spatial reasoning are standard in artificial intelligence. However, there has also been a significant amount of research initiated by [Bodirsky and Neˇ setřil 2003] on the algebraic structure of such problems. Although infinite CSPs use a variety of specific methods, the overall approach is to identify a finite algebraic structure in an as large as possible class of infinite CSPs [Barto and Pinsker 2016; Barto et al. 2017; Pinsker 2015]. The current dichotomy conjecture for infinite CSPs [Barto and Pinsker 2016; Barto et al. 2017] extends that for finite CSPs. For a recent survey on infinite CSPs see [Bodirsky and Mamino 2017].

Alternative parametrizations. In nonuniform CSPs we restrict a constraint language or a template relational structure. Clearly, other kinds of restrictions are also possible. For instance, in database theory one cannot assume any restrictions on the possible content of a database — which is a template structure in the CONJUNCTIVE QUERY EVALUATION problem — but some restrictions on the possible form of queries make much sense. If a CSP is viewed as in Definition 2.1, the constraint scopes of an instance $\mathcal{I}$ form a hypergraph on the set of variables. In a series of works [Gottlob et al. 2000; Flum et al. 2002; Gottlob et al. 2002; Grohe 2007; Grohe and Marx 2014] it has been shown that if this hypergraph allows some sort of decomposition, or is tree-like, then the CSP can be solved in polynomial time. The tree-likeness of a hypergraph is usually formalized as having bounded tree width, or bounded hypertree width, or bounded fractional hypertree width. This line of work culminated in [Marx 2013], in which Marx gave an almost tight description of classes of hypergraphs that give rise to a CSP solvable in polynomial time. *Hybrid* restrictions are also possible, although research in this direction has been more limited, see, [Feder et al. 2003; Feder and Hell 2006; Cooper and Zivny 2017] as an example.

The Promise CSP. Recently, Brakensiek and Guruswami [Brakensiek and Guruswami 2018b] suggested the following generalization of the CSP that they called the Promise CSP or PCSP. An instance of the PCSP consists of a pair of CSP instances $(\mathcal{I}, \mathcal{I}')$ such that they have the same number of constraints and for each constraint $\langle \mathrm{s}, R\rangle$ of $\mathcal{I}$ there is a constraint $\langle \mathrm{s}, R'\rangle$ of $\mathcal{I}'$ such that $R \subseteq R'$. The goal is to distinguish between the case when $\mathcal{I}$ is satisfiable and the case when $\mathcal{I}'$ is unsatisfiable. PCSP can express a much wider class of problems than the regular CSP, which includes, for instance, approximate graph and hypergraph colouring. It also uses a wider variety of solution algorithms such as LP and combinations of LP with other techniques [Brakensiek and Guruswami 2018b; 2018a]. On the other hand, PCSP allows for algebraic approach (although more limited than the regular CSP) as was demonstrated in [Brakensiek and Guruswami 2018b] and further developed in [Krokhin and Opr ̌ sal 2018]. In the latter work a connection between the LABEL COVER problem and checking the triviality of certain systems of algebraic identities has been established.

Variations of the CSP. Numerous variations and generalizations of the regular CSP have been studied over the last two decades. These include quantified CSPs, counting CSP, enumeration problems, CSPs with global constraints, a number of optimization problems such as Max- and Min-CSP, Valued CSP, the Min-Homomorphism problem. Many of the counting and optimization problems admit approximation algorithms, which have also been extensively studied. A dichotomy or other complexity classification results have been proved (or sometimes conjectured) for a number of those problems starting from the early works for 2-element structures, see, [Creignou et al. 2001]. A dichotomy theorem has been proved for the counting CSP [Bulatov 2013; Cai and Chen 2017]. Similarly, for the optimization problem of (Valued) CSP a dichotomy result is proved in [Thapper and Zivny 2016; Kolmogorov et al. 2017], and a (conditional) complexity classification of approximation of Valued CSP [Raghavendra 2008] was established. We should also mention the recent advances in complexity classification of Quantified CSP, enumeration problems, and a large number of related problems, in each of which the hope is to obtain some dichotomy-like results. Unfortunately, there is no room in this column to stop even briefly on any of these fascinating problems; each of them requires its own survey. The keen reader is however referred to a recent collection of such surveys [Krokhin and Zivny 2017].

