Constraint Satisfaction Problems: Complexity and Algorithms

Andrei A. Bulatov[(*)]

School of Computing Science
Simon Fraser University
8888 University Drive, Burnaby, Canada
abulatov@sfu.ca

Abstract. 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 Constraint Satisfaction Problem

We begin with definitions, examples and brief historical remarks on the Constraint Satisfaction Problem.

1.1 The Problem

The archetypal example of the Constraint Satisfaction Problem is a Sudoku puzzle, see, Fig. 1: One needs to assign values to every cell of the puzzle so that the assignment satisfies certain constraints, such as the values in every row, column, and smaller block are different. This example can be naturally generalized in the following way. 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 $\mathbf{a}[i]$.

Definition 1. *Let $A_1, \ldots, A_n$ be finite sets. An instance $\mathcal{I}$ of the Constraint Satisfaction Problem (CSP for short) over $A_1, \ldots, A_n$ consists of a finite set of variables V such that each $v \in V$ is assigned a domain A_{i_v}, $i_v \in \{1, \ldots, n\}$, and a finite set of constraints $\mathcal{C}$. Each constraint is a pair $\langle \mathbf{s}, R\rangle$ where R is a relation over $A_1, \ldots, A_n$ (say, k-ary), often called the* constraint relation*, and $\mathbf{s}$ is a k-tuple of variables from V, called the* constraint scope*. Let $\sigma : V \to A = A_1 \cup \cdots \cup A_n$ with $\sigma(v) \in A_{i_v}$; we write $\sigma(\mathbf{s})$, for $(\sigma(\mathbf{s}[1]), \ldots, \sigma(\mathbf{s}[k]))$. A* solution *of $\mathcal{I}$ is a mapping $\sigma : V \to A$ such that for every constraint $\langle \mathbf{a}, R\rangle \in \mathcal{C}$ 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.*

[(*)] This research was supported by an NSERC Discovery grant

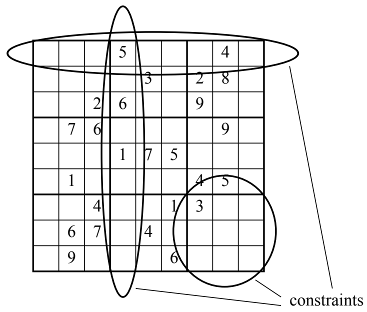

Fig. 1. A Sudoku puzzle as a CSP. The ellipses inducate some of the constraints

Since its inception in the early 70s [59], the CSP has become a very popular and powerful framework, widely used to model computational problems first in artificial intelligence, [34] and later in many other areas.

Modeling a specific computational problem usually gives rise to a *restricted CSP*. Such restrictions can be imposed either on the type of the allowed constraint relations, or on the way the constraint scopes interact, or both. Restrictions of the first kind — the main subject of this paper — are usually given by specifying a constraint language, that is, a set of relations Γ over a set, or a collection of sets such that every constraint relation has to belong to Γ. More formally, let $A_1, \ldots, A_\ell$ be finite sets and Γ a set (finite or infinite) of relations over $A_1, \ldots, A_\ell$, called a *constraint language*. Then CSP(Γ) 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 CSP(Γ).

k-Col The standard k-Coloring problem has the form 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).

3-SAT An instance of the 3-SAT problem is a propositional logic formula in CNF each clause of which contains 3 literals, and asking if it has a satisfying assignment. Thus, 3-SAT is equivalent to CSP($\Gamma_{3\mathrm{SAT}}$), where $\Gamma_{3\mathrm{SAT}}$ is the constraint language on $\{0, 1\}$ and containing 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 3Lin(F) be the problem of deciding the consistency of a system of linear equations over F each of which contains at most 3 variables. Then 3Lin(F) is equivalent to CSP($\Gamma_{3\mathrm{Lin}(F)}$), where $\Gamma_{3\mathrm{Lin}(F)}$ is the constraint language over F whose relations are given by an equation with at most 3 variables.

MonEq Let M be a monoid (or a semigroup). An equation over M is an expression of the form $t = s$, where t and s are words that involve indeterminates and constants from M. A solution of $t = s$ is an assignment of elements from M to the indeterminates such that t and s evaluate to the same element of M. In the problem MonEq(M) we are given a system of equations over monoid M, and the objective is to decide, whether or not there exists an assignment to the indeterminates that is a solution for each of the given equations. Similar to

3Lin, MonEq(M) is the problem CSP($\Gamma_{\text{MonEq}(M)}$), where $\Gamma_{\text{MonEq}(M)}$ is the constraint language consisting of all relations representable by an equation over M. Note that $\Gamma_{\text{MonEq}(M)}$ is infinite in general.

1.2 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 an answer). The Query Containment problem asks, given two queries Φ and Ψ, whether $\Phi \to \Psi$ is true in any database with the appropriate 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 $R_1, \dots, R_n$ each of which has a fixed arity $\text{ar}(R_i)$. A *relational structure* over vocabulary $R_1, \dots, R_n$ is a tuple $\mathcal{H} = (H; R_1^{\mathcal{H}}, \dots, R_n^{\mathcal{H}})$ such that H is a 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 2. *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 sentence 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 original definition of the CSP, we consider its special case, k-Colouring. The vocabulary corresponding to the problem contains just one binary predicate $R_{\neq}$. Let $\mathcal{H}_k$ be the relational structure with universe $[k] = \{1, \dots, k\}$ in the vocabulary $\{R_{\neq}\}$, where $R_{\neq}^{\mathcal{H}_k}$ is interpreted as the disequality relation on the set $[k]$. (In the future we will tend to omit the superscripts indicating an interpretation, whenever it does not lead to a confusion.) Then an instance $G = (V, E)$ of k-Colouring is equivalent to testing whether conjunctive sentence $\bigwedge_{(u,v) \in E} R_{\neq}(u, v)$ (we omit the quantifier prefix) is true in $\mathcal{H}$.

The Query Evaluation problem is thus just the CSP, when restricted to conjunctive queries. A database is then cosidered as the input relational structure. The Chandra-Merlin Theorem [29] 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. We will return to this subject in Section 3.1.

1.3 Homomorphisms and Dichotomy

The complexity of the CSP and its solution algorithms have been a major theme since the problem was introduced. The general CSP is NP-complete, as it can be easily shown. However, various restrictions of the CSP may result in more tractable problems. Paper [36, 37] by Feder and Vardi marked the beginning of a systematic research of the complexity of the CSP. Among the numerous insights of this paper, it introduced a new definition of the CSP.

Let $\mathcal{G}$ and $\mathcal{H}$ be relational structures over the same vocabulary. A *homomorphism* from $\mathcal{G}$ to $\mathcal{H}$ is a mapping $\varphi\colon G \to H$ from the universe G of $\mathcal{G}$ (the *instance*) to the universe H of $\mathcal{H}$ (the *template*) such that, for every relation $R^{\mathcal{G}}$ of $\mathcal{G}$ and every tuple $\mathbf{a} \in R^{\mathcal{G}}$, we have $\varphi(\mathbf{a}) \in R^{\mathcal{H}}$.

Definition 3. *An instance of the CSP is a pair of relational structures $\mathcal{G}, \mathcal{H}$ over the same vocabulary. The objective is to decide whether or not there exists a homomorphism from $\mathcal{G}$ to $\mathcal{H}$.*

The homomorphic definition of the CSP makes its restricted version very elegant. Let $\mathcal{H}$ be a relational structure. An instance of the *nonuniform* constraint satisfaction problem CSP($\mathcal{H}$) is a structure $\mathcal{G}$ over the same vocabulary as $\mathcal{H}$, and the question is whether there is a homomorphism from $\mathcal{G}$ to $\mathcal{H}$.

We again illustrate the correspondence between the definition above and Definition 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 framework 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 is 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. [48, 49].

Feder and Vardi in [36, 37] also initiated the line of research that is central for this paper, the study of the complexity of nonuniform CSPs. They observed that in all known cases a nonuniform CSP either can be solved in polynomial time, e.g. CSP(Γ_{3LIN}) or 2-COLOURING, or is NP-complete, e.g., 3-SAT or k-COLOURING for $k > 2$. Two results were quite suggestive at that point. The first one is the classification of the complexity of CSP($\mathcal{H}$) for 2-element structures (or the GENERALIZED SATISFIABILITY problem, as it was referred to) by Schaefer [68]; who proved that every such problem is either solvable in polynomial time, or is NP-complete. The second result by Hell and Nesetril [49] establishes that the H-COLOURING problem, where H is a graph, follows the same pattern: The H-COLOURING problem can be solved in polynomial time if H is bipartite or has a loop, and it is NP-complete otherwise. This allowed Feder and Vardi to pose the following

Conjecture 4 (The Dichotomy Conjecture). For every finite relational structure $\mathcal{H}$ the problem CSP($\mathcal{H}$) is either solvable in polynomial time or is NP-complete.

Most of the remaining part of this paper is devoted to resolving the Dichotomy Conjecture.

1.4 The Other Side and Other Types

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 1, the cosntraint scopes of an instance $\mathcal{I}$ form a hypergraph on the set of variables. In a series of works [43, 40, 44, 46, 47] 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 treewidth, or bounded hypertree width, or bounded fractional hypertree width. This line of work culminated in [64], 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, [38, 39, 30] as an example.

