

The Collapse of the Bounded Width Hierarchy

Libor Barto*

June 18, 2014

Abstract

We show that every constraint satisfaction problem over a fixed constraint language that has bounded relational width has also relational width $(2, 3)$. Together with known results this gives a trichotomy for width: a constraint satisfaction problem has either relational width 1, or relational width $(2, 3)$ (and no smaller width), or does not have bounded relational width.

Keywords: constraint satisfaction problem, bounded width, relational width, local consistency, Prague instance

1 Introduction

The constraint satisfaction problem (CSP) provides a common framework for expressing a wide range of computational problems. An instance of the CSP consists of variables and constraints, where each constraint is presented as a list of admissible evaluations for some of the variables.

The decision problem for CSP asks whether a given instance admits an evaluation of variables satisfying all the constraints. While this problem is NP-complete, a number of subclasses have been shown to be tractable (that is, in P). One way to define a subclass of the CSP is to restrict the constraint relations that occur in an instance to a fixed set of relations Γ (on a finite set), called a constraint language. Examples of computational problems that can be viewed as CSPs over a fixed language include various Boolean satisfiability problems (eg. k -SAT, HORN-SAT), the problem q -LIN of solving systems of linear equations over the field $\text{GF}(q)$, or graph

*Department of Algebra, Faculty of Mathematics and Physics, Charles University in Prague, Sokolovská 83, 18675 Praha 8, Czech Republic, libor.barto@gmail.com.

homomorphism problems with fixed target graph (eg. k -coloring, graph reachability).

One of the central problems in this area is the dichotomy conjecture [14] postulating that every CSP over a fixed language Γ is either NP-complete, or tractable. The original conjecture was in fact formulated for finite Γ , the more general version would follow from the local-global conjecture from [8] stating that the CSP over Γ is tractable whenever the CSP over every finite $\Gamma' \subseteq \Gamma$ is tractable.

All known tractable cases are solvable either by the few subpowers algorithm [16], by local consistency algorithms [1, 14], or by a combination of these two. This paper sharpens a result of [1] that characterizes the applicability of local consistency algorithms for finite constraint languages.

¹

The idea of local consistency algorithms is to derive as much information as possible about evaluations for small number of variables. More specifically, for fixed $k \leq l$, we find all possible constraints on k variables which can be derived by “considering” l variables at a time. Precise definition depends on what information is considered, we work with so called relational width. The instance is rejected iff a contradiction (an empty constraint) is derived. This algorithm can have false positives. If this is never the case for CSPs over a language Γ , we say that Γ , or the CSP over Γ , has relational width (k, l) . We say that Γ has bounded relational width if it has relational width (k, l) for some k, l .

As an example, consider the language $\Gamma = \{=, \neq\}$, where $=, \neq$ are understood as binary relations on $\{0, 1\}$, and the instance over Γ containing the constraints $x = y, y = z, z = w, x \neq w$. By considering variables $\{x, y, z\}$ we can derive $x = z$. Then, by considering $\{x, z, w\}$, we derive $x = w$, which is a contradiction with the last constraint. In fact, Γ has relational width $(2, 3)$, that is, a contradiction can be found in such a way for any unsolvable instance of the CSP over Γ . A more general problem of relational width $(2, 3)$ is the CSP over the language consisting of all unary and binary relations on $\{0, 1\}$ (this includes 2-SAT). An example of a “thinner” problem is HORN-SAT, which has relational width $(1, 1)$. No examples of “thicker” problems than $(2, 3)$ are known; for a good reason – the main result of this paper shows that all languages of bounded relational width have relational width $(2, 3)$.

Typical problems which do not have bounded relational width are the

¹The paper [4] generalizes [1] in a different direction. The approach in this paper combines [1] and [4].

problems q -LIN [14]. No language which can, in some sense, simulate q -LIN has thus bounded relational width [18]. In [1, 4], it was proved that this is the only obstacle in case of finite languages. The parameters (k, l) depend there on the maximum arity of a relation in Γ . It turned out that when definitions in [1] are properly modified, then the proof can be adjusted to give our stronger result. An interesting consequence is that there is an algorithm for testing whether a given finite Γ has bounded relational width. This algorithm is polynomial for so called core languages or when the size of the domain is fixed.

Essentially the same result was independently obtained by Bulatov [7]. His proof is rather different. He works directly with relational width and the proof is based on detailed local analysis of the instance. The proof presented in this paper uses more global arguments and works with a different (weaker) consistency notion, a Prague instance, which is of independent interest.

2 Notation and terminology

For a positive integer n , we define $[n] = \{1, 2, \dots, n\}$.

An n -ary *relation* on a set D is a subset of D^n , an n -ary *operation* on D is a mapping $D^n \rightarrow D$.

We use D^V to denote the set of all mappings from V to D and $f|_W$ to denote the restriction of $f \in D^V$ to a subset $W \subseteq V$. The projection of a subset C of D^V onto $W \subseteq V$ is denoted by

$$C|_W = \{f|_W : f \in C\} \subseteq D^W .$$

For $W = \{x\}$ we simply write $C|_x$ and sometimes view it as a subset of D (rather than $D^{\{x\}}$).

By an *ordering* of $C \subseteq D^V$ we mean a $|V|$ -ary relation of the form $\{(f(x_1), \dots, f(x_k)) : f \in C\}$ where (x_1, \dots, x_k) is a list of all elements of V .

If t is an n -ary operation on D and $f_1, \dots, f_n \in D^V$, we write $t(f_1, \dots, f_n)$ for the element of D^V defined by

$$(t(f_1, \dots, f_n))(x) = t(f_1(x), \dots, f_n(x)), \quad x \in V ,$$

that is, operations are applied coordinate-wise. Similarly, we can apply an n -ary operation to an n -tuple of members of a k -ary relation.