REFERENCES

J.F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26 (1983), 832–843.

E. Allender, M. Bauland, N. Immerman, H. Schnoor, and H. Vollmer. 2009. The complexity of satisfiability problems: Refining Schaefer's theorem. *J. Comput. Syst. Sci.* 75, 4 (2009), 245–254.

L. Barto. 2011. The Dichotomy for Conservative Constraint Satisfaction Problems Revisited. In *LICS*. 301–310.

L. Barto. 2014a. The collapse of the bounded width hierarchy. *J. of Logic and Comput.* (2014).

L. Barto. 2014b. Constraint satisfaction problem and universal algebra. *SIGLOG News* 1, 2 (2014), 14–24.

L. Barto and J. Bulin. 2013. CSP Dichotomy for Special Polyads. *IJAC* 23, 5 (2013), 1151–1174.

L. Barto, M. Kompatscher, M. Olsák, T. Van Pham, and M. Pinsker. 2017. The equivalence of two dichotomy conjectures for infinite domain constraint satisfaction problems. In *LICS*. 1–12.

L. Barto and M. Kozik. 2012. Absorbing Subalgebras, Cyclic Terms, and the Constraint Satisfaction Problem. *Logical Methods in Comp. Sci.* 8, 1 (2012).

L. Barto and M. Kozik. 2014. Constraint Satisfaction Problems Solvable by Local Consistency Methods. *J. ACM* 61, 1 (2014), 3:1–3:19.

L. Barto and M. Kozik. 2017. Absorption in Universal Algebra and CSP. In *The Constraint Satisfaction Problem: Complexity and Approximability*. 45–77.

L. Barto, M. Kozik, M. Maroti, and T. Niven. 2009. CSP dichotomy for special triads. *Proc. Amer. Math. Soc* 137, 9 (2009), 2921–2934.

L. Barto, M. Kozik, and T. Niven. 2009. The CSP Dichotomy Holds for Digraphs with No Sources and No Sinks (A Positive Answer to a Conjecture of Bang-Jensen and Hell). *SIAM J. Comput.* 38, 5 (2009), 1782–1802.

L. Barto, M. Kozik, and R. Willard. 2012. Near Unanimity Constraints Have Bounded Pathwidth Duality. In *LICS*. 125–134.

L. Barto, A.A. Krokhin, and R. Willard. 2017. Polymorphisms, and How to Use Them. In *The Constraint Satisfaction Problem: Complexity and Approximability*. 1–44.

L. Barto and M. Pinsker. 2016. The algebraic dichotomy conjecture for infinite domain Constraint Satisfaction Problems. In *LICS*. 615–622.

L. Barto, M. Pinsker, and J. Oprˇ sal. 2018. The wonderland of reflections. *Israel J. of Math.* 223 (2018), 363–398.

J. Berman, P. Idziak, P. Marković, R. McKenzie, M. Valeriote, and R. Willard. 2010. Varieties with few subalgebras of powers. *Trans. Amer. Math. Soc.* 362, 3 (2010), 1445–1473.

M. Bodirsky and M. Mamino. 2017. Constraint Satisfaction Problems over Numeric Domains. In *The Constraint Satisfaction Problem: Complexity and Approximability*. 79–111.

M. Bodirsky, B. Martin, and A. Mottet. 2018. Discrete Temporal Constraint Satisfaction Problems. *J. ACM* 65, 2 (2018), 9:1–9:41.

M. Bodirsky and J. Neˇ setřil. 2003. Constraint Satisfaction with Countable Homogeneous Templates. In *CSL*. 44–57.

M. Bodirsky and P. Pinsker. 2015. Schaefer's Theorem for Graphs. *J. ACM* 62, 3 (2015), 19:1–19:52.

V.G. Bodnarchuk, L.A. Kaluzhnin, V.N. Kotov, and B.A. Romov. 1969. Galois theory for Post algebras. I. *Kibernetika* 3 (1969), 1–10.

J. Brakensiek and V. Guruswami. 2018a. An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. *CoRR* abs/1807.05194 (2018).