Along with the decision version of the CSP, other versions of the problem have been intensively studied, see, [31] for definitions and early results on many of them. These include the Quantified CSP, which is the problem of checking whether or nor a conjunctive sentence allowing both universal and existential quantifiers is true in a given relational structure [11]. In the MaxCSP one needs to maximize the number of satisfied constraints. Note that often constraints in MaxCSP are considered weighted and the the problem is to maximize the total weight of satisfied constraints. Another variation of this problem is Valued CSP, in which the constraints are replaced by functions that give a weight to each assignment. The problem is to minimize (or maximize) the weight of an assignment. This problem has been considered for both exact optimization [57, 69, 56] and approximation algorithms [66, 33, 5]. The Counting CSP has received much attention, particularly due to its connections to statistical physics. In this problem the goal is to count the number of solutions of a CSP (the unweighted version) or to evaluate the total weight of the assignments (the weighted version). The complexity of exact counting is well understood [35, 18, 28], while approximate counting remains a largely open area [54].

2 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 determe 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 [45, 50], surveys [7, 6, 27, 26], and research papers [25, 16, 17, 19, 24, 9, 2, 4, 3, 51].

2.1 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 5. Let $K_3 = ([3], E)$ be a 3-element complete graph. Its edge relation is the binary disequality relation on [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 in [52].

Theorem 6 ([52]). *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*[1] *to* $\mathrm{CSP}(\Gamma)$.

It was later shown that pp-definability in Theorem 6 can be replaced with a more general notion of *pp-constructibility* [7, 8].

2.2 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 $\mathbf{a}_1, \dots, \mathbf{a}_k \in R$ the tuple $f(\mathbf{a}_1, \dots, \mathbf{a}_k)$ also belongs to R, where $f(\mathbf{a}_1, \dots, \mathbf{a}_k)$ stands for $(f(\mathbf{a}_1[1], \dots, \mathbf{a}_k[1]), \dots, f(\mathbf{a}_1[n], \dots, \mathbf{a}_k[n]))$. 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 7. 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 the system defining R, and $\mathbf{x}, \mathbf{y}, \mathbf{z} \in R$. Then

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

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.

[1] In fact, due to the result of [67] this reduction can be made log-space.

Example 8. In [55] it was shown that $\textsc{MonEq}(M)$ for a monoid M can be solved in poltnomial time if and only if M is commutative and is the union of its subgroups. If this is the case then the operation $t(x, y, z) = xy^{\omega-1}z$ is a polymorphism of $\Gamma_{\textsc{MonEq}(M)}$ (see also [58]). Here x^ω denotes the power of x such that x^ω is an idempotent of M.

Several other useful polymorphisms are the following

Example 9 ([52, 53, 24]). (1) A binary semilattice operation.
(2) A k-ary operation g on A is called a *near-unanimity* operation, or NU if

$$g(y, x, \dots, x) = g(x, y, x, \dots, x) = \dots = g(x, \dots, x, y) = x$$

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

$$g(y, x, \dots, x) = g(x, y, x, \dots, x) = \dots = g(x, \dots, x, y).$$

(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 7 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. Note that the operation $xy^{\omega-1}z$ from Example 8 is not necessarily Mal'tsev.
(5) If every polymorphism f of a relational structure $\mathcal{H}$ is such that $f(x_1, \dots, x_n) = x_i$ for some i and all $x_1, \dots, x_n \in H$, then CSP($\mathcal{H}$) is NP-complete.
(6) Schaefer's Theorem [68] can be stated in terms of polymorphisms. Let $\mathcal{H}$ be a 2-element relational structure (we assume its universe to be $\{0, 1\}$). The problem CSP($\mathcal{H}$) is solvable in polynomial time if and only if one of the following operations is a polymorphism of $\mathcal{H}$: 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 CSP($\mathcal{H}$) is NP-complete.

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

Theorem 10 (Galois connection, [10, 42]). *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.*

2.3 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 $\mathcal{A} = (A, R_1, \ldots, R_k)$ such that $R_1, \ldots, R_k \in \mathsf{Inv}(F)$.

This correspondence can be extended to CSPs: For an algebra $\mathbb{A}$ by CSP($\mathbb{A}$) we denote the class of problems CSP($\mathcal{A}$), $\mathcal{A} \in \mathsf{Str}(\mathbb{A})$. Equivalently, CSP($\mathbb{A}$) can be thought of as 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 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 it guarantees that there is any uniformity among those algorithms. In general, it is plausible that for a tractable algebra $\mathbb{A} = (A, F)$ the problem CSP($\mathsf{Inv}(F)$) is NP-hard. If the problem CSP($\mathsf{Inv}(F)$) is solvable in polynomial time, we call $\mathbb{A}$ *globally tractable.* Algebra $\mathbb{A}$ is called *NP-complete* if some CSP($\mathcal{A}$), $\mathcal{A} \in \mathsf{Str}(\mathbb{A})$ is NP-complete. Algebra $\mathbb{A}$ is *globally NP-complete* if CSP($\mathsf{Inv}(F)$) is NP-complete.

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

Conjecture 11 (Dichotomy Conjecture+). Every finite algebra is either globally tractable or NP-complete (in the local sense).

Our next goal is to make Conjecture 11 more precise. We achieve this goal in Section 2.4, while now we observe that the standard algebraic constructions behave quite well with respect to reducibility between CSPs.

Theorem 12 ([25]). *Let* $\mathbb{A} = (A; F)$ *be a finite algebra. Then*

(1) *if* $\mathbb{A}$ *is tractable then so is every subalgebra, homomorphic image, and direct power of* $\mathbb{A}$*.*

(2) *if* $\mathbb{A}$ *has a NP-hard subalgebra, homomorphic image, or direct power, then* $\mathbb{A}$ *is NP-hard.*

More reducibility properties related to term operations of an algebra can be proved. Recall that an operation f on a set A is said to be *idempotent* if the equality $f(x, \ldots, x) = x$ holds for all $x \in A$. An algebra all of whose term operations are idempotent is said to be *idempotent.*

Theorem 13 ([25]). *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 13 reduces the Dichotomy Conjecture+ 11 to idempotent algebras.

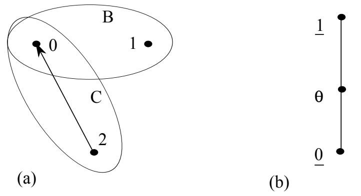

Fig. 2. (a) Algebra $\mathbb{A}_M$. (b) The congruence lattice of $\mathbb{A}_M$

Example 14. The next example will be our running example throughout the paper. Let $A = \{0, 1, 2\}$, and let $\mathbb{A}_M$ be the algebra with universe A and two basic operations: a binary operation r such that $r(0, 0) = r(0, 1) = r(2, 0) = r(0, 2) = r(2, 1) = 0$, $r(1, 1) = r(1, 0) = r(1, 2) = 1$, $r(2, 2) = 2$; and a ternary operation t such that $t(x, y, z) = x - y + z$ if $x, y, z \in \{0, 1\}$, where $+, -$ are the operations of $\mathbb{Z}_2$, $t(2, 2, 2) = 2$, and otherwise $t(x, y, z) = t(x', y', z')$, where $x' = x$ if $x \in \{0, 1\}$ and $x' = 0$ if $x = 2$; the values y', z' are obtained from y, z by the same rule. It is an easy excercise to verify the following facts: (a) $\mathbb{B} = (\{0, 1\}, r|_{\{0,1\}}, t|_{\{0,1\}})$ and $\mathbb{C} = (\{0, 2\}, r|_{\{0,2\}}, t|_{\{0,2\}})$ are subalgebras of $\mathbb{A}_M$, (b) the partition $\{0, 1\}, \{2\}$ is a congruence of $\mathbb{A}_M$, let us denote it θ, (c) algebra $\mathbb{C}$ is basically a semilattice, that is, a set with a semilattice operation, see Fig 2(a).

The classes of congruence θ are $0^\theta = \{0, 1\}, 2^\theta = \{2\}$. Then the quotient algebra $\mathbb{A}_M/_\theta$ is also basically a semilattice, as $r/_\theta(0^\theta, 0^\theta) = r/_\theta(0^\theta, 2^\theta) = r/_\theta(2^\theta, 0^\theta) = 0^\theta$ and $r/_\theta(2^\theta, 2^\theta) = 2^\theta$. ◇

2.4 The CSP and Omitting Types

In the 1980s Hobby and McKenzie developed tame congruence theory that studies the local structure of algebras [50]. They discovered that the local structure of universal algebras is surprisingly well behaved and can be classified into just five types. Each type is associated with a certain basic algebra, and if an algebra admits a type, it means that its local structure resembles that of the corresponding basic algebra. The five basic algebras and corresponding types are:

1. A *unary* algebra whose basic operations are permutations (*unary type*);
2. A one-dimensional vector space over some finite field (*affine type*);
3. A 2-element boolean algebra whose basic operations include conjunction, disjunction, and negation (*boolean type*);
4. A 2-element lattice whose basic operations include conjunction and disjunction (*lattice type*);
5. A 2-element semilattice whose basic operations include a semilattice operation (*semilattice type*).

