58th Annual IEEE Symposium on Foundations of Computer Science

A dichotomy theorem for nonuniform CSPs

Andrei A. Bulatov
School of Computing Science
Simon Fraser University
Burnaby, Canada
Email: abulatov@sfu.ca

***Abstract*—In a non-uniform Constraint Satisfaction problem $\mathrm{CSP}(\Gamma)$, where Γ is a set of relations on a finite set A, the goal is to find an assignment of values to variables subject to constraints imposed on specified sets of variables using the relations from Γ. The Dichotomy Conjecture for the non-uniform CSP states that for every constraint language Γ the problem $\mathrm{CSP}(\Gamma)$ is either solvable in polynomial time or is NP-complete. It was proposed by Feder and Vardi in their seminal 1993 paper. In this paper we confirm the Dichotomy Conjecture.**

***Keywords*-Constraint Satisfaction problem, dichotomy conjecture**

I. Introduction

In a Constraint Satisfaction Problem (CSP) the question is to decide whether or not it is possible to satisfy a given set of constraints. One of the standard ways to specify a constraint is to require that a combination of values of a certain set of variables belongs to a given relation. If the constraints allowed in a problem have to come from some set Γ of relations, such a restricted problem is referred to as a *nonuniform CSP* and denoted $\mathrm{CSP}(\Gamma)$. The set Γ is then called a *constraint language*. Nonuniform CSPs not only provide a powerful framework ubiquitous across a wide range of disciplines from theoretical computer science to computer vision, but also admit natural and elegant reformulations such as the homomorphism problem and characterizations, in particular, as the class of problems equivalent to a logic class MMSNP. Many different versions of the CSP have been studied across various fields. These include CSPs over infinite sets, counting CSPs (and related Holant problem, and the problem of computing partition functions), several variants of optimization CSPs, valued CSPs, quantified CSPs, and numerous related problems. The reader is referred to the recent book [46] for a survey of the state-of-the art in some of these areas. In this paper we, however, focus on the decision nonuniform CSP and its complexity.

A systematic study of the complexity of nonuniform CSPs was started by Schaefer in 1978 [54] who showed that for every constraint language Γ over a 2-element set the problem $\mathrm{CSP}(\Gamma)$ is either solvable in polynomial time or is NP-complete. Schaefer also asked about the complexity of $\mathrm{CSP}(\Gamma)$ for languages over larger sets. The next step in the study of nonuniform CSPs was made in the seminal paper by Feder and Vardi [32], [33], who apart from considering numerous aspects of the problem, posed the *Dichotomy Conjecture* that states that for every finite constraint language Γ over a finite set the problem $\mathrm{CSP}(\Gamma)$ is either solvable in polynomial time or is NP-complete. This conjecture has become a focal point of the CSP research and most of the effort in this area revolves to some extent around the Dichotomy Conjecture.

The complexity of the CSP in general and the Dichotomy Conjecture in particular has been studied by several research communities using a variety of methods, each contributing an important aspect of the problem. The CSP has been an established area in artificial intelligence for decades, and apart from developing efficient general methods of solving CSPs researchers tried to identify tractable fragments of the problem [31]. A very important special case of the CSP, the (Di)Graph Homomorphism problem and the H-Coloring problem have been actively studied in the graph theory community, see, e.g. [37], [38] and subsequent works by Hell, Feder, Bang-Jensen, Rafiey and others. Homomorphism duality introduced in these works has been very useful in understanding the structure of constraint problems. The CSP plays a major role and has been successfully studied in database theory, logic and model theory [44], [43], [36], although the version of the problem mostly used there is not necessarily nonuniform. Logic games and strategies are now a standard tool in most of CSP algorithms. An interesting approach to the Dichotomy Conjecture through long codes was suggested by Kun and Szegedy [47]. Brown-Cohen and Raghavendra proposed to study the conjecture using techniques based on decay of correlations [11]. In this paper we use the algebraic structure of the CSP, which is briefly discussed

0272-5428/17 $31.00 © 2017 IEEE
DOI 10.1109/FOCS.2017.37

next.

The most effective approach to the study of the CSP turned out to be the *algebraic approach* that associates every constraint language with its (universal) algebra of polymorphisms. This approach was first developed in a series of papers by Jeavons and coauthors [40], [41], [42] and then refined by Bulatov, Krokhin, Barto, Kozik, Maroti, Zhuk and others [5], [8], [6], [25], [15], [27], [50], [51], [55], [56]. While the complexity of $\mathrm{CSP}(\Gamma)$ has been already solved for some interesting classes of structures such as graphs [37], the algebraic approach allowed the researchers to confirm the Dichotomy Conjecture in a number of more general cases: for languages over a set of size up to 7 [13], [16], [49], [56], so called conservative languages [14], [17], [18], [3], and some classes of digraphs [7]. It also helped to design the main classes of CSP algorithms [6], [23], [20], [10], [39], and to refine the exact complexity of the CSP [1], [8], [30], [48].

In this paper we confirm the Dichotomy Conjecture for arbitrary languages over finite sets. More precisely we prove the following

Theorem 1: For any finite constraint language Γ over a finite set the problem $\mathrm{CSP}(\Gamma)$ is either solvable in polynomial time or is NP-complete.

The same result has been independently obtained by Dmitriy Zhuk and is also presented at FOCS 2017.

The proved criterion matches the algebraic form of the Dichotomy Conjecture suggested in [25]. The hardness part of the conjecture has been known for long time. Therefore the main achievement of this paper is a polynomial time algorithm for problems satisfying the tractability condition from [25].

Using the algebraic language we can state the result in a stronger form. Let $\mathbb{A}$ be a finite idempotent algebra and let $\mathrm{CSP}(\mathbb{A})$ denote the union of problems $\mathrm{CSP}(\Gamma)$ such that every term operation of $\mathbb{A}$ is a polymorphism of Γ. Problem $\mathrm{CSP}(\mathbb{A})$ is no longer a nonuniform CSP, and Theorem 1 allows for problems $\mathrm{CSP}(\Gamma) \subseteq \mathrm{CSP}(\mathbb{A})$ to have different solution algorithms even when $\mathbb{A}$ meets the tractability condition. We show that the solution algorithm only depends on the algebra $\mathbb{A}$.

Theorem 2: For a finite idempotent algebra that satisfies the conditions of the Dichotomy Conjecture there is a uniform solution algorithm for $\mathrm{CSP}(\mathbb{A})$.

An interesting question arising from Theorems 1,2 is known as the *Meta-problem*: Given a constraint language or a finite algebra, decide whether or not it satisfies the conditions of the theorems. The answer to this question is not quite trivial, for a thorough study of the Meta-problem see [29], [35].

We start with introducing the terminology and notation for CSPs that is used throughout the paper and reminding the basics of the algebraic approach. Then in Section IV we introduce the key ingredients used in the algorithm: separation of congruences and quasi-centralizers. Then in Section V we apply these concepts to CSPs, first, to demonstrate how quasi-centralizers help to decompose an instance into smaller subinstances, and, second, to introduce a new kind of minimality condition for CSPs, *block minimality*. After that we state the main results used by the algorithm and describe the algorithm itself. The full version of this paper is found in [22].