3 CSP

There are three definitions of the decision problem for CSP widely used in the literature: homomorphism version, satisfiability version and variable-value version. We use a variation of the variable-value form.

In the standard variable-value definition, an instance of the CSP consists of a set of variables V , a domain D and a list (or a set) of constraints, where each constraint (\mathbf{x}, R) comprises a k -tuple of variables $\mathbf{x} = (x_1, \dots, x_k) \in V^k$ and a k -ary relation $R \subseteq D^k$ (so called *constraint relation*). The question is whether there exists a mapping $f : V \rightarrow D$ which satisfies all the constraints, that is, $f(\mathbf{x}) \in R$ for every constraint (\mathbf{x}, R) .

Constraints with repetition of variables can be replaced with constraints without repetitions. For instance, a constraint $((x, y, x), R)$ can be replaced by a pair of constraints $((x, y, z), R), ((x, z), =)$, where z is a new variable. Moreover, order of variables can be changed arbitrarily; for instance, a constraint $((x, y), R)$ is equivalent to $((y, x), R^{-1})$, where R^{-1} is obtained from R by swapping the coordinates. This leads to the following “unordered” version of the definition which is more convenient for our purposes.

Definition 3.1. *An instance of the constraint satisfaction problem (CSP) is a triple $\mathcal{I} = (V, D, \mathcal{C})$ with*

- V a nonempty, finite set of variables,
- D a nonempty, finite domain,
- \mathcal{C} a finite nonempty set of constraints, where each constraint is a subset C of D^W . Here W is a subset of V called the scope of C and the cardinality $|W|$ of W is referred to as the arity of C .

An instance is trivial if it contains the empty constraint.

The question is whether there exists a solution to \mathcal{I} , that is, a function $f : V \rightarrow D$ such that, for each constraint $C \in \mathcal{C}$, say with scope W , the restriction $f|_W$ is in C .

A CSP instance can be viewed as a multipartite colored hypergraph. Partite sets are copies of D , one for each variable, and a constraint $C \in \mathcal{C}$ is a set of hyperedges (between partite sets corresponding to variables in the scope of C) colored by C .

We consider the restriction of this problem to a fixed set of allowed constraint relations.

Definition 3.2. A constraint language Γ is a set of relations on a finite set D containing the binary equality relation. The constraint satisfaction problem over Γ , denoted $\text{CSP}(\Gamma)$, is the subclass of the CSP defined by the property that any constraint has an ordering belonging to Γ .

The definition also slightly deviates from the standard one: in order to be able to simulate repetitions of variables we always demand the equality relation to be in a constraint language. However, this tweak do not significantly change the complexity of the problem [8] nor (relational) width [18].

4 Consistency

In this section, we first introduce relational width and state the main result about the collapse of the relational width hierarchy. Next, we discuss a more technical notion of a (k, l) -system which gives an additional insight into (relational) width and which was used in several papers on the subject (e.g. [1]). Finally, we compare relational width to width.

4.1 Relational width

An elegant way to introduce relational width (k, l) is via the notion of a (k, l) -minimal instance. It is a refinement of the notion of a k -minimal instance used in e.g. [9].

Definition 4.1. Let $l \geq k > 0$ be natural numbers. An instance $\mathcal{I} = (V, D, \mathcal{C})$ of the CSP is (k, l) -minimal, if:

- (M1) Every at most l -element set of variables is contained in the scope of some constraint in \mathcal{C} .
- (M2) For every set W of at most k variables and every pair of constraints C_1 and C_2 from \mathcal{C} whose scopes contain W , the projections of the constraints C_1 and C_2 onto W are the same.

A (k, k) -minimal instance is also called k -minimal.

A (k, l) -minimal instance is (k', l') -minimal for any $l' \geq k' > 0$ such that $k' \leq k$ and $l' \leq l$.

For fixed k, l , every instance \mathcal{I} of the CSP can be converted to an *equivalent* (k, l) -minimal instance (that is, a (k, l) -minimal instance with the same set of solutions) in polynomial time. A straightforward way to do this is as follows. We call this algorithm the (k, l) -minimality algorithm.

1. For each l -element set $W \subset V$ add a “dummy” constraint A^W . (This is only necessary when W is not contained in the scope of any constraint.)
2. Repeat the following process until it stabilizes: For every set W of at most k variables and every pair of constraints C_1 and C_2 from \mathcal{C} whose scopes contain W , remove from C_1 and C_2 all functions f such that $f|_W \notin C_{1|W} \cap C_{2|W}$

It is explained below that the obtained (k, l) -minimal instance \mathcal{J} does not depend on the particular order in which we perform the removals, thus we will call \mathcal{J} *the (k, l) -minimal instance associated to \mathcal{I}* .

This instance \mathcal{J} is equivalent to the original one as removals in the second step clearly do not change the set of solutions. Therefore, if \mathcal{J} is trivial, then \mathcal{I} does not have a solution. However, in the opposite case, we cannot in general conclude that the original instance does have a solution. Those constraint languages for which this converse implication is true are said to have relational width (k, l) .

Definition 4.2. *A constraint language Γ has relational width (k, l) if, for every instance \mathcal{I} of $\text{CSP}(\Gamma)$, \mathcal{I} has a solution provided that the (k, l) -minimal instance associated to \mathcal{I} is nontrivial.*

Γ has relational width k if it has width (k, k) , Γ has bounded relational width if it has relational width (k, l) for some k, l .

In particular, if a constraint language Γ has bounded relational width, then $\text{CSP}(\Gamma)$ is tractable.