Omitting or admitting types is strongly related to the complexity of the CSP. Theorem 5 from [25] claims that if a relational structure $\mathcal{A}$ is such that $\mathsf{Alg}(\mathcal{A})$ is idempotent and admits the unary type then CSP($\mathcal{A}$) is NP-complete. Combined with Theorem 12 this allows for a more precise Dichotomy Conjecture.

Conjecture 15. If a relational structure $\mathcal{A}$ is such that $\mathsf{Alg}(\mathcal{A})$ is idempotent, then CSP($\mathcal{A}$) is solvable in polynomial time if and only if no subalgebra of $\mathsf{Alg}(\mathcal{A})$ admits the unary type. Otherwise it is NP-complete.

Or in the stronger algebraic version

Conjecture 16 (Dichotomy Conjecture ++). An idempotent algebra $\mathbb{A}$ is globally tractable if and only if none of its subalgebras admits the unary type. Otherwise it is NP-complete.

The results [63] imply that the latter condition in Conjecture 16 is also equivalent to the existence of a weak near-unanimity term operation in $\mathbb{A}$.

Conjecture 16 has been confirmed in a number of special cases.

- Schaefer's classification of 2-element structures [68] with respect to complexity 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 [65] this is equivalent to having a term weak near-unanimity operation.
- Let H be a graph, $\mathbb{A} = \mathsf{Alg}(H)$, and let $\mathbb{B}$ the idempotent algebra constructed from $\mathbb{A}$ as in Theorem 13. If H is bipartite then $\mathbb{B}$ is 2-element and has a majority term operation. Otherwise $\mathbb{B}$ admits the unary type [15]. Thus the classification from [49] matches the Dichotomy Conjecture++.
- The Dichotomy Conjecture++ was confirmed for 3-element algebras in [12, 16], and for 4-element algebras in [61].
- It was shown in [13, 17] that the Dichotomy Conjecture++ holds for *conservative* algebras, that is, algebras in which every subset of the universe is a subalgebra. These results have also been simplified in [1, 19].
- Finally, Zhuk in [71, 70] proved the conjecture for 5- and 7-element algebras.

In the rest of this paper we show that Conjecture 16 is true. The hardess part of the conjecture follows from the mentioned result of [25]; so we focus on the algorithmic part. The algorithm presented here is based on [23] (a full version can be found in [22]). Note that the conjecture was also independently proved by Zhuk [72].

3 CSP Algorithms

It would be natural to expect a wide variety of algorithms solving the CSP in those cases in which it can be solved in polynomial time. However, surprisingly, only two types of such algorithms are known, and for each type there is 'the most general' algorithm, which means that basically only two CSP algorithms exist.

3.1 Local Propagation Algorithms

The first type can be described as local propagation algorithms. We describe one such algorithm, applicable whenever any other propagation algorithm solves the problem.

Let $R \subseteq A^n$ be a relation, $\mathbf{a} \in A^n$, and $J = \{i_1, \dots, i_k\} \subseteq [n]$. Let $\mathrm{pr}_J \mathbf{a} = (\mathbf{a}[i_1], \dots, \mathbf{a}[i_k])$ and $\mathrm{pr}_J R = \{\mathrm{pr}_J \mathbf{a} \colon \mathbf{a} \in R\}$. Often we will use sets of CSP variables to index entries of tuples and relations. Projections in these 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 $C = \langle \mathbf{s}, R \rangle \in \mathcal{C}$, the set $\mathcal{C}_W$ includes the constraint $C_W = \langle \mathbf{s} \cap W, \mathrm{pr}_{\mathbf{s} \cap W} R \rangle$. 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 intersections 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 $C = \langle \mathbf{s}, R \rangle \in \mathcal{C}$ such that $v \in \mathbf{s}$, 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 and assume that CSP instances may have different domains, which nevertheless are always subalgebras or quotient algebras of the original algebra $\mathbb{A}$. It will be convenient to denote the domain of v by $\mathbb{A}_v$. The domain $\mathbb{A}_v$ 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 $c \in \mathbb{A}_w$ 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 $k, k+1$.

Instance $\mathcal{I}$ is said to be *minimal* (or *globally minimal*) if for every $C = \langle \mathbf{s}, R \rangle \in \mathcal{C}$ and every $\mathbf{a} \in R$ there is a solution φ such that $\varphi(\mathbf{s}) = \mathbf{a}$. Similarly, $\mathcal{I}$ is said to be *globally 1-minimal* if for every $v \in V$ and $a \in \mathbb{A}_v$ there is a solution φ with $\varphi(v) = a$.

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. [34]). These algorithms work by changing the constraint relations and the domains of the variables eliminating some tuples and elements from them. We call such a process *tightening* the instance. It is important to notice that if the original instance belongs to CSP($\mathbb{A}$) for some algebra $\mathbb{A}$, that is, all its constraint relations are invariant under the basic operations of $\mathbb{A}$, the constraint relations obtained by propagation algorithms are also invariant under the basic operations of $\mathbb{A}$, and so the resulting instance also belongs to CSP($\mathbb{A}$). Establishing minimality amounts to solving the problem and so not always can be easily done.

If a constraint propagation algorithm solves a CSP, the problem is said to be of bounded width. More precisely, CSP(Γ) (or CSP($\mathbb{A}$)) is said to have *bounded width* if for some k every $(k, k+1)$-minimal instance from CSP(Γ) (or CSP($\mathbb{A}$)) has a solution (we also say that CSP(Γ) has width k in this case). Problems of bounded width are well studied, see the older survey [26] and more recent [2].

Theorem 17 ([2, 21, 14, 60]). *For an idempotent algebra* $\mathbb{A}$ *the following are equivalent:*

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

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

(3) $\mathbb{A}$ *has a weak near-unanimity term 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).*

Example 18. (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 NP-complete otherwise.

(3) The 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 CSP($\Gamma_{k\text{-HORN}}$) and has width k.

3.2 Gaussian Elimination and Few Subpowers

The simplest algorithm of the second type is known from basic linear algebra — Gaussian elimination. While propagation algorithms cannot solve the Lin problem, it is solvable by Gaussian elimination. A similar algorithm solving group constraints, defined in terms of finite groups, was suggested in [37].

Algebraic techniques make it possible to generalize the Gaussian elimination algorithm. The algorithm from [24] solving CSP($\mathcal{A}$) for a relational structure $\mathcal{A}$ with a Maltsev polymorphism can be viewed as a generalization of Gaussian elimination in the following sense. Similar to the output of Gaussian elimination it constructs some sort of a basis or a compact representation of the set of all solutions of a CSP.

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 [9, 51]. 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. [9, 51]. A minor generalization of the algorithm from [32] solves CSP($\mathbb{A}$), where $\mathbb{A}$ has few subpowers.

Here the few subpowers algorithm is used in the context of semilattice edges. 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 $\{a,b\}$. For example, the set $\{0,2\}$ from Example 14 is a semilattice edge, and the operation r of $\mathbb{A}_M$ witnesses that.

Proposition 19 ([21]). *If an idempotent algebra $\mathbb{A}$ has no semilattice edges, it has few subpowers, and therefore* CSP($\mathbb{A}$) *is solvable in polynomial time.*

Semilattice edges have other useful properties including the following one that we use for reducing a CSP to smaller problems.

Lemma 20 ([20]). *For any idempotent algebra $\mathbb{A}$ there is a term operation xy (think multiplication) such that xy is a semilattice operation on any semilattice edge and for any $a, b \in \mathbb{A}$ either $ab = a$ or $\{a, ab\}$ is a semilattice edge.*

Note that any semilattice operation satisfies the conditions of Lemma 20. The operation r of the algebra $\mathbb{A}_M$ from Example 14 is not a semilattice operation (it is not commutative), but it satisfies the conditions of Lemma 20.

4 Congruence Separation and Centralizers

We now move on to describe the algorithm resolving the Dichotomy Conjecture. In this section we introduce two of the key ingredients of our algorithm.

4.1 Separating Congruences

Unlike the vast majority of the literature on the algebraic approach to the CSP we use not only term operations, but also polynomial operations of an algebra. It should be noted however that the first to use polynomials for CSP algorithms was Maroti in [62]. We make use of some ideas from that paper in the next section. Let $f(x_1, \ldots, x_k, y_1, \ldots, y_\ell)$ be a $k+\ell$-ary term operation of an algebra $\mathbb{A}$ and $b_1, \ldots, b_\ell \in \mathbb{A}$. The operation $g(x_1, \ldots, x_k) = f(x_1, \ldots, x_k, b_1, \ldots, b_\ell)$ is called a *polynomial* of $\mathbb{A}$. A polynomial for which $k = 1$ is said to be a *unary* polynomial. If α is a congruence, and f is a unary polynomial, by $f(\alpha)$ we denote the set of pairs $\{(f(a), f(b)) \mid (a,b) \in \alpha\}$.