II. CSP, UNIVERSAL ALGEBRA AND THE DICHOTOMY CONJECTURE

For a detailed introduction to the CSP and the algebraic approach to its structure the reader is referred to a very recent and very nice survey by Barto et al. [9]. Basics of universal algebra can be learned from the textbook [28]. In preliminaries to this paper we therefore focus on what is needed for our result.

A. The CSP

The 'AI' formulation of the CSP best fits our purpose. Fix a finite set A and let Γ be a *constraint language* over A, that is, a set — not necessarily finite — of relations over A. The (*nonuniform*) *Constraint Satisfaction Problem* (*CSP*) associated with language Γ is the problem $\mathrm{CSP}(\Gamma)$, in which, an *instance* is a pair $(V, \mathcal{C})$, where V is a set of variables; and $\mathcal{C}$ is a set of *constraints*, i.e. pairs $\langle \mathbf{s}, R\rangle$, where $\mathbf{s} = (v_1, \dots, v_k)$ is a tuple of variables from V, the *constraint scope*, and $R \in \Gamma$, the k-ary *constraint relation*. We always assume that relations are given explicitly by a list of tuples. The way constraints are represented does not matter if Γ is finite, but it may change the complexity of the problems for infinite languages. The goal is to find a *solution*, i.e., a mapping $\varphi : V \to A$ such that for every constraint $\langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\varphi(\mathbf{s}) \in R$.

B. Algebraic methods in the CSP

Jeavons et al. in [40], [41] were the first to observe that higher order symmetries of constraint languages, called polymorphisms, play a significant role in the study of the complexity of the CSP. A *polymorphism* of a relation R over A is an operation $f(x_1, \dots, x_k)$ on A such that for any choice of $\mathbf{a}_1, \dots, \mathbf{a}_k \in R$ we have $f(\mathbf{a}_1, \dots, \mathbf{a}_k) \in R$. If this is the case we also say that f *preserves* R, or that R is *invariant* with respect to f. A polymorphism of a constraint language Γ is an operation that is a polymorphism of every $R \in \Gamma$.

Theorem 3 ([40], [41]): For constraint languages Γ, Δ, where Γ is finite, if every polymorphism of Δ is also a polymorphism of Γ, then $\mathrm{CSP}(\Gamma)$ is polynomial time reducible to $\mathrm{CSP}(\Delta)$.[1]

Listed below are several types of polymorphisms that occur frequently throughout the paper. The presence of each of these polymorphisms imposes strong restrictions on the structure of invariant relations that can be used in designing a solution algorithm. Some of such results we will mention later.

– *Semilattice* operation is a binary operation $f(x, y)$ such that $f(x, x) = x$, $f(x, y) = f(y, x)$, and $f(x, f(y, z)) = f(f(x, y), z)$ for all $x, y, z \in A$;

– k-ary *near-unanimity* operation is a k-ary operation $u(x_1, \dots, x_k)$ such that $u(y, x, \dots, x) = u(x, y, x, \dots, x) = \dots = u(x, \dots, x, y) = x$ for all $x, y \in A$; a ternary near-unanimity operation m is called a *majority* operation, it satisfies the equations $m(y, x, x) = m(x, y, x) = m(x, x, y) = x$;

– *Mal'tsev* operation is a ternary operation $h(x, y, z)$ satisfying the equations $h(x, y, y) = h(y, y, x) = x$ for all $x, y \in A$; the *affine* operation $x - y + z$ of an Abelian group is a special case of a Mal'tsev operation;

– k-ary *weak near-unanimity* operation is a k-ary operation w that satisfies the same equations as a near-unanimity operation $w(y, x, \dots, x) = \dots = w(x, \dots, x, y)$, except for the last one ($= x$).

To illustrate the effect of polymorphisms on the structure of invariant relations we give a few examples that involve polymorphisms introduced above. First, we need some terminology and notation.

By $[n]$ we denote the set $\{1, \dots, n\}$. For sets $A_1, \dots, A_n$ tuples from $A_1 \times \dots \times A_n$ are denoted in boldface, say, $\mathbf{a}$; the ith component of $\mathbf{a}$ is referred to as $\mathbf{a}[i]$. An n-ary relation R over sets $A_1, \dots, A_n$ is any subset of $A_1 \times \dots \times A_n$. For $I = \{i_1, \dots, i_k\} \subseteq [n]$ by $\mathrm{pr}_I\mathbf{a}, \mathrm{pr}_I R$ we denote the *projections* $\mathrm{pr}_I\mathbf{a} = (\mathbf{a}[i_1], \dots, \mathbf{a}[i_k])$, $\mathrm{pr}_I R = \{\mathrm{pr}_I\mathbf{a} \mid \mathbf{a} \in R\}$ of tuple $\mathbf{a}$ and relation R. If $\mathrm{pr}_i R = A_i$ for each $i \in [n]$, relation R is said to be a *subdirect product* of $A_1 \times \dots \times A_n$. Sometimes it is convenient to label the coordinate positions of relations by elements of some set other than $[n]$, e.g. by variables of a CSP.

Example 1: (1) Let $\vee$ be the binary operation of disjunction on $\{0, 1\}$, as is easily seen, it is a semilattice operation. The following property of relations invariant under $\vee$ helps solving the corresponding CSP: A relation R contains the tuple $(1, \dots, 1)$ whenever for each coordinate position R contains a tuple with a 1 in that position. Similarly, relations invariant under other semilattice operations on larger sets always contain a sort of 'maximal' tuple.

(2) By the results of [2] a tuple $\mathbf{a}$ belongs to a (n-ary) relation R invariant under a k-ary near-unanimity operation if and only if for every $(k-1)$-element set $I \subseteq [n]$ we have $\mathrm{pr}_I\mathbf{a} \in \mathrm{pr}_I R$. In particular, if f is the majority operation on $\{0, 1\}$ given by $(x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$, and R is a relation on $\{0, 1\}$, then $\mathbf{a} \in R$ if and only if $(\mathbf{a}[i], \mathbf{a}[j]) \in \mathrm{pr}_{ij} R$. This property easily gives rise to a reduction of the corresponding CSP to 2-SAT.

(3) If $m(x, y, z) = x - y + z$ is the affine operation of, say, $\mathbb{Z}_p$, p prime, then relations invariant with respect to m are exactly those that can be represented as solutions sets of systems of linear equations over $\mathbb{Z}_p$, and the corresponding CSP can be solved by Gaussian Elimination. $\diamond$

The next step in discovering more structure behind nonuniform CSPs was made in [25], where universal algebras were brought into the picture. A *(universal) algebra* is a pair $\mathbb{A} = (A, F)$ consisting of a set A, the *universe* of $\mathbb{A}$, and a set F of operations on A. Operations from F (called *basic*) together with operations that can be obtained from them by means of composition are called the *term* operations of $\mathbb{A}$.