It follows from [14] that every language of width $(1, k)$ for some k has width 1. The main result of [13] proves that every language of width 2 has width 1. A consequence of the main result of this paper (see Corollary 6.7) is that every language of bounded width has width $(2, 3)$. Therefore we obtain the following trichotomy for relational width.

Theorem 4.3. *For every constraint language Γ precisely one of the following statements are true.*

1. Γ has relational width 1.
2. Γ has relational width $(2, 3)$ and does not have relational width 2, nor $(1, l)$ for any $l \geq 1$.
3. Γ does not have bounded relational width.

4.2 (k, l) -system

Every (k, l) -minimal instance \mathcal{J} implicitly contains a unique constraint P_W with any given at most k -element scope W . Indeed, for $W \subseteq V$, $|W| \leq k$, there exists a constraint C whose scope contains W (by (M1)) and the projection $P_W = C|_W$ does not depend on the choice of W (by (M2)). Moreover, these induced constraints and any constraint $C \subseteq D^U$ in \mathcal{J} satisfy the following property:

$(k$ -forth property for C) For every $W \subseteq U$ of size at most k and every $f \in P_W$ there exists $g \in C$ such that $g|_W = f$ and $g|_Z \in P_Z$ for every Z , $|Z| \leq k$.

Indeed, as $C|_W = P_W$ (by (M2)) there exists $g \in C$ such that $g|_W = f$ and, by (M2) again, $g|_Z \in P_Z$ for every Z of size at most k .

From (M1) it follows that, for any set $U \subseteq V$ of size l , the instance \mathcal{J} satisfies the k -forth property for some constraint with scope U , therefore it satisfies this property for the constraint D^U .

Definition 4.4. An instance $\mathcal{K} = (V, D, \{P_W : W \subseteq V, |W| \leq k\})$ is called a (k, l) -system if $(P_W)|_Z = P_Z$ for every $Z, W \subseteq V$, $Z \subseteq W$, $|W| \leq k$ and \mathcal{K} satisfies the k -forth property for D^U for every U of size l .

An instance \mathcal{I} is said to be compatible with a (k, l) -system \mathcal{K} if \mathcal{I} and \mathcal{K} have the same set of variables and the same domain, and \mathcal{K} satisfies the k -forth property for every constraint in \mathcal{I} .

In the discussion above we started with a (k, l) -minimal instance \mathcal{J} and associated to it a compatible (k, l) -system \mathcal{K} . Now we start with an arbitrary CSP instance \mathcal{I} and establish a connection between the output of the (k, l) -minimality algorithm and (k, l) -systems compatible with \mathcal{I} .

Let \mathcal{J} be the (k, l) -minimal instance associated to \mathcal{I} and let $\mathcal{K} = (V, D, \{P_W\})$ be the (k, l) -system associated to \mathcal{J} . We claim that \mathcal{K} is equal to the largest (with respect to inclusion of constraints with the same scope) (k, l) -system $\mathcal{K}' = (V, D, \{P'_W\})$ compatible with \mathcal{I} . Such a largest system exists since, as is easily seen, a union² of (k, l) -systems compatible with \mathcal{I} is a (k, l) -system compatible with \mathcal{I} . The system \mathcal{K} itself is compatible with \mathcal{I} , therefore \mathcal{K} is included in \mathcal{K}' . On the other hand, no mapping $g \in C$ proving the compatibility of \mathcal{K}' with \mathcal{I} is removed from C during the execution of the (k, l) -minimality algorithm, thus \mathcal{K}' is contained in \mathcal{K} . We conclude that $\mathcal{K} = \mathcal{K}'$.

²By the union of (k, l) -systems $\mathcal{K}^i = (V, D, \{P_W^i : W \subseteq V, |W| \leq k\})$ we mean the (k, l) -system $(V, D, \{\cup P_W^i : W \subseteq V, |W| \leq k\})$

Observation 4.5. *Let \mathcal{I} be an instance of the CSP and $l \geq k > 0$ natural numbers. The (k, l) -minimality algorithm returns a nontrivial instance if and only if there exists a nontrivial (k, l) -system compatible with \mathcal{I} .*

In particular, a constraint language Γ has relational width (k, l) if and only if every instance of $\text{CSP}(\Gamma)$ which admits a compatible nontrivial (k, l) -system has a solution.

It is also readily seen that a mapping g in a constraint $C \subseteq D^U$ survives the execution of the (k, l) -minimality algorithm if and only if $g|_W \in P_W$ for every $W \subseteq U$ of size at most k . It follows that the result of the (k, l) -minimality algorithm does not depend on a particular order of removals.

Observation 4.5 can also be used to give a more efficient algorithm for CSPs of width (k, l) than the presented naive algorithm: Instead of adding all the dummy constraints of arity l , it is enough to add constraints of arity k and enforce the k -forth property for every constraint and every D^U , $|U| = l$.

4.3 Width

The notion of width is natural in the homomorphism viewpoint on the CSP. We provide a straightforward translation of the standard definition via (k, l) -consistency to the variable-value form. For alternative definitions using Datalog programs, pebble games, or tree width duality we refer to [14, 10].

For our hierarchy collapse result, relational width is better suited since $(2, 3)$ -consistency handles unary and binary constraints in the same way as $(2, 3)$ -minimality, derives unnecessarily more information for ternary constraints, and ignores constraints of arity greater than three.

For the translation of width to the variable-value form, we need the concept of a partial solution. Let $\mathcal{I} = (V, D, \mathcal{C})$ be an instance of the CSP. A mapping $f \in D^U$ (where $U \subseteq V$) is a *partial solution* of \mathcal{I} , if $f|_W \in C$ for every $W \subseteq U$ and every constraint C with scope W .

Now we can describe the (k, l) -consistency algorithm.