Let $\mathbb{A}$ be an algebra and let $\mathsf{Con}(\mathbb{A})$ denote its congruence lattice. For $\alpha, \beta \in \mathsf{Con}(\mathbb{A})$ we write $\alpha \prec \beta$ if $\alpha < \beta$ (that is, $\alpha \subset \beta$ as sets of pairs) and $\alpha \le \gamma \le \beta$ in $\mathsf{Con}(\mathbb{A})$ implies $\gamma = \alpha$ or $\gamma = \beta$. If this is the case we call (α, β) a *prime interval* in $\mathsf{Con}(\mathbb{A})$. Let $\alpha \prec \beta$ and $\gamma \prec \delta$ be prime intervals in $\mathsf{Con}(\mathbb{A})$. We say that $\alpha \prec \beta$ can be *separated* from $\gamma \prec \delta$ if there is a unary polynomial f of $\mathbb{A}$ such that $f(\beta) \not\subseteq \alpha$, but $f(\delta) \subseteq \gamma$. The polynomial f in this case is said to *separate* $\alpha \prec \beta$ from $\gamma \prec \delta$.

Example 21. The unary polynomials of the algebra $\mathbb{A}_M$ from Example 14 include the following unary operations (these are the polynomials we will use, there are more unary polynomials of $\mathbb{A}_M$):
$h_1(x) = r(x,0) = r(x,1)$, such that $h_1(0) = h_1(2) = 0, h_1(1) = 1$;

$h_2(x) = r(2, x)$, such that $h_2(0) = h_2(1) = 0$, $h_2(2) = 2$;
$h_3(x) = r(0, x) = 0$.

The lattice $\mathsf{Con}(\mathbb{A}_M)$ has two prime intervals $\underline{0} \prec \theta$ and $\theta \prec \underline{1}$ (see Example 14 and Fig 2(b)). As is easily seen, $h_3(\underline{1}) \subseteq \underline{0}$, therefore h_3 collapses both prime intervals. Since $h_1(\theta) \not\subseteq \underline{0}$, but $h_1(\underline{1}) \subseteq \theta$, polynomial h_1 separates $(\underline{0}, \theta)$ from $(\theta, \underline{1})$. Similarly, the polynomial h_2 separates $(\theta, \underline{1})$ from $(\underline{0}, \theta)$, because $h_2(\underline{1}) \not\subseteq \theta$, while $h_2(\theta) \subseteq \underline{0}$. ⋄

In a similar way separation can be defined for prime intervals in different coordinate positions of a relation. Let R be a subdirect product of $\mathbb{A}_1 \times \cdots \times \mathbb{A}_n$, that is, $\mathrm{pr}_i R = \mathbb{A}_i$ for $i \in [n]$. Then R can also be viewed as an algebra with operations acting component-wise, and polynomials of R can be defined in the same way. Since every basic operation acts on R component-wise, its unary polynomials also act component-wise. Therefore, for a unary polynomial f of R it makes sense to consider $f(a)$, where $a \in \mathbb{A}_i$, $i \in [n]$. Let $i, j \in [n]$ and let $\alpha \prec \beta$, $\gamma \prec \delta$ be prime intervals in $\mathsf{Con}(\mathbb{A}_i)$ and $\mathsf{Con}(\mathbb{A}_j)$, respectively. Interval $\alpha \prec \beta$ can be separated from $\gamma \prec \delta$ if there is a unary polynomial f of R such that $f(\beta) \not\subseteq \alpha$ but $f(\delta) \subseteq \gamma$. The binary relation 'cannot be separated' on the set of prime intervals of an algebra or factors of a relation is easily seen to be reflexive and transitive. We will say that $\alpha \prec \beta$, $\gamma \prec \delta$ cannot be separated if $\alpha \prec \beta$ and $\gamma \prec \delta$ cannot be separated from each other.

Example 22. Let R be a ternary relation over $\mathbb{A}_M$ invariant under r, t, given by

$$R = \begin{pmatrix} 0\,0\,1\,1\,0\,0\,1\,1\,2\,2 \\ 0\,1\,1\,0\,0\,1\,1\,0\,2\,2 \\ 0\,0\,0\,0\,1\,1\,1\,1\,0\,2 \end{pmatrix},$$

where triples, the elements of the relation are written vertically. It will be convenient to distinguish congruences in the three factors of R, so we denote them by $\underline{0}_i, \theta_i, \underline{1}_i$ for the ith factor. Since $\mathrm{pr}_{12} R$ is the congruence θ, any unary polynomial h of R acts identically modulo θ on the first and the second coordinate positions. In particular, the prime interval $(\theta_1, \underline{1}_1)$ cannot be separated from the prime interval $(\theta_2, \underline{1}_2)$. Consider the polynomial $h(x)$ of R given by

$$h(x) = r\left(\begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix}, x\right) = \begin{pmatrix} r(2,x) \\ r(2,x) \\ r(0,x) \end{pmatrix} = \begin{pmatrix} h_2(x) \\ h_2(x) \\ h_3(x) \end{pmatrix},$$

it is a polynomial of R because $(2, 2, 0) \in R$. Since $h_2(\underline{1}) \not\subseteq \theta$, but $h_3(\underline{1}) \subseteq \theta$ and $h_3(\theta) \subseteq \underline{0}$, the prime interval $(\theta_2, \underline{1}_2)$ can be separated from $(\underline{0}_3, \theta_3)$ and $(\theta_3, \underline{1}_3)$. Similarly, the interval $(\theta_3, \underline{1}_3)$ can be separated from $(\underline{0}_1, \theta_1), (\underline{0}_2, \theta_2)$. Through a slightly more involved argument it can be shown that $(\theta_3, \underline{1}_3)$ cannot be separated from $(\theta_1, \underline{1}_1)$, $(\theta_2, \underline{1}_2)$. In the next section we explain why the prime intervals $(\underline{0}_i, \theta_i), (\underline{0}_j, \theta_j)$ cannot be separated from each other. ⋄

4.2 Quasi-Centralizers

The second ingredient we will use here is the notion of quasi-centralizer of a pair of congruences. It is similar to the centralizer as it is defined in commutator theory [41], albeit the exact relationship between the two concepts is not quite clear, and so we name it differently for safety.

For an algebra $\mathbb{A}$, a term operation $f(x, y_1, \ldots, y_k)$, 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 an only if, for any term operation $f(x, y_1, \ldots, y_k)$, 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 $\mathbb{A}$. Next we show how it is related to the structure of algebra $\mathbb{A}$ and the corresponding CSP.

Example 23. In the algebra $\mathbb{A}_M$, see Example 14, the quasi-centralizer acts as follows: $\zeta(\underline{0}, \theta) = \underline{1}$ and $\zeta(\theta, \underline{1}) = \theta$. We start with the second centralizer. Since every polynomial preserves congruences, for any term operation $h(x, y_1, \ldots, y_k)$ and any $\mathbf{a}, \mathbf{b} \in \mathbb{A}_M^k$ such that $(\mathbf{a}[i], \mathbf{b}[i]) \in \theta$ for $i \in [k]$, we have $(h^{\mathbf{a}}(x), h^{\mathbf{b}}(x)) \in \theta$ for any x. This of course implies $\zeta(\theta, \underline{1}) \geq \theta$. On the other hand, let $f(x, y) = r(y, x)$. Then as we saw before, $f^0(x) = f(x, 0) = r(0, x) = h_3(x)$ and $f^2(x) = f(x, 2) = r(2, x) = h_2(x)$, and $f^0(\underline{1}) \subseteq \theta$, while $f^2(\underline{1}) \not\subseteq \theta$. This means that $(0, 2) \notin \zeta(\theta, \underline{1})$ and so $\zeta(\theta, \underline{1}) \subset \underline{1}$. For the first centralizer it suffices to demonstrate that the condition in the definition of quasi-centrailizer is satisfied for pairs of twin polynomials produced by r, t of the form $(r(a, x), r(b, x))$, $(r(x, a), r(x, b))$, $(t(x, a_1, a_2), t(x, b_1, b_2))$, $(t(a_1, x, a_2), t(b_1, x, b_2))$, $(t(a_1, a_2, x), t(b_1, b_2, x))$, which can be verified directly.

Note that the equality $\zeta(\underline{0}, \theta) = \underline{1}$ explains why prime intervals $(\underline{0}_i, \theta_i), (\underline{0}_j, \theta_j)$ in Example 22 cannot be separated. For that the relation $\mathrm{pr}_{ij} R$ has to contain tuples $(a, b), (c, d)$ such that $(a, c) \in \zeta(\underline{0}_i, \theta_i)$ while $(b, d) \notin \zeta(\underline{0}_j, \theta_j)$, which is impossible. ◇

5 The Algorithm

In this section we introduce the reductions used in the algorithm, and then explain the algorithm itself.