Algebras allow for a more general definition of CSPs than is used above. Let $\mathrm{CSP}(\mathbb{A})$ denote the class of nonuniform CSPs $\{\mathrm{CSP}(\Gamma) \mid \Gamma \subseteq \mathsf{Inv}(F), \Gamma \text{ finite}\}$, where $\mathsf{Inv}(F)$ denotes the set of all relations invariant with respect to all operations from F. Note that the tractability of $\mathrm{CSP}(\mathbb{A})$ can be understood in two ways: as the existence of a polynomial-time algorithm for every $\mathrm{CSP}(\Gamma)$ from this class, or as the existence of a uniform polynomial-time algorithm for all such problems. One of the implications of our results is that these two types of tractability are the same. From the formal standpoint we will use the stronger one.

C. Structural features of universal algebras

We use some structural elements of algebras, the main of which are subalgebras, congruences, and quotient algebras. For $B \subseteq A$ and an operation f on A by $f|_B$ we denote the restriction of f on B. Algebra $\mathbb{B} = (B, \{f|_B \mid f \in F\})$ is a *subalgebra* of $\mathbb{A}$ if $f(b_1, \dots, b_k) \in B$ for any $b_1, \dots, b_k \in B$ and any $f \in F$.

Congruences play a very significant role in our algorithm, and we discuss them in more detail. A *congruence*

[1]Using the $s-t$-Connectivity algorithm by Reingold [53] this reduction can be improved to a log-space one.

is an equivalence relation $\alpha \in \mathsf{Inv}(F)$. This means that for any operation $f \in F$ and any $(a_1, b_1), \ldots, (a_k, b_k) \in \alpha$ it holds $(f(a_1, \ldots, a_k), f(b_1, \ldots, b_k)) \in \alpha$. Hence one can define an algebra on $A/_\alpha$, the set of α-blocks, by setting $f/_\alpha(a_1^\alpha, \ldots, a_k^\alpha) = (f(a_1, \ldots, a_k))/_\alpha$ for $a_1, \ldots, a_k \in A$, where a^α denotes the α-block containing a. The algebra $\mathbb{A}/_\alpha$ is called the *quotient algebra modulo* α.

Example 2: The following are examples of congruences and quotient algebras.

(1) Let $\mathbb{A}$ be any algebra. Then the equality relation $\underline{0}_\mathbb{A}$ and the full binary relation $\underline{1}_\mathbb{A}$ on $\mathbb{A}$ are congruences of $\mathbb{A}$. The quotient algebra $\mathbb{A}/_{\underline{0}_\mathbb{A}}$ is $\mathbb{A}$ itself, while $\mathbb{A}/_{\underline{1}_\mathbb{A}}$ is a 1-element algebra.

(2) Let $\mathbb{L}_n$ be an n-dimensional vector space and $\mathbb{L}'$ its k-dimensional subspace, $k \leq n$. The binary relation π given by: $(\overline{a}, \overline{b}) \in \pi$ iff $\overline{a}, \overline{b}$ have the same orthogonal projection on $\mathbb{L}'$, is a congruence of $\mathbb{L}_n$ and $\mathbb{L}_n/_\pi$ is $\mathbb{L}'$. (3) The next example will be our running example throughout the paper. Let $A = \{0, 1, 2\}$, and $\mathbb{A}_M$ is 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 1(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 $r/_\theta(2^\theta, 2^\theta) = 2^\theta$. $\diamond$

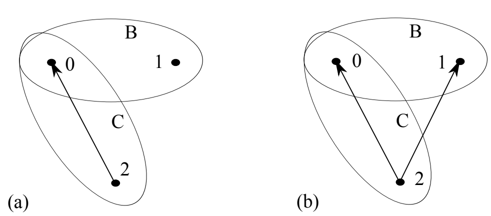

Fig. 1. (a) Algebra $\mathbb{A}_M$; (b) algebra $\mathbb{A}_N$. Dots represent elements, ovals represent subalgebras, and arrows represent semilattice edges (see Section III-B).

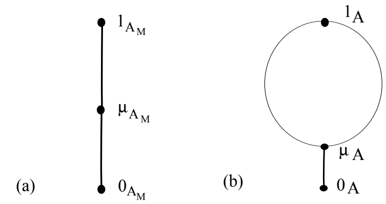

Fig. 2. (a) The congruence lattice of algebra $\mathbb{A}_M$; (b) congruence lattice of a subdirectly irreducible algebra.

The (ordered) set of all congruences of $\mathbb{A}$ is denoted by $\mathsf{Con}(\mathbb{A})$. This set is actually a lattice, that is, the operations of meet $\wedge$ and join $\vee$ can be defined so that $\alpha \wedge \beta$ is the greatest lower bound of $\alpha, \beta \in \mathsf{Con}(\mathbb{A})$ and $\alpha \vee \beta$ is the least upper bound of α, β. Fig. 2(a) shows $\mathsf{Con}(\mathbb{A}_M)$ for the algebra $\mathbb{A}_M$ from Example 2(3). By $\mathsf{HS}(\mathbb{A})$ we denote the set of all quotient algebras of all subalgebras of $\mathbb{A}$.

D. The Dichotomy Conjecture

The results of [25] reduce the dichotomy conjecture to idempotent algebras. An algebra $\mathbb{A} = (A, F)$ is said to be *idempotent* if every operation $f \in F$ satisfies the equation $f(x, \ldots, x) = x$. If $\mathbb{A}$ is idempotent, then all the *constant* relations $\{(a)\}$ are invariant under F. Therefore studying CSPs over idempotent algebras is the same as studying the CSPs that allow all constant relations. Another useful property of idempotent algebras is that every block of every its congruence is a subalgebra. We now can state the algebraic version of the dichotomy theorem.

Theorem 4: For a finite idempotent algebra $\mathbb{A}$ the following are equivalent:

(1) $\mathrm{CSP}(\mathbb{A})$ is solvable in polynomial time;

(2) $\mathbb{A}$ has a weak near-unanimity term operation;

(3) every algebra from $\mathsf{HS}(\mathbb{A})$ has a nontrivial term operation (that is, not a *projection*, an operation of the form $f(x_1, \ldots, x_k) = x_i$);

Otherwise $\mathrm{CSP}(\mathbb{A})$ is NP-complete.

The hardness part of this theorem is proved in [25]; the equivalence of (2) and (3) was proved in [24] and [52]. The equivalence of (1) to (2) and (3) is the main result of this paper. In the rest of the paper we assume all algebras to satisfy conditions (2),(3).

III. Bounded width and the few subpowers algorithm

Leaving aside occasional combinations thereof, there are only two standard types of algorithms solving the CSP. In this section we give a brief introduction into them.

A. CSPs of bounded width

Algorithms of the first kind are based on the idea of local propagation, that is formally described below.

Let $\mathcal{P} = (V, \mathcal{C})$ be a CSP instance. For $W \subseteq V$ by $\mathcal{P}_W$ we denote the *restriction* of $\mathcal{P}$ 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 $\mathcal{P}_W$ 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 $\mathcal{P}$ 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 $\mathcal{P}$ is said to be *(2,3)-consistent* if it has a *(2,3)-strategy*, that is, a collection of relations R^X, $X \subseteq V$, $|X| = 2$ satisfying the following conditions:
– for every $X \subseteq V$ with $|X| \le 2$, $\mathrm{pr}_{\mathbf{s}\cap X} R^X \subseteq \mathcal{S}_X$;
– for every $X = \{u, v\} \subseteq V$, any $w \in V - X$, and any $(a, b) \in R^X$, there is $c \in \mathbb{A}_w$ such that $(a, c) \in R^{\{u,w\}}$ and $(b, c) \in R^{\{v,w\}}$.