1. Create a new instance \mathcal{I}' formed by all partial solutions on every l -element subset of variables, formally

$$\begin{aligned} \mathcal{I}' &= (V, D, \{S_U : U \subseteq V, |U| = l\}), \\ S_U &= \{f \in D^U : f \text{ is a partial solution of } \mathcal{I}\} . \end{aligned}$$

2. Run the (k, l) -minimality algorithm for the instance \mathcal{I}' .

We say that a constraint language Γ has *width* (k, l) if, for every instance \mathcal{I} of $\text{CSP}(\Gamma)$, \mathcal{I} has a solution provided that the (k, l) -consistency algorithm returns a nontrivial instance. Similarly, a constraint language Γ has *bounded width* if it has width (k, l) for some k, l .

An analogue to Observation 4.5 is true with suitable modification of compatibility. We say that a (k, l) -system \mathcal{K} is *compatible** with an instance \mathcal{I} if

for every U of size at most l , $W \subseteq U$ of size at most k , and every $f \in P_W$, there exists a partial solution $g \in D^U$ such that $g|_W = f$ and $g|_Z \in P_Z$ for every Z , $|Z| \leq k$.

It is left as an exercise that the (k, l) -consistency algorithm returns a nontrivial instance if and only if there exists a nontrivial (k, l) -system compatible* with \mathcal{I} . In particular, a constraint language Γ has width (k, l) if and only if every instance of $\text{CSP}(\Gamma)$ which admits a compatible* nontrivial (k, l) -system has a solution.

If a constraint language Γ contains only relations of arity less than or equal to k then there is no difference between relational width (k, l) and width (k, l) since the two notions of compatibility coincide. In particular, a finite constraint language has bounded width if and only if it has bounded relational width.

For relations of arity between $k+1$ and l , the (k, l) -consistency algorithm is in general stronger than the (k, l) -minimality algorithm. More precisely, if the (k, l) -consistency algorithm returns a nontrivial instance then so does the (k, l) -minimality algorithm, but the converse is not true in general.

Example 4.6. *The instance $\mathcal{I} = (V, D, \{C_0, C_1\})$, where*

$$\begin{aligned} V &= \{x, y, z\}, \\ D &= \{0, 1\}, \\ C_i &= \{f \in D^V : (f(x) + f(y) + f(z)) \bmod 2 = i\}, \quad i \in \{0, 1\}, \end{aligned}$$

is $(2, 3)$ -minimal, but the $(2, 3)$ -consistency algorithm returns a trivial instance since there is no partial solution $f \in D^V$.

The result of the (k, l) -consistency algorithm does not depend on constraints of arity greater than l , so this algorithm is not well suited for constraint languages containing “interesting” relations of high arity. A natural modification which partially overcomes this problem is to change the notion of a partial solution $f \in D^U$ in such a way that f is required to satisfy

$f|_W \in C|_W$ for every $W \subseteq U$ and every constraint C whose scope contains W . The next example shows that this modification is not sufficient to guarantee width $(2, 3)$ for all bounded width problems.

Example 4.7. *The instance $\mathcal{I} = (V, D, \{C_0, C_1, \dots, C_4\})$, where*

$$\begin{aligned} V &= \{x_1, x_2, x_3, x_4\}, \\ D &= \{0, 1\}, \\ C_0 &= \{f \in D^V : (f(x_1), f(x_2), f(x_3), f(x_4)) \neq (0, 0, 0, 0)\}, \\ C_i &= \{f \in D^{\{x_i\}} : f(x_i) = 0\} \end{aligned}$$

does not have a solution while the modified $(2, 3)$ -consistency algorithm returns a nontrivial instance. It may be checked using, for instance, Theorem 5.3 and Corollary 6.6 that the constraint language $\Gamma = \{=, \{0\}, \{1\}, \{0, 1\}^4 \setminus \{(0, 0, 0, 0)\}\}$ has relational width $(2, 3)$ (and therefore also width $(2, 4)$).

In terms of width, Theorem 4.3 has the following consequence.

Corollary 4.8. *Let Γ be a finite constraint language of bounded width. Then Γ has width $(2, \max\{3, k\})$, where k is the maximum arity of a relation in Γ .*

Proof. As discussed above, if Γ has bounded width then it has bounded relational width, hence relational width $(2, 3)$ by Theorem 4.3. But the $(2, \max\{3, k\})$ -consistency algorithm is, under the assumption on maximum arity of relations, stronger than $(2, 3)$ -minimality algorithm, and the claim follows. \square

5 Characterization of bounded width

Systems of linear equations over a finite field $\text{GF}(q)$ are resistant to local consistency methods, even when restricted to systems consisting of equations in at most 3 variables. More precisely, the language $\Gamma_{q-3-LIN}$ formed by all affine subspaces of $\text{GF}(q)^3$ does not have bounded (relational) width [14].

The obvious necessary condition for a language Γ to have bounded (relational) width is thus that Γ cannot “simulate” $\Gamma_{q-3-LIN}$.³ Not so obvious is what precisely “simulate” should mean. From today’s perspective, the

³Similar “obvious” necessary conditions were found for $\text{CSP}(\Gamma)$ being tractable (assuming $P \neq NP$), or belonging to the complexity classes L, NL (with suitable assumptions), and for some other properties.

short, slightly imprecise, answer is that “simulates” simply means “positively primitively interprets”.⁴

Positive primitive interpretability is closely related to central objects of interest in universal algebra – varieties and Mal’tsev conditions. This is the basis for the success of the algebraic approach to the CSP. We refer to [8, 11] for an introduction to the algebraic approach and to [6] for the connection to positive primitive interpretability. Here we only define algebraic notions required for the proof of the main result.