J. Brakensiek and V. Guruswami. 2018b. Promise Constraint Satisfaction: Structure Theory and a Symmetric Boolean Dichotomy. In *SODA*. 1782–1801.

A.A. Bulatov. 2002a. A dichotomy theorem for constraints on a three-element set. In *FOCS'*. 649–658.

A.A. Bulatov. 2002b. Mal'tsev constraints are tractable. *Electronic Colloquium on Computational Complexity (ECCC)* 034 (2002).

A.A. Bulatov. 2003. Tractable conservative Constraint Satisfaction Problems. In *LICS*. 321.

A.A. Bulatov. 2004. A Graph of a Relational Structure and Constraint Satisfaction Problems. In *LICS*. 448–457.

A.A. Bulatov. 2005. H-Coloring dichotomy revisited. *Theor. Comput. Sci.* 349, 1 (2005), 31–39.

A.A. Bulatov. 2006a. Combinatorial problems raised from 2-semilattices. *Journal of Algebra* 298, 2 (2006), 321–339.

A.A. Bulatov. 2006b. A dichotomy theorem for constraint satisfaction problems on a 3-element set. *J. ACM* 53, 1 (2006), 66–120.

A.A. Bulatov. 2011. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.* 12, 4 (2011), 24.

A.A. Bulatov. 2013. The complexity of the counting constraint satisfaction problem. *J. ACM* 60, 5 (2013), 34:1–34:41.

A.A. Bulatov. 2016a. Conservative constraint satisfaction re-revisited. *J. of Comp. and Syst. Sci.* 82, 2 (2016), 347–356.

A.A. Bulatov. 2016b. Graphs of finite algebras, edges, and connectivity. (2016). available at `https://www.cs.sfu.ca/ abulatov/papers/graph-connect-1.pdf`.

A.A. Bulatov. 2016c. Graphs of relational structures: restricted types. In *LICS*. 642–651.

A.A. Bulatov. 2017a. Constraint satisfaction problems over semilattice block Mal'tsev algebras. In *LICS*. 1–11.

A.A. Bulatov. 2017b. A Dichotomy Theorem for Nonuniform CSPs. In *FOCS*. 319–330.

A.A. Bulatov. 2017c. A Dichotomy Theorem for Nonuniform CSPs. *CoRR* abs/1703.03021 (2017).

A.A. Bulatov. 2018. Constraint Satisfaction Problems: Complexity and Algorithms. In *LATA*. 1–25.

A.A. Bulatov and V. Dalmau. 2006. A Simple Algorithm for Mal'tsev Constraints. *SIAM J. Comput.* 36, 1 (2006), 16–27.

A.A. Bulatov and P.G. Jeavons. 2001. *Algebraic structures in combinatorial problems*. Technical Report MATH-AL-4-2001. Technische universität Dresden, Dresden, Germany. available at `http://web.comlab.ox.ac.uk/oucl/research/areas/constraints/publications/index.html`.

A.A. Bulatov and P.G. Jeavons. 2003. An algebraic approach to multi-sorted constraits. In *CP*. 197–202.

A.A. Bulatov, P.G. Jeavons, and A.A. Krokhin. 2005. Classifying the Complexity of Constraints Using Finite Algebras. *SIAM J. Comput.* 34, 3 (2005), 720–742.

A.A. Bulatov, A.A. Krokhin, and B. Larose. 2008. Dualities for Constraint Satisfaction Problems. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar].* 93–124.

A.A. Bulatov and M. Valeriote. 2008. Recent Results on the Algebraic Approach to the CSP. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar].* 68–92.

J.-Y. Cai and X. Chen. 2017. Complexity of Counting CSP with Complex Weights. *J. ACM* 64, 3 (2017), 19:1–19:39.

A.K. Chandra and P.M. Merlin. 1977. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *STOC*. 77–90.

H. Chen and B. Larose. 2017. Asking the Metaquestions in Constraint Tractability. *TOCT* 9, 3 (2017), 11:1–11:27.

M.C. Cooper and S. Zivny. 2017. Hybrid Tractable Classes of Constraint Problems. In *The Constraint Satisfaction Problem: Complexity and Approximability*. 113–135.