Let the collection of relations R^X be denoted by $\mathcal{R}$. A tuple $\mathbf{a}$ whose entries are indexed with elements of $W \subseteq V$ and such that $\mathrm{pr}_X \mathbf{a} \in R^X$ for any $X \subseteq W$, $|X| = 2$, will be called *$\mathcal{R}$-compatible*. If a (2,3)-consistent instance $\mathcal{P}$ with a (2,3)-strategy $\mathcal{R}$ satisfies the additional condition
– for every constraint $C = \langle \mathbf{s}, R\rangle$ of $\mathcal{P}$ every tuple $\mathbf{a} \in R$ is $\mathcal{R}$-compatible,

it is called *(2,3)-minimal*. For $k \in \mathbb{N}$, $(k, k+1)$-strategies, $(k, k+1)$-consistency, and $(k, k+1)$-minimality are defined in a similar way replacing 2,3 with $k, k+1$.

Instance $\mathcal{P}$ 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 $\varphi \in \mathcal{S}$ such that $\varphi(\mathbf{s}) = \mathbf{a}$. Similarly, $\mathcal{P}$ 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, (2,3)-consistent, or (2,3)-minimal instance in polynomial time using the standard constraint propagation algorithms (see, e.g. [31]). 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 $\mathrm{CSP}(\mathbb{A})$ 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 $\mathrm{CSP}(\mathbb{A})$. Establishing minimality amounts to solving the problem and therefore 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, $\mathrm{CSP}(\Gamma)$ (or $\mathrm{CSP}(\mathbb{A})$) is said to have *bounded width* if for some k every $(k, k+1)$-minimal instance from $\mathrm{CSP}(\Gamma)$ (or $\mathrm{CSP}(\mathbb{A})$) has a solution. Problems of bounded width are very well studied, see the older survey [26] and a more recent paper [4].

Theorem 5 ([4], [20], [15], [45]): 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 of arity k for every $k \ge 3$;

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

B. Omitting semilattice edges and the few subpowers property

The second type of CSP algorithms can be viewed as a generalization of Gaussian elimination, although, it utilizes just one property also used by Gaussian elimination: the set of solutions of a system of linear equations or a CSP has a set of generators of size polynomial in the number of variables. The property that for every instance $\mathcal{P}$ of $\mathrm{CSP}(\mathbb{A})$ its solution space $\mathcal{S}$ has a set of generators of polynomial size is nontrivial, because there are only exponentially many such sets, while, as is easily seen CSPs may have up to double exponentially many different sets of solutions. Formally, an algebra $\mathbb{A} = (A, F)$ has *few subpowers* if for every n there are only exponentially many n-ary relations in $\mathsf{Inv}(F)$.

Algebras with few subpowers are well studied and the CSP over such an algebra has a polynomial-time solution algorithm, see, [10], [39]. In particular, such algebras admit a characterization in terms of the existence of a term operation with special properties, an *edge* term. We need only a subclass of algebras with few subpowers that appeared in [20] and is defined as follows.

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 2(3) is a semilattice edge, and the operation r of $\mathbb{A}_M$ witnesses that.

Proposition 6 ([20]): If an idempotent algebra $\mathbb{A}$ has no semilattice edges, it has few subpowers, and therefore $\mathrm{CSP}(\mathbb{A})$ 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 7 ([19]): For any idempotent algebra $\mathbb{A}$ there is a binary term operation xy of $\mathbb{A}$ (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 7. The operation r of the algebra $\mathbb{A}_M$ from Example 2(3) is not a semilattice operation (for instance, it does not satisfy the equation $r(x,y) = r(y,x)$), but it satisfies the conditions of Lemma 7.

IV. Congruence separation and centralizers

In this section we introduce two of the key ingredients of our algorithm.

A. 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 [51]. We make use of some ideas from that paper in the next section.

Let $f(x_1,\dots,x_k,y_1,\dots,y_\ell)$ be a $k+\ell$-ary term operation of an algebra $\mathbb{A}$ and $b_1,\dots,b_\ell \in \mathbb{A}$. The operation $g(x_1,\dots,x_k) = f(x_1,\dots,x_k,b_1,\dots,b_\ell)$ is called a *polynomial* of $\mathbb{A}$. The term 'polynomial' refers to usual polynomials. Indeed, if $\mathbb{A}$ is a ring, its polynomials as just defined are the same as polynomials in the regular sense. A polynomial for which $k = 1$ is said to be a *unary* polynomial.

While polynomials of $\mathbb{A}$ do not have to be polymorphisms of relations from $\mathsf{Inv}(F)$, congruences and unary polynomials are in a special relationship. More precisely, it is a well known fact that an equivalence relation over $\mathbb{A}$ is a congruence if and only if it is preserved by all the unary polynomials of $\mathbb{A}$. 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. 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 3: The unary polynomials of the algebra $\mathbb{A}_M$ from Example 2(3) 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 2(3)). 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 $h_2(\theta) \subseteq \underline{0}$. $\diamond$

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 $\mathbb{A}_1 \times \cdots \times \mathbb{A}_n$. 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.

Example 4: 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 $(\theta_3, \underline{1}_3)$. Also, the interval $(\theta_3, \underline{1}_3)$ can be separated from $(\underline{0}_1, \theta_1), (\underline{0}_2, \theta_2)$. Indeed, consider the polynomial

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

then $h_2(\underline{1}) \not\subseteq \theta$, but $h_2(\theta) \subseteq \underline{0}$. 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. $\diamond$

B. Quasi-Centralizers

The second ingredient introduced 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 [34], 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 $f(x, y_1, \dots, y_k)$, 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 an 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.) It can be shown that the relation $\zeta(\alpha, \beta)$ is always a congruence of $\mathbb{A}$ and its effect on the structure of algebra $\mathbb{A}$ is illustrated by the following statement.

Lemma 8: Let $\zeta(\alpha,\beta) = \underline{1}_{\mathbb{A}}$, $a,b,c \in \mathbb{A}$ and $(b,c) \in \beta$. Then $(ab, ac) \in \alpha$, where multiplication is as in Lemma 7.

Example 5: In the algebra $\mathbb{A}_M$, see Example 2(3), 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 $h(x, y_1, \dots, y_k)$ 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

$$f^0(x) = f(x,0) = r(0,x) = h_3(x),$$
$$f^2(x) = f(x,2) = r(2,x) = h_2(x),$$

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 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.