An *algebra* \mathbf{A} is a set A , called the *universe* of \mathbf{A} , together with a set of operations on A . An algebra \mathbf{A} is a *clone* if it contains all projection operations (of all arities) and is closed under superposition. A subset B of the universe A of an algebra \mathbf{A} is its *subuniverse*, written $B \leq \mathbf{A}$, if it is closed under all operations in \mathbf{A} . In such a situation, we call B , together with restrictions of all operations of \mathbf{A} to B , a *subalgebra* of \mathbf{A} , we denote this subalgebra \mathbf{B} and write $\mathbf{B} \leq \mathbf{A}$. (We will abuse the notation by using the same letter for an operation in an algebra and in its subalgebra.) More generally, a subset C of \mathbf{A}^W is a *subpower* of \mathbf{A} if C is closed under all operations in \mathbf{A} (applied coordinate-wise, see Section 2). It is easily seen that the projection of a subpower $C \leq \mathbf{A}^W$ onto any subset $U \subseteq W$ is a subpower of \mathbf{D} . In particular, the projection onto any coordinate $x \in W$ is a subuniverse of \mathbf{A} .

Let Γ be a constraint language on a finite domain D . An operation t on D is called a *polymorphism* of Γ if every $R \in \Gamma$ is closed under t . The set D together with all polymorphisms of Γ forms a clone \mathbf{D} , the *clone of polymorphism* of Γ .

A constraint language Γ is a *core* if all of its unary polymorphisms are bijections. To every Γ we can associate an essentially unique core Γ' such that Γ has relational width (k, l) iff Γ' does. Moreover, if Γ is a core, then bounded relational width is preserved by adding all the singleton unary relations $\{d\}$, $d \in D$ [18] (the cited paper works with width, but the proofs can be done in a similar way for relational width).

Lemma 5.1. [18]

1. For any $k \leq l$, a constraint language has relational width (k, l) iff its core does.
2. A core constraint language Γ has bounded width iff Γ together with all singletons does.

⁴More precisely, Γ can simulate $\Gamma_{q-3-LIN}$ if the core of Γ positively primitively interprets a language whose core is $\Gamma_{q-3-LIN}$.

Note that if Γ is a constraint language with all singletons then the clone \mathbf{D} of polymorphisms is *idempotent*, that is, for every operation t of \mathbf{D} we have $t(a, \dots, a) = a$ for all $a \in D$.

A constraint language Γ with all singletons cannot simulate $\Gamma_{q-3-LIN}$ if and only if the clone of polymorphisms \mathbf{D} satisfies a certain algebraic property, namely that the variety generated by \mathbf{D} is congruence meet semi-distributive, or \mathbf{D} is $\text{SD}(\wedge)$ for short. It will require further definitions to explain this properly. We only state a nice characterization of such clones from [19]. (Using Corollary 6.5 this characterization was improved in [17], see Section 8.)

Definition 5.2. *An operation $t : D^n \rightarrow D$, $n \geq 2$, is called a weak near-unanimity operation if $t(a, a, \dots, a) = a$ and*

$$t(b, a, \dots, a) = t(a, b, a, \dots, a) = \dots = t(a, \dots, a, b)$$

for every $a, b \in D$.

Theorem 5.3. [19] *An idempotent clone \mathbf{D} with finite universe D is $\text{SD}(\wedge)$ iff there exists n such that \mathbf{D} has a weak near unanimity operation t of arity m for every $m \geq n$.*

For the proof of the main result, we will only use two properties of $\text{SD}(\wedge)$ clones stated in Theorems 7.2 and 7.3.

As discussed, a necessary condition for bounded width is that the clone of polymorphisms is $\text{SD}(\wedge)$.

Theorem 5.4. [18] *Let Γ be a constraint language with all singletons and \mathbf{D} be its clone of polymorphisms. If Γ has bounded relational width, then \mathbf{D} is $\text{SD}(\wedge)$.*

For finite constraint languages, this necessary condition is also sufficient:

Theorem 5.5. [1, 4] *Let Γ be a finite constraint language with all singletons and \mathbf{D} its clone of polymorphisms. If \mathbf{D} is $\text{SD}(\wedge)$, then Γ has bounded (relational) width.*

Corollary 6.6 generalize this theorem to arbitrary constraint languages and strengthens bounded width to relational width (2,3). The collapse result (Theorem 4.3) then easily follows, see Corollary 6.7.

It will be convenient for us to parametrize the CSP by algebras.

Definition 5.6. *Let \mathbf{D} be an algebra. The CSP over \mathbf{D} , denoted $\text{CSP}(\mathbf{D})$, is the CSP restricted to instances where each constraint is a subpower of \mathbf{D} .*

By the definition of a polymorphism, if Γ is a constraint language and \mathbf{D} its clone of polymorphisms, then every constraint in an instance of $\text{CSP}(\Gamma)$ is a subpower of \mathbf{D} , thus an instance of $\text{CSP}(\Gamma)$ is an instance of $\text{CSP}(\mathbf{D})$.

The following lemma is a folklore.

Lemma 5.7. *Let \mathbf{D} be an algebra and $k \leq l$. Then the (k, l) -minimal instance associated to an instance of $\text{CSP}(\mathbf{D})$ is an instance of $\text{CSP}(\mathbf{D})$.*

Proof. It is enough to observe that removals in step 2 of the (k, l) -minimality algorithms will result in subpowers of \mathbf{D} provided we started with subpowers of \mathbf{D} . To see that, let W be a set of variables, let C_1, C_2 be constraints whose scopes contain W , and assume that C_1, C_2 are subpowers of \mathbf{D} . Let $P = C_1|_W \cap C_2|_W$. Since P is an intersection of projections of subpowers, P is a subpower of \mathbf{D} . After the removals, C_i becomes $C'_i = \{f \in C_i : f|_W \in P\}$ which is, again, a subpower of \mathbf{D} . \square

6 Prague instance

Why does the relational width hierarchy stop precisely at the level $(2, 3)$? An explanation for this phenomenon is that the full strength of $(2, 3)$ -minimality is not needed to guarantee a solution. It is enough when the instance is 1-minimal and satisfies a certain global connectivity property which is implied by $(2, 3)$ -minimality (and not implied by (k, l) -minimality for $k < 2$ or $l < 3$).