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

V. Dalmau. 2006. Generalized Majority-Minority Operations are Tractable. *Logical Methods in Comp. Sci.* 2, 4 (2006).

V. Dalmau and A.A. Krokhin. 2008. Majority constraints have bounded pathwidth duality. *Eur. J. Comb.* 29, 4 (2008), 821–837.

R. Dechter. 2003. *Constraint processing*. Morgan Kaufmann Publishers.

T. Feder and P. Hell. 1998. List Homomorphisms to Reflexive Graphs. *J. Comb. Theory, Ser. B* 72, 2 (1998), 236–250.

T. Feder and P. Hell. 2006. Matrix partitions of perfect graphs. *Discrete Math.* 306, 19-20 (2006), 2450–2460.

T. Feder, P. Hell, and J. Huang. 1999. List Homomorphisms and Circular Arc Graphs. *Combinatorica* 19, 4 (1999), 487–505.

T. Feder, P. Hell, S. Klein, and R. Motwani. 2003. List partitions. *SIAM J. on Discrete Math.* 16 (2003), 449–478.

T. Feder and M.Y. Vardi. 1993. Monotone Monadic SNP and Constraint Satisfaction. In *STOC*. 612–622.

T. Feder and M.Y. Vardi. 1998. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. of Comput.* 28 (1998), 57–104.

J. Flum, M. Frick, and M. Grohe. 2002. Query evaluation via tree-decompositions. *J. ACM* 49, 6 (2002), 716–752.

R. Freese and R. McKenzie. 1987. *Commutator theory for congruence modular varieties*. London Math. Soc. Lecture Notes, Vol. 125. London.

R. Freese and M. Valeriote. 2009. On the Complexity of Some Maltsev Conditions. *IJAC* 19, 1 (2009), 41–77.

M.L. Furst, J.E. Hopcroft, and E.M. Luks. 1980. Polynomial-Time Algorithms for Permutation Groups. In *STOC*. 36–41.

D. Geiger. 1968. Closed systems of function and predicates. *Pacific J. of Math.* (1968), 95–100.

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

G. Gottlob, L. Leone, and F. Scarcello. 2002. Hypertree decomposition and tractable queries. *J. of Comp. and Syst. Sci.* 64, 3 (2002), 579–627.

G. Grätzer. 2008. *Universal algebra* (2nd ed.). Springer.

M. Grohe. 2007. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *J. ACM* 54, 1 (2007), 1:1–1:24.

M. Grohe and D. Marx. 2014. Constraint Solving via Fractional Edge Covers. *ACM Trans. Algorithms* 11, 1 (2014), 4:1–4:20.

P. Hell and Nešetřil. 2004. *Graphs and homomorphisms*. Oxford Lecture Series in Mathematics and its Applications, Vol. 28. Oxford University Press.

P. Hell and J. Nešetřil. 1990. On the Complexity of H-coloring. *J. of Combinatorial Th., Ser.B* 48 (1990), 92–110.

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

P. Idziak, M. Marković, R. McKenzie, M. Valeriote, and R. Willard. 2010. Tractability and Learnability Arising from Algebras with Few Subpowers. *SIAM J. Comput.* 39, 7 (2010), 3023–3037.

P.G. Jeavons, D.A. Cohen, and M.C. Cooper. 1998. Constraints, Consistency and Closure. *Artif. Intell.* 101, 1-2 (1998), 251–265.

P.G. Jeavons, D.A. Cohen, and M. Gyssens. 1997. Closure Properties of Constraints. *J. ACM* 44 (1997), 527–548.

P. Jonsson and A.A. Krokhin. 2004. Complexity classification in qualitative temporal constraint reasoning. *Artif. Intell.* 160, 1-2 (2004), 35–51.

A. Kazda. 2018. n-permutability and linear Datalog implies symmetric Datalog. *Logical Methods in Comp. Sci.* 14, 2 (2018).

P.G. Kolaitis. 2007. On the expressive power of logics on finite models. Springer.

P.G. Kolaitis and M.Y. Vardi. 1995. On the Expressive Power of Datalog: Tools and a Case Study. *J. Comput. Syst. Sci.* 51, 1 (1995), 110–134.

V. Kolmogorov, A.A. Krokhin, and M. Rolínek. 2017. The Complexity of General-Valued CSPs. *SIAM J. Comput.* 46, 3 (2017), 1087–1110.