Interestingly, Lemma 8 implies that if we change the operation r in just one point, it has a profound effect on the quasi-centralizer $\zeta(\underline{0}, \theta)$. Let $\mathbb{A}_N$ be the same algebra as $\mathbb{A}_M$ with operations r', t' defined in the same way as r, t, except $r'(1,2) = 1$ replacing the value $r(1,2) = 0$. In this case $\{1,2\}$ is also a semilattice edge, see Fig. 1(b). Let again $f(x,y) = r'(y,x)$ and $a = 0, b = 2$. This time we have

$$f^0(x) = f(x,0) = r'(0,x) = h'_3(x),$$
$$f^2(x) = f(x,2) = r'(2,x) = h'_2(x),$$

where $h'_3(x) = 0$ for all $x \in \{0,1,2\}$ and $h'_2(0) = 0, h'_2(1) = 1$ showing that $f^0(\theta) \subseteq \underline{0}$, while $f^2(\theta) \not\subseteq \underline{0}$. $\diamond$

Fig. 3(a),(b) shows the effect of large quasi-centralizers $\zeta(\alpha, \beta)$ on the structure of algebra $\mathbb{A}$, which is a generalization of the phenomena observed in Example 5. Dots there represent α-blocks (assume α is the equality relation), ovals represent β-blocks, let they be B and C, and such that there is at least one semilattice edge between B and C. If $\zeta(\alpha,\beta)$ is the full relation, Lemmas 7 and 8 imply that for any $a \in B$ and any $b, c \in C$ we have $ab = ac$, and so ab is the only element of C such that $\{a, ab\}$ is a semilattice edge (represented by arrows). In other words, we have a mapping from B to C that can also be shown injective. We will use this mapping to lift any solution with a value from B to a solution with a value from C.

V. The algorithm

In this section we introduce the reductions used in the algorithm, and then explain the algorithm itself. The reductions heavily use the algebraic structure of the domains of an instance, and the structure of the instance itself.

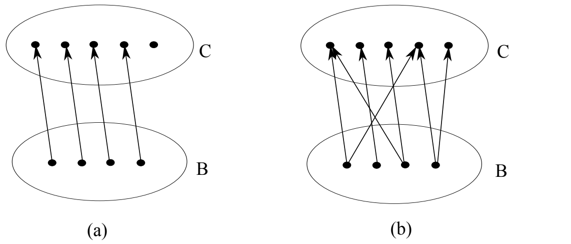

Fig. 3. (a) $\zeta(\alpha,\beta)$ is the full relation; (b) $\zeta(\alpha,\beta)$ is not the full relation

A. Decomposition of CSPs

We have seen in the previous section that big centralizers impose strong restrictions on the structure of an algebra. We start this section showing that small centralizers restrict the structure 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 $A_1,\dots,A_k$ 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

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

Lemma 9: 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 9 provides a way to decompose CSP instances. Let $\mathcal{P}=(V,\mathcal{C})$ be a (2,3)-minimal instance from CSP($\mathbb{A}$). We will always assume that a (2,3)-consistent or (2,3)-minimal instance has a constraint $C^X=\langle X,\mathcal{S}_X\rangle$ for every $X\subseteq V$, $|X|=2$. So, $\mathcal{C}$ contains a constraint $C^{\{v,w\}}=\langle(v,w),R^{\{v,w\}}\rangle$ for every $v,w\in V$, and these relations form a (2,3)-strategy for $\mathcal{P}$. 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 $\mathcal{P}_W$ can be split into several instances, whose domains are ζ_v-blocks.

Lemma 10: Let $\mathcal{P},W,\alpha_v,\beta_v$ for each $v\in W$, be as above. Then $\mathcal{P}_W$ can be decomposed into a collection of instances $\mathcal{P}_1,\dots,\mathcal{P}_k$, k constant, $\mathcal{P}_i=(W,\mathcal{C}_i)$ such that every solution of $\mathcal{P}_W$ is a solution of one of the $\mathcal{P}_i$ and for every $v\in W$ its domain in $\mathcal{P}_i$ is a ζ_v-block.

Example 6: Consider the following simple CSP instance from CSP($\mathbb{A}_M$), where $\mathbb{A}_M$ is the algebra introduced in Example 2(3), and R is the relation introduced in Example 4: $\mathcal{P}=(V=\{v_1,v_2,v_3,v_4,v_5\},\{C^1=\langle\mathbf{s}_1=(v_1,v_2,v_3),R_1\rangle,C^2=\langle\mathbf{s}_2=(v_2,v_4,v_5),R_2\rangle\}$, 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. It is not hard to see that this algorithm does not change constraints C^1,C^2, and that the new binary relations are as follows: $R^{\{v_1,v_2\}}=R^{\{v_2,v_4\}}=R^{\{v_1,v_4\}}=\theta$, $R^{\{v_1,v_3\}}=R^{\{v_2,v_3\}}=R^{\{v_2,v_5\}}=R^{\{v_4,v_5\}}=Q$, and $R^{\{v_1,v_5\}}=R^{\{v_3,v_4\}}=R^{\{v_3,v_5\}}=S$, where

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

In order to distinguish elements and congruences of domains belonging to different variables let the 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 5, 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 10 the instance $\mathcal{P}_W=(\{v_1,v_2,v_4\},\{C^1_W=\langle(v_1,v_2),\mathrm{pr}_{v_1v_2}R_1\rangle,C^2_W=\langle(v_2,v_4),\mathrm{pr}_{v_2v_4}R_2\rangle\})$ can be decomposed into a disjoint union of two instances

$$\begin{aligned}\mathcal{P}_1 &= (\{v_1,v_2,v_4\},\{\langle(v_1,v_2),Q_1\rangle,\langle(v_2,v_4),Q_2\rangle),\\ \mathcal{P}_2 &= (\{v_1,v_2,v_4\},\{\langle(v_1,v_2),S_1\rangle,\langle(v_2,v_4),S_2\rangle),\end{aligned}$$

where $Q_1=\{0_1,1_1\}\times\{0_2,1_2\},Q_2=\{0_2,1_2\}\times\{0_4,1_4\}$, $S_1=\{(2_1,2_2)\},S_2=\{(2_2,2_4)\}$. $\diamond$

B. Irreducibility

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 $\mathbb{A}$, see Fig. 2(b). For instance, the algebra $\mathbb{A}_M$ from Example 2(3) 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.

C. Block-minimality

Using Lemma 10 we introduce a new type of consistency of a CSP instance, block-minimality, which will be crucial for our algorithm. In a certain sense it is similar to the standard local consistency notions, as it 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 10, or by replacing large domains with their quotient algebras.

Let α_v be a congruence of $\mathbb{A}_v$ for $v \in V$. By $\mathcal{P}/_{\overline{\alpha}}$ 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 $C = \langle \mathbf{s}, R\rangle \in \mathcal{C}$, $\mathbf{s} = (v_1, \dots, v_k)$, the set $\mathcal{C}_{\overline{\alpha}}$ includes the constraint $\langle \mathbf{s}, R/_{\overline{\alpha}}\rangle$, where $R/_{\overline{\alpha}} = \{(\mathbf{a}[v_1]^{\alpha_{v_1}}, \dots, \mathbf{a}[v_k]^{\alpha_{v_k}}) \mid \mathbf{a} \in R\}$.