Instances satisfying this weaker consistency property are called Prague instances. This concept is used here as a tool for the proof of the main result. However, it may be of independent importance since a result in [3] on robust satisfiability shows a connection (of a similar notion) to the canonical semidefinite programming relaxations of the CSP.

We proceed to define Prague instances. From now on, all instances we shall work with are 1-minimal. For such an instance $\mathcal{I} = (V, D, \mathcal{C})$ and a variable $x \in V$ we write P_x instead of $P_{\{x\}}$ (see Section 4.2 for a definition). In situations when we wish to emphasize the instance, we write $P_x^{(\mathcal{I})}$. Recall that the projection $C|_x$ of any constraint $C \in \mathcal{C}$ whose scope contains x onto x is equal to P_x .

The following definitions can be nicely visualized on the multipartite hypergraph described after Definition 3.1 of the CSP. One can restrict the x -th partite set to P_x (instead of considering the whole D) since the other elements play no role.

Definition 6.1. *Let $\mathcal{I} = (V, D, \mathcal{C})$ be a 1-minimal instance of the CSP. A*

pattern of length $k - 1 > 0$ from x_1 to x_k (in \mathcal{I}) is a tuple

$$p = (x_1, C_1, x_2, C_2, \dots, x_{k-1}, C_{k-1}, x_k) ,$$

where $x_i \in V$ for every $i \in [k]$, $C_i \in \mathcal{C}$, and $\{x_i, x_{i+1}\}$ is in the scope of C_i for every $i \in [k - 1]$. The set of all variables in p is denoted by $\llbracket p \rrbracket$, that is, $\llbracket p \rrbracket = \{x_1, x_2, \dots, x_k\}$. The pattern p is closed if $x_1 = x_k$.

A realization of p in \mathcal{I} is a tuple (f_1, \dots, f_{k-1}) such that, for every $i \in [k - 1]$, $f_i \in C_i$ and $f_i(x_{i+1}) = f_{i+1}(x_i)$. We say that this realization connects $f_1(x_1)$ to $f_{k-1}(x_k)$. We say that p connects $a \in P_x$ to $b \in P_y$ if there exists a realization of p that connects a to b .

For $X \subseteq V$, $x \in X$, and $a, b \in P_x$, we say that a and b are connected in X if there exists a pattern p from x to x which connects a to b such that $\llbracket p \rrbracket \subseteq X$. (The notion depends on the variable x , which should always be clear from the context.)

If p is a pattern from x to y and $A \subseteq P_x$, we define a subset $A + p$ of P_y by

$$A + p = \{b \in P_y : (\exists a \in A) \text{ } p \text{ connects } a \text{ to } b\} .$$

If $p = (x_1, C_1, \dots, x_k)$ and $q = (y_1, C'_1, \dots, y_l)$ are patterns such that $x_k = y_1$ we define $p + q = (x_1, C_1, \dots, x_k = y_1, C'_1, \dots, y_l)$ and $-p = (x_k, \dots, C_1, x_1)$. For a closed pattern p and $m \geq 1$ we put

$$m \times p = \underbrace{p + p + \dots + p}_{m \times} .$$

We write $A - p$ instead of $A + (-p)$. Since $(A + p) + q = A + (p + q)$ (whenever the expressions make sense) we can simply write $A + p + q$. Also observe that $P_x + p = P_y$, $A \subseteq A + p - p$ and $A + (x, C, x) = A$, where p is a pattern from x to y , $A \subseteq P_x$, and C is a constraint whose scope contains x .

We are ready for the main definition.

Definition 6.2. An instance $\mathcal{I} = (V, D, \mathcal{C})$ of the CSP is a Prague instance if it is 1-minimal and for any $x \in V$, any closed pattern p from x to x , and any $a, b \in P_x$, if a and b are connected in $\llbracket p \rrbracket$ then there exists $k > 0$ such that $k \times p$ connects a to b .

Prague instance is a weaker notion than (2, 3)-minimal instance:

Lemma 6.3. Every (2, 3)-minimal instance is a Prague instance.

Proof sketch, for details see Lemma IV.8. in [1]. The lemma is easily proved using the following claim: If $\mathcal{I} = (V, D, \mathcal{C})$ is a $(2, 3)$ -minimal instance, $x, y \in V$, and $a, b \in D$ are such that $(a, b) = (h(x), h(y))$ for some $h \in P_{\{x, y\}}$, then any pattern $q = (x = x_1, C_1, \dots, x_l = y)$ connects a to b .

To prove the claim, let $a_1 = a$, $a_l = b$, $h_1 = h$. By the 2-forth property for $D^{\{x_1, x_2, x_l\}}$ there exists $a_2 \in P_{x_2}$ such that $(a_1, a_2) = (g(x_1), g(x_2))$ for some $g \in P_{\{x_1, x_2\}}$ and $(a_2, a_l) = (h_2(a_2), h_2(a_l))$ for some $h_2 \in P_{\{x_2, x_l\}}$. By the 2-forth property for C_1 , there exists $f_1 \in C_1$ such that $f_1(x_1) = a_1$ and $f_1(x_2) = a_2$. In a similar way, we can find $a_3 \in P_{x_3}$ and $f_2 \in C_2$ such that $f_2(x_2) = a_2$ and $f_2(x_3) = a_3$, and so on. After $l-1$ steps we get a realization (f_1, \dots, f_{l-1}) connecting $a_1 = a$ to $a_l = b$. \square

The next theorem is the core result of the paper. Its proof covers Section 7.