M. Kozik. 2016. Weak consistency notions for all the CSPs of bounded width. In *LICS*. 633–641.

M. Kozik, A.A. Krokhin, M. Valeriote, and R. Willard. 2015. Characterizations of several Maltsev conditions. *Algebra Universalis* 73, 3 (2015), 205224.

A.A. Krokhin and J. Opr ˇ sal. 2018. Algebraic approach to promise constraint satisfaction. (2018). Oral communication, slides available at `https://materials.dagstuhl.de/files/18/18231/18231.JakubOprsal.Slides.pdf`.

A.A. Krokhin and S. Zivny (Eds.). 2017. *The Constraint Satisfaction Problem: Complexity and Approximability*. Dagstuhl Follow-Ups, Vol. 7. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik. http://www.dagstuhl.de/dagpub/978-3-95977-003-3

G. Kun. 2013. Constraints, MMSNP and expander relational structures. *Combinatorica* 33, 3 (2013), 335–347.

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

B. Larose, C. Loten, and C. Tardif. 2007. A Characterisation of First-Order Constraint Satisfaction Problems. *Logical Methods in Comp. Sci.* 3, 4 (2007).

B. Larose and P. Tesson. 2009. Universal algebra and hardness results for constraint satisfaction problems. *Theor. Comput. Sci.* 410, 18 (2009), 1629–1647.

B. Larose, M. Valeriote, and L. Zádori. 2009. Omitting Types, Bounded Width and the Ability to Count. *IJAC* 19, 5 (2009), 647–668.

A.K. Mackworth. 1977. Consistency in Networks of Relations. *Artif. Intell.* 8 (1977), 99–118.

P. Marković. 2011. The Complexity of CSPs on a 4-element set. (2011). Personal communication.

M. Maróti. 2011a. Malcev on top. (2011). Manuscript, available at `http://www.math.u-szeged.hu/ mmaroti/pdf/200x%20Maltsev%20on%20top.pdf`.

M. Maróti. 2011b. Tree on top of Malcev. (2011). Manuscript, available at `http://www.math.u-szeged.hu/ mmaroti/pdf/200x%20Tree%20on%20top%20of%20Maltsev.pdf`.

M. Maróti and R. McKenzie. 2008. Existence theorems for weakly symmetric operations. *Algebra Universalis* 59, 3-4 (2008), 463–489.

D. Marx. 2013. Tractable Hypergraph Properties for Constraint Satisfaction and Conjunctive Queries. *J. ACM* 60, 6 (2013), 42:1–42:51.

M. Pinsker. 2015. Algebraic and model theoretic methods in constraint satisfaction. *CoRR* abs/1507.00931 (2015).

A.F. Pixley. 1979. Characterizations of arithmetical varieties. *Algebra Universalis* 9, 1 (1979), 87–98.

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

P. Raghavendra. 2008. Optimal algorithms and inapproximability results for every CSP?. In *STOC*. 245–254.

O. Reingold. 2008. Undirected connectivity in log-space. *J. ACM* 55, 4 (2008), 17:1–17:24.

T.J. Schaefer. 1978. The Complexity of Satisfiability Problems. In *STOC*. 216–226.

J. Thapper and S. Zivny. 2016. The Complexity of Finite-Valued CSPs. *J. ACM* 63, 4 (2016), 37:1–37:33.

L.G. Valiant. 2006. Accidental Algorithms. In *FOCS*. 509–517.

L.G. Valiant. 2008. Holographic Algorithms. *SIAM J. Comput.* 37, 5 (2008), 1565–1594.

D. Zhuk. 2016a. On CSP Dichotomy Conjecture. In *AAA'92*. 32.

D. Zhuk. 2016b. The proof of CSP Dichotomy Conjecture for 5-element domain. In *AAA'91*.

D. Zhuk. 2017a. A Proof of CSP Dichotomy Conjecture. In *FOCS*. 331–342.

D. Zhuk. 2017b. The Proof of CSP Dichotomy Conjecture. *CoRR* abs/1704.01914 (2017). http://arxiv.org/abs/1704.01914

D. Zhuk. 2018. A modification of the CSP algorithm for infinite languages. *CoRR* abs/1803.07465 (2018).