Example 7: Consider the instance $\mathcal{P}$ from Example 6, and let $\alpha_{v_i} = \theta_i$ for each $i \in [5]$. Then $\mathcal{P}/_{\overline{\alpha}}$ is the instance over $\mathbb{A}_M/_\theta$ given by $\mathcal{P}/_{\overline{\alpha}} = (V, \{\langle \mathbf{s}_1, R_1/_{\overline{\alpha}}\rangle, \langle \mathbf{s}_2, R_2/_{\overline{\alpha}}\rangle\})$, where

$$R_1/_{\overline{\alpha}} = R_2/_{\overline{\alpha}} = \begin{pmatrix} 0^\theta & 2^\theta & 2^\theta \\ 0^\theta & 2^\theta & 2^\theta \\ 0^\theta & 0^\theta & 2^\theta \end{pmatrix}.$$

$\diamond$

We start with several definitions. Let $\mathcal{P} = (V, \mathcal{C})$ be a (2,3)-minimal instance and $\{R^X \mid X \subseteq V, |X| = 2\}$ is its (2,3)-strategy. Let $\mathcal{U}^{\mathcal{P}}$ 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 $(v, \alpha, \beta) \in \mathcal{U}^{\mathcal{P}}$, 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 $(w, \gamma, \delta) \in \mathcal{U}^{\mathcal{P}}$. Sets of the form $W_{v,\alpha\beta}$ will be called *coherent sets*. Let $\mathcal{Z}^{\mathcal{P}}$ denote the set of triples $(v, \alpha, \beta) \in \mathcal{U}^{\mathcal{P}}$, 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 $\mathsf{size}(\mathcal{P})$ denote the maximal size of domains of $\mathcal{P}$ that are not semilattice free and $\mathsf{MAX}(\mathcal{P})$ be the set of variables $v \in V$ such that $|\mathbb{A}_v| = \mathsf{size}(\mathcal{P})$ and $\mathbb{A}_v$ is not semilattice free. For instances $\mathcal{P}, \mathcal{P}'$ we say that $\mathcal{P}'$ is *strictly smaller* than $\mathcal{P}$ if $\mathsf{size}(\mathcal{P}') < \mathsf{size}(\mathcal{P})$. For $Y \subseteq V$ let $\mu_v^Y = \mu_v$ if $v \in Y$ and $\mu_v^Y = \underline{0}_v$ otherwise.

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

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

Example 8: Let us consider again the instance $\mathcal{P}$ from Example 6. In that example 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 $\mathcal{P}$ we have $\mathcal{U}^{\mathcal{P}} = \{(v_i, \underline{0}_i, \theta_i), (v_i, \theta_i, \underline{1}_i) \mid i \in [5]\}$ and $\mathcal{Z}^{\mathcal{P}} = \{(v_i, \underline{0}_i, \theta_i) \mid i \in [5]\}$. As we noticed in Example 4, interval $(\underline{0}, \theta_i)$ 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 4 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 6, and $(\underline{0}_i, \theta_i)$, $i \in \{1, 2, 4\}$ can be separated from $(\theta_3, \underline{1}_3)$. Therefore, for $i \in \{3, 5\}$ we have $W_{v_i,\theta_i\underline{1}_i} = \{v_i\}$.

Now we check the conditions (B1)–(B3) for $\mathcal{P}$. 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 $\mathcal{P}$ 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 $\mathcal{P}_{\{v_1,v_2,v_4\}}$ 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 $\mathcal{P}$, say, C^1, and any tuple $\mathbf{a}$ from its constraint relation R_1, the projection $\mathrm{pr}_{v_1v_2}\mathbf{a}$ can be extended to a solution of $\mathcal{P}_{\{v_1,v_2,v_4\}}$. Since $\mathrm{pr}_{v_1v_2}\mathbf{a} \in \theta$, this can always be done. For other constraints (B1) is verified in a similar way.

Now consider $W_{v_1,\underline{0}_1\theta_1} = V$. As $\zeta(v_1, \underline{0}_1, \theta_1) = \underline{1}_1$, 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 $\mu_{v_i}^Y$ 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 $\mathcal{P}/_{\overline{\mu}^Y}$ is as follows: $\mathcal{P}/_{\overline{\mu}^Y} = (V, \{C'^1 = \langle \mathbf{s}_1, R_1\rangle, C'^2 = \langle \mathbf{s}_2, R_2/_{\overline{\mu}}\rangle\})$. The constraint relation, of C'^1 equals R_1, as $\mu_{v_i} = \underline{0}_i$ for $i \in \{1,2,3\}$. The constraint relation of C'^2 then equals

$$R'_2 = R_2/_{\overline{\mu}^Y} = \left(\begin{array}{cccc} 0 & 1 & 2 & 2 \\ 0^\theta & 0^\theta & 2^\theta & 2^\theta \\ 0^\theta & 0^\theta & 0^\theta & 2^\theta \end{array}\right).$$

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 $\mathcal{P}/_{\overline{\mu}^Y}$ 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 $\mathcal{P}/_{\overline{\mu}^Y}$. 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 $\mathcal{P}/_{\overline{\mu}^Y}$. $\diamond$

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 8. However if $(v,\alpha,\beta) \notin \mathcal{Z}^\mathcal{P}$, by Lemma 10 the problem $\mathcal{P}_{W_{v,\alpha\beta}}$ 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 $(v,\alpha,\beta) \in \mathcal{Z}^\mathcal{P}$ then $\mathcal{P}_{W_{v,\alpha\beta}}$ 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) $\mathcal{P}_{W_{v,\alpha\beta}}/_{\overline{\mu}^Y}$ (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) $\mathcal{P}_{W_{v,\alpha\beta} \cap W_{w,\gamma\delta}}/_{\overline{\mu}^Y}$ splits into disjoint subproblems, and we branch on those strictly smaller subproblems.

Lemma 11: Let $\mathcal{P} = (V, \mathcal{C})$ be a (2,3)-minimal instance. Then by solving a quadratic number of strictly smaller CSPs $\mathcal{P}$ can be transformed to an equivalent block-minimal instance $\mathcal{P}'$.

D. 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 $\mathcal{P} = (V, \mathcal{C})$ be a subdirectly irreducible, (2,3)-minimal instance. Let $\mathsf{Center}(\mathcal{P})$ 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 $v \in \mathsf{MAX}(\mathcal{P}) \cap \mathsf{Center}(\mathcal{P})$ and $\mu^*_v = \underline{0}_v$ otherwise.

a) Semilattice free domains: If no domain of $\mathcal{P}$ contains a semilattice edge then by Proposition 6 $\mathcal{P}$ can be solved in polynomial time, using the few subalgebras algorithm, as shown in [39], [20].

b) Small centralizers: If $\mu^*_v = \underline{0}_v$ for all $v \in V$, block-minimality guarantees the existence of a solution, as Theorem 12 shows, and we can use Lemma 11 to solve the instance.

Theorem 12: If $\mathcal{P}$ is subdirectly irreducible, (2,3)-minimal, block-minimal, and $\mathsf{MAX}(\mathcal{P}) \cap \mathsf{Center}(\mathcal{P}) = \varnothing$, then $\mathcal{P}$ has a solution.

Proof of Theorem 12 is the most technically involved part of our result.

c) Large centralizers: Suppose that $\mathsf{MAX}(\mathcal{P}) \cap \mathsf{Center}(\mathcal{P}) \neq \varnothing$. In this case the algorithm proceeds in three steps.