Theorem 6.4. *Let \mathbf{D} be an idempotent $\text{SD}(\wedge)$ clone. Then every nontrivial Prague instance of $\text{CSP}(\mathbf{D})$ has a solution.*

Before moving further, we state some immediate consequences.

Corollary 6.5. *Let \mathbf{D} be an idempotent $\text{SD}(\wedge)$ clone. Then every $(2, 3)$ -minimal instance of $\text{CSP}(\mathbf{D})$ has a solution.*

Proof. The claim follows from Lemma 6.3 and Theorem 6.4. \square

Corollary 6.6. *Let Γ be a constraint language with all singletons and \mathbf{D} be its clone of polymorphisms. Then the following are equivalent.*

1. \mathbf{D} is $\text{SD}(\wedge)$.
2. Γ has relational width $(2, 3)$.
3. Γ has bounded relational width.

Proof. Assume that \mathbf{D} is $\text{SD}(\wedge)$ and let \mathcal{I} be an instance of $\text{CSP}(\Gamma)$. Then \mathcal{I} is an instance of $\text{CSP}(\mathbf{D})$ and, by Lemma 5.7, the $(2, 3)$ -minimal instance \mathcal{J} associated to \mathcal{I} is an instance of $\text{CSP}(\mathbf{D})$. If \mathcal{J} is nontrivial then it has a solution by Corollary 6.5, as required.

The implication “2 \Rightarrow 3” is trivial and “3 \Rightarrow 1” follows from Theorem 5.4. \square

Corollary 6.7. *Let Γ be a constraint language. If Γ has bounded relational width, then Γ has relational width $(2, 3)$.*

Proof. Assume that Γ has bounded relational width. Let Γ' be the core of Γ expanded with all singleton unary relations and let \mathbf{D} be the algebra of polymorphisms of Γ' . By Lemma 5.1, Γ' has bounded relational width and thus, by Theorem 5.4, \mathbf{D} is $\text{SD}(\wedge)$. By Corollary 6.6, Γ' has relational width $(2, 3)$, consequently, Γ has relational width $(2, 3)$. \square

Two useful properties of Prague instances are stated in the following lemmata.

Lemma 6.8. *Let $\mathcal{I} = (V, D, \mathcal{C})$ be a Prague instance, $x \in V$, and p a closed pattern from x to x . Then there exists $m > 0$ such that for every $k \geq m$ and every $a, b \in P_x$, if a and b are connected in $\llbracket p \rrbracket$ then $k \times p$ connects a to b .*

Proof sketch, for details see Lemma IV.10 in [1] part (P α). It is enough to find such an m for a fixed $a = b \in P_x$. From the definition of Prague instance, there exists l such that $l \times p$ connects a to a . Let $c \in P_x$ be such that p connects a to c and $(l-1) \times p$ connects c to a . From the definition of Prague instance again, this time used for a, c and the pattern $l \times p$, we get l' such that $l' \times (l \times p) = (l'l) \times p$ connects a to c . Now $l \times p$ and $(l'l+l-1) \times p$ both connect a to a . Since l and $l'l+l-1$ are coprime, it follows that $k \times p$ connects a to a for all sufficiently large k , as required. \square

Lemma 6.9. *Let $\mathcal{I} = (V, D, \mathcal{C})$ be a Prague instance, $x, y \in V$, p a pattern from x to y , q a pattern from y to x , $A \subseteq P_x$, $B \subseteq P_y$, and $C \in \mathcal{C}$ a constraint whose scope contains $\{x, y\}$. If $A + p = B$ and $B + q = A$, then $A + (x, C, y) = B$.*

Proof sketch, for details see Lemma IV.10 in [1] part (P β). Every element of $A + (x, C, y)$ is connected in $\llbracket p+q \rrbracket$ to an element of B . Using $B + q + p = B$ and the definition of Prague instance we get $A + (x, C, y) \subseteq B$. Similarly $B + (y, C, x) \subseteq A$. Now $B \subseteq B + (y, C, x) - (y, C, x) \subseteq A + (x, C, y)$ and $A + (x, C, y) = B$ follows. \square

Note that in case that $x = y$ the conclusion of the lemma is that $A = B$.

The following properties of Prague instances are proved by induction from Lemma 6.9.

(P2) For every closed pattern p from x to x and every $A \subseteq P_x$, if $A + p = A$, then $A + p - p = A$.

(P3) For every closed patterns p, q from x to x and every $A \subseteq P_x$, if $A + p + q = A$, then $A + p = A$.

Those 1-minimal instances which satisfy (P2) and (P3) and contain only binary constraints, at most one for each pair of variables, were called *weak Prague instances* in [4] and an analogue of Theorem 6.4 was proved in that paper. We do not know whether conditions (P2) and (P3) are sufficient in general.

7 Proof of Theorem 6.4

The strategy for the proof is to gradually make the Prague instance smaller until all the sets P_x are one-element. Then the mapping sending x to the unique element of P_x is clearly a solution.

Let \mathbf{D} be an idempotent $\text{SD}(\wedge)$ clone and let $\mathcal{I} = (P, D, \mathcal{C})$ be a Prague instance of $\text{CSP}(\mathbf{D})$ such that $|P_x| > 1$ for some $x \in V$. Each set P_x is a subuniverse of \mathbf{D} (since it is a projection of a subpower), we denote the corresponding subalgebra of \mathbf{D} by \mathbf{P}_x .