5.1 Decomposition of CSPs

Let R be a binary relation, a subdirect 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 $A_1, \ldots, A_k$ are the α-blocks of $\mathbb{A}$, then there are also k γ-blocks of $\mathbb{B}$ and they can be labeled $B_1, \ldots, B_k$ in such a way that

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

Lemma 24. *Let $R, \mathbb{A}, \mathbb{B}$ be as above and $\alpha, \beta \in \mathsf{Con}(\mathbb{A})$, $\gamma, \delta \in \mathsf{Con}(\mathbb{B})$, with $\alpha \prec \beta$, $\gamma \prec \delta$. If (α, β) and (γ, δ) cannot be separated, then R is $\zeta(\alpha,\beta)\zeta(\gamma,\delta)$-aligned.*

Lemma 24 provides a way to decompose CSP instances. Let $\mathcal{I} = (V, \mathcal{C})$ be a (2,3)-minimal instance from 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 $v \in V$. Also, let $W \subseteq V$ and congruences $\alpha_v, \beta_v \in \mathsf{Con}(\mathbb{A}_v)$ for $v \in W$ be such that $\alpha_v \prec \beta_v$, and for any $v, w \in W$ the intervals (α_v, β_v) and (α_w, β_w) cannot be separated in $R^{\{v,w\}}$.

Denoting $\zeta_v = \zeta(\alpha_v, \beta_v)$ for $v \in W$ we see that there is a one-to-one correspondence between ζ_v- and ζ_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 ζ_u-, ζ_v- and ζ_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 ζ_v-blocks.

Lemma 25. *Let $\mathcal{I}, W, \alpha_v, \beta_v$ 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 a ζ_v-block.*

Example 26. Consider the following simple CSP instance from CSP($\mathbb{A}_M$), where $\mathbb{A}_M$ is the algebra introduced in Example 14, and R is the relation introduced in Example 22: $\mathcal{I} = (V = \{v_1, v_2, v_3, v_4, v_5\}, \{C^1 = \langle \mathbf{s}_1 = (v_1, v_2, v_3), R_1 \rangle, C^2 = \langle \mathbf{s}_2 = (v_2, v_4, v_5), R_2 \rangle\}$, where $R_1 = R_2 = R$. To make the instance (2,3)-minimal we run the appropriate local propagation algorithm on it. First, such an algorithm adds new binary constraints $C^{\{v_i,v_j\}} = \langle (v_i, v_j), R^{\{v_i,v_j\}} \rangle$ for $i, j \in [5]$ starting with $R^{\{v_i,v_j\}} = \mathbb{A}_M \times \mathbb{A}_M$. It then iteratively removes pairs from these relations that do not satisfy the (2,3)-minimality condition. Similarly, it tightens the original constraint relations if they violate the conditions of (2,3)-minimality. This algorithm does not change constraints C^1, C^2, and the new binary relations are as follows: $R^{\{v_1,v_2\}} = R^{\{v_2,v_4\}} = R^{\{v_1,v_4\}} = \theta$, $R^{\{v_1,v_3\}} = R^{\{v_2,v_3\}} = R^{\{v_2,v_5\}} = R^{\{v_4,v_5\}} = R^{\{v_1,v_5\}} = R^{\{v_3,v_4\}} = Q$, and $R^{\{v_3,v_5\}} = S$, where

$$Q = \mathrm{pr}_{13} R = \begin{pmatrix} 0\,0\,1\,1\,2\,2 \\ 0\,1\,0\,1\,0\,2 \end{pmatrix}, \qquad S = \begin{pmatrix} 0\,0\,1\,1\,0\,2\,2 \\ 0\,1\,0\,1\,2\,0\,2 \end{pmatrix}.$$

For convenience let the domain of v_i be denoted by $\mathbb{A}_i$, its elements by $0_i, 1_i, 2_i$, and the congruences of $\mathbb{A}_i$ by $\underline{0}_i, \theta_i, \underline{1}_i$.

Let $W = \{v_1, v_2, v_4\}$, $\alpha_i = \theta_i, \beta_i = \underline{1}_i$ for $v_i \in W$. We have $\zeta_i = \zeta(\alpha_i, \beta_i) = \theta_i = \alpha_i$. Then, as was observed in Example 23, the prime interval (α_i, β_i) cannot be separated from (α_j, β_j) for $v_i, v_j \in W$. Therefore by Lemma 25 the instance $\mathcal{I}_W = (\{v_1, v_2, v_4\}, \{C^1_W = \langle (v_1, v_2), \mathrm{pr}_{v_1 v_2} R_1 \rangle, C^2_W = \langle (v_2, v_4), \mathrm{pr}_{v_2 v_4} R_2 \rangle\})$ can be decomposed into a disjoint union of two instances:
$\mathcal{I}_1 = (\{v_1, v_2, v_4\}, \{\langle (v_1, v_2), Q_1 \rangle, \langle (v_2, v_4), Q_2 \rangle),$
$\mathcal{I}_2 = (\{v_1, v_2, v_4\}, \{\langle (v_1, v_2), \{(2_1, 2_2)\} \rangle, \langle (v_2, v_4), \{(2_2, 2_4)\}\} \rangle),$
where $Q_1 = \{0_1, 1_1\} \times \{0_2, 1_2\}, Q_2 = \{0_2, 1_2\} \times \{0_4, 1_4\}$. ◇

5.2 Block-Minimality

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 $\mathbb{A}$. For instance, the algebra $\mathbb{A}_M$ from Example 14 is subdirectly irreducible, because it has the smallest nontrivial congruence, θ. 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.

Lemma 25 allows us to use a new type of consistency of a CSP instance, block-minimality, which is key for our algorithm. In a certain sense it is similar to the standard local consistency, as it is also defined through a family of relations that have to be consistent in a certain way. However, block-minimality is not quite 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. This is achieved either by allowing for decomposing a subinstance into instances over smaller domains as in Lemma 25, or by replacing large domains with their quotient algebras.

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 $C = \langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \ldots, v_k)$, the set $\mathcal{C}_{\overline{\alpha}}$ includes the constraint $\langle \mathbf{s}, R/_{\overline{\alpha}}\rangle$, where $R/_{\overline{\alpha}} = \{(\mathbf{a}[v_1]^{\alpha_{v_1}}, \ldots, \mathbf{a}[v_k]^{\alpha_{v_k}}) \mid \mathbf{a} \in R\}$.

We start with several definitions. Let $\mathcal{I} = (V, \mathcal{C})$ be a (2,3)-minimal instance and let $\{R^X \mid X \subseteq V, |X| = 2\}$ be the relations introduced after Lemma 24. Let $\mathcal{U}^{\mathcal{I}}$ denote the set of triples (v, α, β) such that $v \in V$, $\alpha, \beta \in \mathsf{Con}(\mathbb{A}_v)$, and $\alpha \prec \beta$. For every $(v, \alpha, \beta) \in \mathcal{U}^{\mathcal{I}}$, let $W_{v,\alpha\beta}$ denote the set of all variables $w \in V$ such that (α, β) and (γ, δ) cannot be separated in $R^{\{v,w\}}$ for some $\gamma, \delta \in \mathsf{Con}(\mathbb{A}_w)$ with $(w, \gamma, \delta) \in \mathcal{U}^{\mathcal{I}}$. Sets of the form $W_{v,\alpha\beta}$ are called *coherent sets*. Let $\mathcal{Z}^{\mathcal{I}}$ denote the set of triples $(v, \alpha, \beta) \in \mathcal{U}^{\mathcal{I}}$, for which $\zeta(\alpha, \beta)$ is the full relation.

We say that algebra $\mathbb{A}_v$ is *semilattice free* if it does not contain semilattice edges. 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 $\mu_v^Y = \underline{0}_v$ otherwise.

Instance $\mathcal{I}$ is said to be *block-minimal* if for every $(v, \alpha, \beta) \in \mathcal{U}^{\mathcal{I}}$ the following conditions hold:

(B1) if $(v, \alpha, \beta) \notin \mathcal{Z}^{\mathcal{I}}$, the problem $\mathcal{I}_{W_{v,\alpha\beta}}$ is minimal;
(B2) if $(v, \alpha, \beta) \in \mathcal{Z}^{\mathcal{I}}$, for every $C = \langle \mathbf{s}, R\rangle \in \mathcal{C}$ the problem $\mathcal{I}_{W_{v,\alpha\beta}}/_{\overline{\mu}^Y}$, where $Y = \mathsf{MAX}(\mathcal{I}) - \mathbf{s}$, is minimal;
(B3) if $(v, \alpha, \beta) \in \mathcal{Z}^{\mathcal{I}}$, then for every $(w, \gamma, \delta) \in \mathcal{U}^{\mathcal{I}} - \mathcal{Z}^{\mathcal{I}}$ the problem $\mathcal{I}_{W_{v,\alpha\beta}}/_{\overline{\mu}^Y}$, where $Y = \mathsf{MAX}(\mathcal{I}) - (W_{v,\alpha\beta} \cap W_{w,\gamma\delta})$ is minimal.