Step 1. Consider the problem $\mathcal{P}/_{\overline{\mu}^*}$. 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 $\mathcal{P}/_{\overline{\mu}^*}$. The resulting problem belongs to $\mathrm{CSP}(\mathbb{A})$, 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 $\mathcal{P}'$. There are two possibilities. First, if $\mathsf{size}(\mathcal{P}') < \mathsf{size}(\mathcal{P})$ then $\mathcal{P}'$ is a problem strictly smaller than $\mathcal{P}$ and can be solved by recursivly calling Algorithm 1 on $\mathcal{P}'$. If $\mathsf{size}(\mathcal{P}') = \mathsf{size}(\mathcal{P})$ then, as all the domains $\mathbb{A}_v$ of maximal size for $v \in \mathsf{Center}(\mathcal{P})$ are replaced with their quotient algebras, there is $w \notin \mathsf{Center}(\mathcal{P})$ such that $|\mathbb{A}_v| = \mathsf{size}(\mathcal{P})$ and $\mathbb{A}_v$ is not semilattice free. Therefore for every $u \in \mathsf{Center}(\mathcal{P}')$, for the corresponding domain $\mathbb{A}'_u$ we have $|\mathbb{A}'_u| < \mathsf{size}(\mathcal{P}) = \mathsf{size}(\mathcal{P}')$. Thus, $\mathsf{MAX}(\mathcal{P}') \cap \mathsf{Center}(\mathcal{P}') = \varnothing$, and $\mathcal{P}'$ has a solution by Theorem 12.

Step 2. We find a solution φ of $\mathcal{P}/_{\overline{\mu}^*}$ satisfying the following condition: For every $v \in \mathsf{Center}(\mathcal{P})$ 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 $v \in \mathsf{Center}(\mathcal{P})$ and $b \in \mathbb{A}_v/_{\mu^*_v}$ such that $\{a, b\}$ is a semilattice edge in $\mathbb{A}_v/_{\mu^*_v}$ for some $a \in \mathbb{A}_v/_{\mu^*_v}$. Since $\mathcal{P}/_{\overline{\mu}^*}$ is globally 1-minimal, there is a solution $\varphi_{v,b}$ such that $\varphi_{v,b}(v) = b$. Setting

$$\varphi = \varphi_{v_1,b_1} \cdot (\varphi_{v_2,b_2} \cdot \ldots (\varphi_{v_\ell,a_\ell}) \ldots),$$

where $(v_1, a_1), \ldots, (v_\ell, a_\ell)$ is a list of pairs specified above, we obtain a required solution by Lemma 7.

CASE 3. We apply the transformation of $\mathcal{P}$ suggested by Maroti in [51]. By $\mathcal{P} \cdot \varphi$ 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 $\mathcal{P}/_{\overline{\mu}^*}$. Then set $R \cdot \varphi = \{\mathbf{a} \cdot \mathbf{b} \mid \mathbf{a} \in R\}$. By the results of [51] and Lemma 8 the instance $\mathcal{P} \cdot \varphi$ has a solution if and only if $\mathcal{P}$ does, and also $\mathsf{size}(\mathcal{P} \cdot \varphi) < \mathsf{size}(\mathcal{P})$.

This last case can be summarized as the following

Theorem 13: If $\mathcal{P}/_{\overline{\mu}^*}$ is globally 1-minimal, then $\mathcal{P}$ 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 $\mathcal{P} = (V, \mathcal{C})$ from CSP($\mathbb{A}$)
Ensure: A solution of $\mathcal{P}$ if one exists, 'NO' otherwise

1: **if** all the domains are semilattice free **then**
2:     Solve $\mathcal{P}$ using the few subpowers algorithm and RETURN the answer
3: **end if**
4: Transform $\mathcal{P}$ to a subdirectly irreducible, block-minimal, and (2,3)-minimal instance
5: $\mu^*_v = \mu_v$ for $v \in \mathsf{MAX}(\mathcal{P}) \cap \mathsf{Center}(\mathcal{P})$ and $\mu^*_v = \underline{0}_v$ otherwise
6: $\mathcal{P}^* = \mathcal{P}/_{\overline{\mu}^*}$
7: *%% Check the 1-minimality of* $\mathcal{P}^*$
8: **for** every $v \in V$ and $a \in \mathbb{A}_v/_{\mu^*_v}$ **do**
9:     $\mathcal{P}' = \mathcal{P}^*_{(v,a)}$ *%% Add the constraint* $\langle (v), \{a\}\rangle$ *fixing the value of* v *to* a
10:     Transform $\mathcal{P}'$ to a subdirectly irreducible, (2,3)-minimal instance $\mathcal{P}''$
11:     If $\mathsf{size}(\mathcal{P}'') < \mathsf{size}(\mathcal{P})$ call $\mathsf{SolveCSP}$ on $\mathcal{P}''$ and flag a if $\mathcal{P}''$ has no solution
12:     Establish block-minimality of $\mathcal{P}''$; if the problem changes, return to Step 10
13:     If the resulting instance is empty, flag element a
14: **end for**
15: If there are flagged values, tighten the instance by removing the flagged elements and start over
16: Use Theorem 13 to reduce $\mathcal{P}$ to an instance $\mathcal{P}'$ with $\mathsf{size}(\mathcal{P}') < \mathsf{size}(\mathcal{P})$
17: Call $\mathsf{SolveCSP}$ on $\mathcal{P}'$ and RETURN the answer

Example 9: We illustrate the algorithm $\mathsf{SolveCSP}$ on the instance from Example 6. 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 $\mathsf{size}(\mathcal{P}) = 3$, $\mathsf{MAX}(\mathcal{P}) = V$ and $\mathsf{Center}(\mathcal{P}) = V$, as well. Therefore we are in the case of nontrivial centralizers. Set $\mu^*_{v_i} = \theta_i$ for each $i \in [5]$ and consider the problem $\mathcal{P}/_{\overline{\mu}*} = (V, \{C^*_1 = \langle \mathbf{s}_1, R^*_1\rangle, C^*_2 = \langle \mathbf{s}_2, R^*_2\rangle\rangle)$, where

$$R^* = \begin{pmatrix} 0^\theta & 2^\theta & 2^\theta \\ 0^\theta & 2^\theta & 2^\theta \\ 0^\theta & 2^\theta & 2^\theta \end{pmatrix}.$$

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 φ in our case can be chosen by $\varphi(v_i) = 0^\theta$; and *Step 2* is completed. For *Step 3* first note that in $\mathbb{A}_M$ the operation r plays the role of multiplication $\cdot$ 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 $\mathcal{P}' = (\{v_1, \ldots, v_5\}, \{C'_1 = \langle (v_1, v_2, v_3), R'_1\rangle, C'_2 = \langle (v_2, v_4, v_5), R'_2\rangle\})$, 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 $\mathcal{P}'$ 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. $\diamond$

Using Lemma 11 and Theorems 12,13 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] Eric Allender, Michael Bauland, Neil Immerman, Henning Schnoor, and Heribert Vollmer. The complexity of satisfiability problems: Refining Schaefer's theorem. In *MFCS*, pages 71–82, 2005.

[2] Kirby Baker and Alden Pixley. Polynomial interpolation and the Chinese remainder theorem for algebraic systems. *Mathematische Zeitschrift*, 143(2), pages 165–174, 1975.