Example 27. Let us consider again the instance $\mathcal{I}$ from Example 26. There we found all its binary solutions, and now we use them to find coherent sets and to verify that this instance is block-minimal. For the instance $\mathcal{I}$ we have $\mathcal{U}^{\mathcal{I}} = \{(v_i, \underline{0}_i, \theta_i), (v_i, \theta_i, \underline{1}_i) \mid i \in [5]\}$ and $\mathcal{Z}^{\mathcal{I}} = \{(v_i, \underline{0}_i, \theta_i) \mid i \in [5]\}$. As we noticed in Example 22, interval $(\underline{0}_i, \theta_i)$ cannot be separated from $(\underline{0}_j, \theta_j)$ for any $i, j \in [5]$. Therefore, for each $i \in [5]$ we have $W_{v_i, \underline{0}_i \theta_i} = V$. Also, it was shown in Example 22 that $(\theta_i, \underline{1}_i)$ cannot be separated from $(\theta_j, \underline{1}_j)$ for $\{i, j\} = \{1, 2\}$ and $\{i, j\} = \{2, 4\}$, while $\{\theta_i, \underline{1}_i\}$ can be separated from $(\theta_j, \underline{1}_j)$ and $(\underline{0}_j, \theta_j)$ for $i \in \{1, 2, 4\}$ and $j \in \{3, 5\}$. Therefore, for $i \in \{1, 2, 4\}$ we have $W_{v_i, \theta_i \underline{1}_i} = \{v_1, v_2, v_4\}$. Finally, $(\theta_3, \underline{1}_3)$ can be separated from $(\underline{0}_5, \theta_5), (\theta_5, \underline{1}_5)$ by considering the relation S from Example 26, and $(\underline{0}_i, \theta_i)$, $i \in \{1, 2, 4\}$ can be separated from $(\theta_3, \underline{1}_3)$ by considering the relation Q. Therefore, $W_{v_i, \theta_i \underline{1}_i} = \{v_i\}$ for $i \in \{3, 5\}$.

Now we check the conditions (B1)–(B3) for $\mathcal{I}$. Since $\zeta(\theta_i, \underline{1}_i) = \theta_i$, $i \in [5]$, for the coherent sets $W_{v_i, \theta_i \underline{1}_i}$ we need to check condition (B1). If $i = 3, 5$ this condition is trivially true, as the set of solutions of $\mathcal{I}$ on every 1-element set of variables is $\mathbb{A}_M$. Consider $W_{v_1, \theta_1 \underline{1}_1} = \{v_1, v_2, v_4\}$; as is easily seen, a triple (a_1, a_2, a_4) is a solution of $\mathcal{I}_{\{v_1, v_2, v_4\}}$ if and only if $(a_1, a_2), (a_1, a_4), (a_2, a_4) \in \theta$. Condition (B1) amounts to saying that for any constraint of $\mathcal{I}$, say, C^1, and any tuple $\mathbf{a}$ from its constraint relation R_1, the projection $\mathrm{pr}_{v_1 v_2} \mathbf{a}$ can be extended to a solution of $\mathcal{I}_{\{v_1, v_2, v_4\}}$. Since $\mathrm{pr}_{v_1 v_2} \mathbf{a} \in \theta$, this can always be done. For other constraints (B1) is verified in a similar way.

Next consider $W_{v_1, \underline{0}_1 \theta_1} = V$. As $\zeta(\underline{0}_1, \theta_1) = \underline{1}_1$, we have to verify conditions (B2),(B3). We consider condition (B2) for constraint C^1, the remaining cases are similar. The monolith of $\mathbb{A}_M$ is θ, therefore in the first case $Y = \{v_4, v_5\}$ and $\mu^Y_{v_i}$ is the equality relation for $i \in \{1, 2, 3\}$ and $\mu^Y_{v_4} = \theta_4, \mu^Y_{v_5} = \theta_5$. The instance $\mathcal{I}/_{\overline{\mu}^Y}$ is as follows: $\mathcal{I}/_{\overline{\mu}^Y} = (V, \{C'^1 = \langle \mathbf{s}_1, R_1 \rangle, C'^2 = \langle \mathbf{s}_2, R_2/_{\overline{\mu}} \rangle\})$. The constraint relation, of C'^1 equals R_1, as $\mu^Y_{v_i} = \underline{0}_i$ for $i \in \{1, 2, 3\}$. The constraint relation of C'^2 then equals $R'_2 = R_2/_{\overline{\mu}^Y} = \{(0, 0^\theta, 0^\theta), (1, 0^\theta, 0^\theta), (2, 2^\theta, 0^\theta), (2, 2^\theta, 2^\theta)\}$. Now, for every tuple $\mathbf{a} \in R_1$, and for every tuple $\mathbf{b} \in R'_2$ we need to find solutions φ, ψ of $\mathcal{I}/_{\overline{\mu}^Y}$ such that $\varphi(v_i) = \mathbf{a}[v_i]$ for $i \in \{1, 2, 3\}$ and $\psi(v_i) = \mathbf{b}[v_i]$ for $i \in \{2, 4, 5\}$. If $\mathbf{a}[v_2] \in \{0, 1\}$ ($\mathbf{b}[v_2] \in \{0, 1\}$) then extending $\mathbf{a}$ by $\varphi(v_4) = \varphi(v_5) = 0^\theta$ (extending $\mathbf{b}$ by $\psi(v_1) = \psi(v_3) = 0$) gives solutions of $\mathcal{I}/_{\overline{\mu}^Y}$. If $\mathbf{a}[v_2] = 2$ ($\mathbf{b}[v_2] = 2$), then tuples $\mathbf{a}, \mathbf{b}$ can be extended by $\varphi(v_4) = \varphi(v_5) = 2^\theta$ and by $\psi(v_1) = \psi(v_3) = 2$ to solutions of $\mathcal{I}/_{\overline{\mu}^Y}$. $\diamond$

Next we observe that establishing block-minimality can be efficiently reduced to solving a polynomial number of strictly smaller instances. First, observe that $W_{v, \alpha\beta}$ can be large, even equal to V, as we saw in Example 27. However if $(v, \alpha, \beta) \notin \mathcal{Z}^{\mathcal{I}}$, by Lemma 25 the problem $\mathcal{I}_{W_{v, \alpha\beta}}$ splits into a union of disjoint problems over smaller domains, and so its minimality can be established by recursing to strictly smaller problems. On the other hand, if $(v, \alpha, \beta) \in \mathcal{Z}^{\mathcal{I}}$ then $\mathcal{I}_{W_{v, \alpha\beta}}$ may not split into such a union. Since we need an efficient procedure of establishing block-minimality, this explains the complications introduced in conditions (B2),(B3). In the case of (B2) $\mathcal{I}_{W_{v, \alpha\beta}}/_{\overline{\mu}^Y}$ (see the definition of block-minimality) can be solved for each tuple $\mathbf{a} \in R$ by fixing the values from this tuple. Taking the quotient algebras of the remaining domains guarantees that we recurse to a strictly smaller instance. In the case of (B3) $\mathcal{I}_{W_{v,\alpha\beta}\cap W_{w,\gamma\delta}}/_{\overline{\mu}^Y}$ splits into disjoint subproblems, and we branch on those strictly smaller subproblems.

Lemma 28. *Let $\mathcal{I} = (V, \mathcal{C})$ be a (2,3)-minimal instance. Then by solving a quadratic number of strictly smaller CSPs $\mathcal{I}$ can be transformed to an equivalent block-minimal instance $\mathcal{I}'$.*

5.3 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. Algorithm 1 gives a more formal description of the solution algorithm.

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 $\zeta(\underline{0}_v, \mu_v) = \underline{1}_v$. Let $\mu^*_v = \mu_v$ if $v \in \mathsf{MAX}(\mathcal{I}) \cap \mathsf{Center}(\mathcal{I})$ and $\mu^*_v = \underline{0}_v$ otherwise.

Semilattice Free Domains If no domain of $\mathcal{I}$ contains a semilattice edge then by Proposition 19 $\mathcal{I}$ can be solved in polynomial time, using the few subalgebras algorithm, as shown in [51, 21].

Small Centralizers If $\mu^*_v = \underline{0}_v$ for all $v \in V$, block-minimality guarantees the existence of a solution, as Theorem 29 shows, and we can use Lemma 28 to solve the instance.

Theorem 29. *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.*

Proof of Theorem 29 is the most technically involved part of our result.

Large Centralizers Suppose that $\mathsf{MAX}(\mathcal{I}) \cap \mathsf{Center}(\mathcal{I}) \neq \varnothing$. 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 tightened in the process, we start solving the new problem from scratch. To check global 1-minimality, for each $v \in V$ and every $a \in \mathbb{A}_v/_{\mu^*_v}$, we need to find a solution of the instance, or show it does not exists. To this end, add the constraint $\langle (v), \{a\} \rangle$ to $\mathcal{I}/_{\overline{\mu}^*}$. The resulting problem belongs to CSP($\mathbb{A}$), since $\mathbb{A}_v$ is idempotent, and hence $\{a\}$ is a subalgebra of $\mathbb{A}_v/_{\mu^*_v}$. Then we establish (2,3)-minimality and block minimality of the resulting problem. Let us denote it $\mathcal{I}'$. There are two possibilities. First, if $\mathsf{size}(\mathcal{I}') < \mathsf{size}(\mathcal{I})$ then $\mathcal{I}'$ is a problem strictly smaller than $\mathcal{I}$ and can be solved by recursivly calling Algorithm 1 on $\mathcal{I}'$. If $\mathsf{size}(\mathcal{I}') = \mathsf{size}(\mathcal{I})$ then, as all the domains $\mathbb{A}_v$ of maximal size for $v \in \mathsf{Center}(\mathcal{I})$ are replaced with their quotient algebras, there is $w \notin \mathsf{Center}(\mathcal{I})$ such that $|\mathbb{A}_w| = \mathsf{size}(\mathcal{I})$ and $\mathbb{A}_w$ is not semilattice free. Therefore for every $u \in \mathsf{Center}(\mathcal{I}')$, for the corresponding domain $\mathbb{A}'_u$ we have $|\mathbb{A}'_u| < \mathsf{size}(\mathcal{I}) = \mathsf{size}(\mathcal{I}')$. Thus, $\mathsf{MAX}(\mathcal{I}') \cap \mathsf{Center}(\mathcal{I}') = \varnothing$, and $\mathcal{I}'$ has a solution by Theorem 29.

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 $\mu_v^* = \underline{0}_v$, or, if $\mu_v^* = \mu_v$, there is $b \in \varphi(v)$ such that $\{a, b\}$ is a semilattice edge. Take $b \in \mathbb{A}_v/_{\mu_v^*}$ such that $\{a, b\}$ is a semilattice edge in $\mathbb{A}_v/_{\mu_v^*}$ for some $a \in \mathbb{A}_v/_{\mu_v^*}$. Since $\mathcal{I}/_{\overline{\mu}^*}$ is globally 1-minimal, there is a solution $\varphi_{v,b}$ such that $\varphi_{v,b}(v) = b$.

Step 3. We apply the transformation of $\mathcal{I}$ suggested by Maroti in [62]. For a solution φ of $\mathcal{I}/_{\overline{\mu}^*}$ by $\mathcal{I} \cdot \varphi$ we denote the instance $(V, \mathcal{C}_\varphi)$ given by the rule: for every $C = \langle \mathbf{s}, R\rangle \in \mathcal{C}$ the set $\mathcal{C}_\varphi$ contains a constraint $\langle \mathbf{s}, R \cdot \varphi\rangle$. To construct $R \cdot \varphi$ choose a tuple $\mathbf{b} \in R$ such that $\mathbf{b}[v]^{\mu_v^*} = \varphi(v)$ for all $v \in \mathbf{s}$; this is possible because φ is a solution of $\mathcal{I}/_{\overline{\mu}^*}$. Then set $R \cdot \varphi = \{\mathbf{a} \cdot \mathbf{b} \mid \mathbf{a} \in R\}$. By the results of [62] it can be shown that the instance $\mathcal{I} \cdot \varphi$ has a solution if and only if $\mathcal{I}$ does. Let $\mathcal{I}' = (\ldots(\mathcal{I} \cdot \varphi_{v_1,b_1}) \cdot \ldots) \cdot \varphi_{v_\ell,b_\ell}$, where $\varphi_{v_1,b_1}, \ldots, \varphi_{v_\ell,b_\ell}$ are the solutions chosen in Step 2. We have $\mathsf{size}(\mathcal{I}') < \mathsf{size}(\mathcal{I})$.

This last case can be summarized as the following

Theorem 30. *If $\mathcal{I}/_{\overline{\mu}^*}$ is globally 1-minimal, then $\mathcal{I}$ can be reduced in polynomial time to a strictly smaller instance over an algebra satisfying the conditions of the Dichotomy Conjecture.*

Algorithm 1 Procedure $\mathsf{SolveCSP}$

Require: A CSP instance $\mathcal{I} = (V, \mathcal{C})$ from CSP($\mathbb{A}$)
Ensure: A solution of $\mathcal{I}$ if one exists, 'NO' otherwise

```
1: if all the domains are semilattice free then
2:    Solve I using the few subpowers algorithm and RETURN the answer
3: end if
4: Transform I to a subdirectly irreducible, block-minimal, and (2,3)-minimal instance
5: μ*_v = μ_v for v ∈ MAX(I) ∩ Center(I) and μ*_v = 0_v otherwise
6: I* = I/μ̄*
7: %% Check the 1-minimality of I*
8: for every v ∈ V and a ∈ A_v/μ*_v do
9:    I' = I*_(v,a)    %% Add the constraint ⟨(v), {a}⟩ fixing the value of v to a
10:   Transform I' to a subdirectly irreducible, (2,3)-minimal instance I''
11:   if size(I'') < size(I) then
12:      Call SolveCSP on I'' and flag a if I'' has no solution
13:   else
14:      Establish block-minimality of I''; if the problem changes, return to Step 10
15:      If the resulting instance is empty, flag element a
16:   end if
17: end for
18: If there are flagged values, tighten the instance by removing the flagged elements
    and start over
19: Use Theorem 30 to reduce I to an instance I' with size(I') < size(I)
20: Call SolveCSP on I' and RETURN the answer
```

Example 31. We illustrate the algorithm SolveCSP on the instance from Example 26. Recall that the domain of each variable is $\mathbb{A}_M$, its monolith is θ, and $\zeta(\underline{0},\theta)$ is the full relation. This means that $\mathsf{size}(\mathcal{I})=3$, $\mathsf{MAX}(\mathcal{I})=V$ and $\mathsf{Center}(\mathcal{I})=V$, as well. Therefore we are in the case of large centralizers. Set $\mu^*_{v_i}=\theta_i$ for each $i\in[5]$ and consider the problem $\mathcal{I}/_{\overline{\mu}*}=(V,\{C^*_1=\langle\mathbf{s}_1,R^*_1\rangle,C^*_2=\langle\mathbf{s}_2,R^*_2\rangle),$ where $R^*=\{(0^\theta,0^\theta,0^\theta),(2^\theta,2^\theta,0^\theta),(2^\theta,2^\theta,2^\theta)\}$. It is an easy excercise to show that this instance is globally 1-minimal (every value 0^θ can be extended to the all-0^θ solution, and every value 2^θ can be extended to the all-2^θ solution). This completes *Step 1*. For every variable v_i we choose $b\in\mathbb{A}_M/_\theta$ such that for some $a\in\mathbb{A}_M/_\theta$ the pair $\{a,b\}$ is a semilattice edge. Since $\mathbb{A}_M/_\theta$ is a 2-element semilattice, setting $b=0^\theta$ and $a=2^\theta$ is the only choice. Therefore all solutions $\varphi_{v_i,0^\theta}$ in our case can be chosen to be φ, where $\varphi(v_i)=0^\theta$; and *Step 2* is completed. For *Step 3* first note that in $\mathbb{A}_M$ the operation r plays the role of multiplication $\cdot$ defined in Lemma 20. Then for each of the constraints C^1,C^2 choose a representative $\mathbf{a}_1\in R_1\cap(\varphi(v_1)\times\varphi(v_2)\times\varphi(v_3))=R_1\cap\{0,1\}^3$, $\mathbf{a}_2\in R_2\cap(\varphi(v_2)\times\varphi(v_4)\times\varphi(v_5))=R_2\cap\{0,1\}^3$, and set $\mathcal{I}'=(\{v_1,\dots,v_5\},\{C'_1=\langle(v_1,v_2,v_3),R'_1\rangle,C'_2=\langle(v_2,v_4,v_5),R'_2\rangle\})$, where $R'_1=r(R_1,\mathbf{a})$, $R'_2=r(R_2,\mathbf{b})$. Since $r(2,0)=r(2,1)=0$, regardless of the choice of $\mathbf{a},\mathbf{b}$ in our case $R'_1\subseteq R_1,R'_2\subseteq R_2$, and are invariant with respect to the affine operation of $\mathbb{Z}_2$. Therefore the instance $\mathcal{I}'$ can be viewed as a system of linear equations over $\mathbb{Z}_2$ (this system is actually empty in our case), and can be easily solved. ◇

Using Lemma 28 and Theorems 29,30 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 CSP($\mathbb{A}$).

References

1. Barto, L.: The dichotomy for conservative constraint satisfaction problems revisited. In: LICS. pp. 301–310 (2011)
2. Barto, L.: The collapse of the bounded width hierarchy. J. of Logic and Comput. (2014)
3. Barto, L., Kozik, M.: Absorbing subalgebras, cyclic terms, and the Constraint Satisfaction Problem. Logical Methods in Computer Science 8(1) (2012)
4. Barto, L., Kozik, M.: Constraint satisfaction problems solvable by local consistency methods. J. ACM 61(1), 3:1–3:19 (2014)
5. Barto, L., Kozik, M.: Robustly solvable constraint satisfaction problems. SIAM J. Comput. 45(4), 1646–1669 (2016)
6. Barto, L., Kozik, M.: Absorption in universal algebra and CSP. In: The Constraint Satisfaction Problem: Complexity and Approximability, pp. 45–77 (2017)
7. Barto, L., Krokhin, A.A., Willard, R.: Polymorphisms, and how to use them. In: The Constraint Satisfaction Problem: Complexity and Approximability, pp. 1–44 (2017)