[3] Libor Barto. The dichotomy for conservative constraint satisfaction problems revisited. In *LICS*, pages 301–310, 2011.

[4] Libor Barto. The collapse of the bounded width hierarchy. *J. Log. Comput.*, 26(3):923–943, 2016.

[5] Libor Barto and Marcin Kozik. Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem. *Log. Meth. in Comp. Sc.*, 8(1), 2012.

[6] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, 2014.

[7] Libor Barto, Marcin Kozik, and Todd Niven. 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):1782–1802, 2009.

[8] Libor Barto, Marcin Kozik, and Ross Willard. Near unanimity constraints have bounded pathwidth duality. In *LICS*, pages 125–134, 2012.

[9] Libor Barto, Andrei A. Krokhin., and Ross Willard. Polymorphisms, and How to Use Them. In *The Constraint Satisfaction Problem: Complexity and Approximability [Result of a Dagstuhl Seminar]*, pages 1–44, 2017.

[10] Joel Berman, Pawel Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Varieties with few subalgebras of powers. *Trans. Amer. Math. Soc.*, 362(3):1445–1473, 2010.

[11] Jonah Brown-Cohen and Prasad Raghavendra. Correlation Decay and Tractability of CSPs. In *ICALP*, pages 79:1–79:13, 2016.

[12] Andrei A. Bulatov. Three-element Mal'tsev algebras. *Acta Sci. Math (Szeged)*, 71(3-4):469–500, 2002.

[13] Andrei A. Bulatov. A dichotomy theorem for constraints on a three-element set. In *FOCS*, pages 649–658, 2002.

[14] Andrei A. Bulatov. Tractable conservative constraint satisfaction problems. In *LICS*, pages 321–330, 2003.

[15] Andrei A. Bulatov. A graph of a relational structure and constraint satisfaction problems. In *LICS*, pages 448–457, 2004.

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

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

[18] Andrei A. Bulatov. Conservative constraint satisfaction re-revisited. *J. of Comp. and Sys. Sci.*, 82(2):347–356, 2016.

[19] Andrei A. Bulatov. Graphs of finite algebras, edges, and connectivity. CoRR abs/1601.07403 (2016).

[20] Andrei A. Bulatov. Graphs of relational structures: restricted types. In *LICS*, pages 642–651, 2016.

[21] Andrei A. Bulatov. Constraint satisfaction problems over semilattice block Mal'tsev algebras. In *LICS*, 2017.

[22] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. *CoRR*, abs/1703.03021, 2017.

[23] Andrei A. Bulatov and Víctor Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM J. Comput.*, 36(1):16–27, 2006.

[24] Andrei A. Bulatov and Peter Jeavons. Algebraic structures in combinatorial problems. Technische universität Dresden, MATH-AL-4-2001, 2001

[25] Andrei A. Bulatov, Peter Jeavons, and Andrei A. Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.

[26] Andrei A. Bulatov, Andrei A. Krokhin, and Benoit Larose. Dualities for constraint satisfaction problems. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, pages 93–124, 2008.

[27] Andrei A. Bulatov and Matthew Valeriote. Recent results on the algebraic approach to the CSP. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar].*, pages 68–92, 2008.

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

[29] Hubie Chen and Benoit Larose. Asking the metaquestions in constraint tractability. *CoRR*, abs/1604.00932, 2016.

[30] Víctor Dalmau and Andrei A. Krokhin. Majority constraints have bounded pathwidth duality. *Eur. J. Comb.*, 29(4):821–837, 2008.

[31] Rina Dechter. *Constraint processing*. Morgan Kaufmann Publishers, 2003.

[32] Tomas Feder and Moshe Y. Vardi. Monotone monadic SNP and constraint satisfaction. In *STOC*, pages 612–622, 1993.

[33] Tomas Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. of Comput.*, 28:57–104, 1998.

[34] Ralph Freese and Ralph McKenzie. *Commutator theory for congruence modular varieties*. volume 125 of London Math. Soc. Lecture Notes, 1987.

[35] Ralph Freese and Matthew Valeriote. On the Complexity of Some Maltsev Conditions. *IJAC*, 19(1):41–77, 2009.

[36] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. Treewidth and Hypertree Width. In *Tractability: Practical Approaches to Hard Problems*, Cambridge Univ. Press, pages 3–38, 2014.

[37] Pavol Hell and Jaroslav Nešetřil. On the complexity of H-coloring. *J. Comb. Th., Ser.B*, 48:92–110, 1990.

[38] Pavol Hell and Jaroslav Nešetřil. Graphs and homomorphisms, volume 28 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, 2004.

[39] Pawel M. Idziak, Petar Markovic, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM J. Comput.*, 39(7):3023–3037, 2010.

[40] Peter G. Jeavons, David A. Cohen, and Marc Gyssens. Closure properties of constraints. *J. ACM*, 44(4):527–548, 1997.

[41] Peter G. Jeavons. On the algebraic structure of combinatorial problems. *Theor. Comput. Sci.*, 200:185–204, 1998.

[42] Peter G. Jeavons, David A. Cohen, and Martin C. Cooper. Constraints, consistency and closure. *Artif. Intel.*, 101(1-2):251–265, 1998.

[43] Phokion Kolaitis and Moshe Vardi. A game-theoretic approach to constraint satisfaction. In *AAAI*, pages 175-181, 2000.

[44] Phokion Kolaitis. Constraint satisfaction, databases, and logic. In *IJCAI*, pages 1587-1595, 2003.

[45] Marcin Kozik. Weak consistency notions for all the CSPs of bounded width. In *LICS*, pages 633–641, 2016.

[46] Andrei A. Krokhin and Stanislav Zivny (eds.). The Constraint Satisfaction Problem: Complexity and Approximability. *Dagstuhl Follow-Ups*, vol. 7, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[47] Gábor Kun and Mario Szegedy. A new line of attack on the dichotomy conjecture. *Eur. J. Comb.*, 52:338–367, 2016.

[48] Benoit Larose, Cynthia Loten, and Claude Tardif. A characterisation of first-order constraint satisfaction problems. *Log. Meth. in Comput. Sci.*, 3(4), 2007.

[49] Petar Marković. The complexity of CSPs on a 4-element set. Personal communication, 2011.

[50] Miklós Maróti. Malcev on top. Manuscript, available at `http://www.math.u-szeged.hu/~mmaroti/pdf/200x%20Maltsev% 20on%20top.pdf`, 2011.

[51] Miklós Maróti. Tree on top of Malcev. Manuscript, available at `http://www.math.u-szeged.hu/~mmaroti/pdf/200x%20Tree%20on%20top%20of %20Maltsev.pdf`, 2011.

[52] Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra universalis*, 59(3), pages 463-489, 2008.

[53] Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4), pages 17:1–17:24, 2008.

[54] Thomas J. Schaefer. The complexity of satisfiability problems. In *STOC*, pages 216–226, 1978.

[55] Dmitriy Zhuk. On key relations preserved by a weak near-unanimity function. In *ISMVL*, pages 61–66, 2014.

[56] Dmitriy Zhuk. On CSP dichotomy conjecture. In *Arbeitstagung Allgemeine Algebra AAA'92*, page 32, 2016.