8. Barto, L., Pinsker, M., Opršal, J.: The wonderland of reflections. Israel J. of Math. (2018), to appear
9. Berman, J., Idziak, P., Marković, P., McKenzie, R., Valeriote, M., Willard, R.: Varieties with few subalgebras of powers. Trans. Amer. Math. Soc. 362(3), 1445–1473 (2010)
10. Bodnarchuk, V., Kaluzhnin, L., Kotov, V., Romov, B.: Galois theory for post algebras. I. Kibernetika 3, 1–10 (1969)
11. Börner, F., Bulatov, A.A., Chen, H., Jeavons, P., Krokhin, A.A.: The complexity of constraint satisfaction games and QCSP. Inf. Comput. 207(9), 923–944 (2009)
12. Bulatov, A.: A dichotomy theorem for constraints on a three-element set. In: FOCS. pp. 649–658, (2002)
13. Bulatov, A.A.: Tractable conservative constraint satisfaction problems. In: LICS. pp. 321–330 (2003)
14. Bulatov, A.A.: A graph of a relational structure and constraint satisfaction problems. In: LICS. pp. 448–457 (2004)
15. Bulatov, A.A.: H-coloring dichotomy revisited. Theor. Comp. Sci. 349(1), 31–39 (2005)
16. Bulatov, A.A.: A dichotomy theorem for constraint satisfaction problems on a 3-element set. J. ACM 53(1), 66–120 (2006)
17. Bulatov, A.A.: Complexity of conservative constraint satisfaction problems. ACM Trans. Comput. Log. 12(4), 24 (2011)
18. Bulatov, A.A.: The complexity of the Counting Constraint Satisfaction Problem. J. ACM 60(5), 34:1–34:41 (2013)
19. Bulatov, A.A.: Conservative constraint satisfaction re-revisited. J. of Comp. and Syst. Sci. 82(2), 347–356 (2016)
20. Bulatov, A.A.: Graphs of finite algebras, edges, and connectivity. CoRR. abs/1601.07403 (2016)
21. Bulatov, A.A.: Graphs of relational structures: restricted types. In: LICS pp. 642–651 (2016)
22. Bulatov, A.A.: A dichotomy theorem for nonuniform CSPs. CoRR. abs/1703.03021 (2017)
23. Bulatov, A.A.: A dichotomy theorem for nonuniform CSPs. In: FOCS pp. 319–330 (2017)
24. Bulatov, A.A., Dalmau, V.: A simple algorithm for Mal'tsev constraints. SIAM J. Comput. 36(1), 16–27 (2006)
25. Bulatov, A.A., Jeavons, P., Krokhin, A.A.: Classifying the complexity of constraints using finite algebras. SIAM J. Comput. 34(3), 720–742 (2005)
26. Bulatov, A.A., Krokhin, A.A., Larose, B.: Dualities for constraint satisfaction problems. In: Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]. pp. 93–124 (2008)
27. Bulatov, A.A., Valeriote, M.: Recent results on the algebraic approach to the CSP. In: Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]. pp. 68–92 (2008)
28. Cai, J., Chen, X.: Complexity of counting CSP with complex weights. In: STOC. pp. 909–920 (2012)
29. Chandra, A.K., Merlin, P.M.: Optimal implementation of conjunctive queries in relational data bases. In: STOC. pp. 77–90 (1977)
30. Cooper, M.C., Zivny, S.: Hybrid tractable classes of constraint problems. In: The Constraint Satisfaction Problem: Complexity and Approximability, pp. 113–135 (2017)

31. Creignou, N., Khanna, S., Sudan, M.: Complexity Classifications of Boolean Constraint Satisfaction Problems, SIAM Monographs on Discrete Mathematics and Applications, vol. 7. SIAM (2001)
32. Dalmau, V.: Generalized majority-minority operations are tractable. Logical Methods in Computer Science 2(4) (2006)
33. Dalmau, V., Kozik, M., Krokhin, A.A., Makarychev, K., Makarychev, Y., Oprsal, J.: Robust algorithms with polynomial loss for near-unanimity CSPs. In: SODA. pp. 340–357 (2017)
34. Dechter, R.: Constraint processing. Morgan Kaufmann Publishers (2003)
35. Dyer, M.E., Greenhill, C.S.: The complexity of counting graph homomorphisms. Random Struct. Algorithms 17(3-4), 260–289 (2000)
36. Feder, T., Vardi, M.: Monotone monadic SNP and constraint satisfaction. In: STOC. pp. 612–622 (1993)
37. Feder, T., Vardi, M.: 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)
38. Feder, T., Hell, P., Klein, S., Motwani, R.: List partitions. SIAM J. Discrete Math. 16(3), 449–478 (2003)
39. Feder, T., Hell, P., Tucker-Nally, K.: Digraph matrix partitions and trigraph homomorphisms. Discr. Appl. Math. 154(17), 2458–2469 (2006)
40. Flum, J., Frick, M., Grohe, M.: Query evaluation via tree-decompositions. J. ACM 49(6), 716–752 (2002)
41. Freese, R., McKenzie, R.: Commutator theory for congruence modular varieties, London Math. Soc. Lecture Notes, vol. 125. London (1987)
42. Geiger, D.: Closed systems of function and predicates. Pacific J. of Math. pp. 95–100 (1968)
43. Gottlob, G., Leone, N., Scarcello, F.: A comparison of structural CSP decomposition methods. Artif. Intell. 124(2), 243–282 (2000)
44. Gottlob, G., Leone, N., Scarcello, F.: Hypertree decompositions and tractable queries. J. Comput. Syst. Sci. 64(3), 579–627 (2002)
45. Grätzer, G.: Universal algebra. Springer, 2nd edn. (2008)
46. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. J. ACM 54(1), 1:1–1:24 (2007)
47. Grohe, M., Marx, D.: Constraint solving via fractional edge covers. ACM Trans. Algorithms 11(1), 4:1–4:20 (2014)
48. Hell, P., Nešetřil: Graphs and homomorphisms, Oxford Lecture Series in Mathematics and its Applications, vol. 28. Oxford University Press (2004)
49. Hell, P., Nešetřil, J.: On the complexity of H-coloring. Journal of Combinatorial Theory, Ser.B 48, 92–110 (1990)
50. Hobby, D., McKenzie, R.: The Structure of Finite Algebras, Contemporary Mathematics, vol. 76. American Mathematical Society, Providence, R.I. (1988)
51. Idziak, P.M., Markovic, P., McKenzie, R., Valeriote, M., Willard, R.: Tractability and learnability arising from algebras with few subpowers. SIAM J. Comput. 39(7), 3023–3037 (2010)
52. Jeavons, P., Cohen, D.A., Gyssens, M.: Closure properties of constraints. J. ACM 44(4), 527–548 (1997)
53. Jeavons, P., Cohen, D., Cooper, M.: Constraints, consistency and closure. Artificial Intelligence 101(1-2), 251–265 (1998)
54. Jerrum, M.: Counting constraint satisfaction problems. In: The Constraint Satisfaction Problem: Complexity and Approximability, pp. 205–231 (2017)

55. Klíma, O., Tesson, P., Thérien, D.: Dichotomies in the complexity of solving systems of equations over finite semigroups. Theory Comput. Syst. 40(3), 263–297 (2007)
56. Kolmogorov, V., Krokhin, A.A., Rolínek, M.: The complexity of general-valued CSPs. SIAM J. Comput. 46(3), 1087–1110 (2017)
57. Krokhin, A.A., Zivny, S.: The complexity of valued csps. In: The Constraint Satisfaction Problem: Complexity and Approximability, pp. 233–266 (2017)
58. Larose, B., Zádori, L.: Taylor terms, constraint satisfaction and the complexity of polynomial equations over finite algebras. IJAC 16(3), 563–582 (2006)
59. Mackworth, A.: Consistency in networks of relations. Artif. Intel. 8, 99–118 (1977)
60. Marcin Kozik, Andrei Krokhin, M.V., Willard, R.: Characterizations of several Maltsev conditions. Algebra Universalis 73(3), 205224 (2015)
61. Markovic, P.: The complexity of CSPs on a 4-element set, oral communication, (2011)
62. Maróti, M.: Tree on top of Malcev (2011), manuscript, available at `http://www.math.u-szeged.hu/ mmaroti/pdf/200x%20Tree%20on %20top%20of%20Maltsev.pdf`
63. Maróti, M., McKenzie, R.: Existence theorems for weakly symmetric operations. Algebra Universalis 59(3-4), 463–489 (2008)
64. Marx, D.: Tractable hypergraph properties for constraint satisfaction and conjunctive queries. J. ACM 60(6), 42:1–42:51 (2013)
65. Post, E.: The two-valued iterative systems of mathematical logic, Annals Mathematical Studies, vol. 5. Princeton University Press (1941)
66. Raghavendra, P.: Optimal algorithms and inapproximability results for every CSP? In: STOC. pp. 245–254 (2008)
67. Reingold, O.: Undirected connectivity in log-space. J. ACM 55(4), 17:1–17:24 (2008)
68. Schaefer, T.: The complexity of satisfiability problems. In: STOC. pp. 216–226 (1978)
69. Thapper, J., Zivny, S.: The complexity of finite-valued CSPs. J. ACM 63(4), 37:1–37:33 (2016)
70. Zhuk, D.: On CSP dichotomy conjecture. In: Arbeitstagung Allgemeine Algebra AAA'92. p. 32 (2016)
71. Zhuk, D.: The proof of CSP dichotomy conjecture for 5-element domain. In: Arbeitstagung Allgemeine Algebra AAA'91. (2016)
72. Zhuk, D.: A proof of CSP dichotomy conjecture. In: FOCS. pp. 331–342 (2017)