We will find an n -ary operation t of \mathbf{D} , a nonempty subset $X \subseteq V$, and subsets $P_x^i \subseteq P_x$, $x \in V$, $i \in \{0, \dots, n\}$ such that

- (D1) P_x^0 is a proper subset of P_x for every $x \in X$,
 $P_x^i = P_x$ for every $x \in V \setminus X$, $i \in \{0, \dots, n\}$,
- (D2) P_x^i is a subuniverse of \mathbf{P}_x for every $x \in V$, $i \in \{0, \dots, n\}$,
- (D3) $P_x^i + (x, C, y) = P_y^i$ for every $x \in X$, $y \in V$, $i \in \{0, \dots, n\}$ and every constraint $C \in \mathcal{C}$ whose scope contains $\{x, y\}$,
- (D4) $t(a_1, \dots, a_n) \in P_x^0$ for every $x \in V$, $a_1, \dots, a_n \in P_x$ such that $a_i \in P_x^i$ holds for all but at most one $i \in [n]$.

The construction depends on the presence or absence of so called absorbing subuniverses.

Definition 7.1. *A subuniverse A of an idempotent clone \mathbf{P} is absorbing if there exists an operation t of \mathbf{P} of arity $n > 1$ such that $t(a_1, \dots, a_n) \in A$ whenever $a_i \in P$ for all $i \in [n]$ and $a_i \in A$ for all but at most one $i \in [n]$.*

An absorbing subuniverse A of \mathbf{P} is proper if $\emptyset \neq A \neq P$.

The case where some \mathbf{P}_x , $x \in V$ has a proper absorbing subuniverse is handled in Subsection 7.1. The operation t will be any operation witnessing the absorption.

The case where no \mathbf{P}_x has a proper absorbing subuniverse is dealt with in Subsection 7.2. We will get the operation t from the following theorem.

Theorem 7.2. *Let \mathbf{P} be an idempotent $\text{SD}(\wedge)$ clone. Then there exists an operation t of \mathbf{P} and elements $c_1, \dots, c_n, b \in P$ such that $t(a_1, \dots, a_n) = b$ whenever $a_i \in P$ for all $i \in [n]$ and $a_i = c_i$ for all but at most one $i \in [n]$.*

Remarks. This theorem was proved in [4] under an additional assumption that \mathbf{P} is simple. Such a weaker version would be sufficient for our purposes. However, to avoid some technicalities, we use this general version from [5]. \square

The construction is a bit more involved than in the absorption case. A crucial fact is that lack of absorption has strong impact on shape of relations, as witnessed by the following theorem. Here a subset R of $P \times Q$ is called *linked* if the projection to the first (the second, resp.) coordinate is equal to P (Q , resp.) and the transitive closure of the relation

$$\{(a, b) \in P^2 : (\exists c \in Q) (a, c), (b, c) \in R\}$$

is equal to P^2 . In other words, R is connected when viewed as a bipartite graph with partite sets P and Q .

Theorem 7.3. *Let \mathbf{D} be an $\text{SD}(\wedge)$ clone, R be a subuniverse of \mathbf{D}^2 , P, Q be the projections of R to the first and the second coordinate, respectively, and let \mathbf{P}, \mathbf{Q} be the corresponding sublagebras of \mathbf{D} . If neither \mathbf{P} nor \mathbf{Q} has a proper absorbing subuniverse and R is linked then $R = P \times Q$.*

Remarks. This theorem holds under a weaker algebraic assumption on \mathbf{D} – it is enough when \mathbf{D} is Taylor [2]. We remark that, by [19], \mathbf{D} is Taylor if and only if \mathbf{D} has a weak near-unanimity operation (compare to Theorem 5.3). \square

In Subsection 7.3, we show that by removing tuples outside the sets P_x^0 we get a Prague instance \mathcal{J} of $\text{CSP}(\mathbf{D})$ with $P_x^{(\mathcal{J})} = P_x^0$ for all $x \in V$. This will finish the proof since, by (D2), at least one of the sets P_x becomes smaller, so by repeating this procedure we obtain an instance with $|P_x| = 1$ for all $x \in V$.

Some straightforward algebraic facts used throughout are summarized in the following two lemmata.

Lemma 7.4. *Let $x, y \in V$, p a pattern from x to y .*

- (1) *The set $S = \{(a, b) \in P_x \times P_y : p \text{ connects } a \text{ to } b\}$ is a subuniverse of \mathbf{D} and its projection to the first (the second, resp.) coordinate is P_x (P_y , resp.).*

- (2) If s is a k -ary operation of \mathbf{D} , $A_1, \dots, A_k, B \subseteq P_x$ and $s(a_1, \dots, a_k) \in B$ for any $a_i \in A_i, i \in [k]$, then for any $a'_1, \dots, a'_k \in P_y$ such that $a'_i \in A_i + p, i \in [k]$ we have $s(a'_1, \dots, a'_k) \in B + p$.
- (3) If A is a subuniverse (an absorbing subuniverse, resp.) of \mathbf{P}_x , then $A + p$ is a subuniverse (an absorbing subuniverse, resp.) of \mathbf{P}_y .

Proof. The first part of (1) follows from the fact that an operation of \mathbf{D} applied (component-wise) to realizations of p is a realization of p . The second part follows from 1-minimality.

(2) follows from (1).

To prove (3), apply (2) to $A_1 = \dots = A_k = B = A$ (for the absorption part we set $A_i = P_x$ for one i). \square

Lemma 7.5. *If $R, S \leq \mathbf{D}^2$ then the relational composition*

$$R \circ S = \{(a, c) \in D \times D : (\exists b \in D) (a, b) \in R \text{ and } (b, c) \in S\}$$

is also a subuniverse of \mathbf{D}^2 .

Proof. Straightforward. \square

7.1 Absorption

We assume that $z \in V$ is such that \mathbf{P}_z has a proper absorbing subuniverse E . Let t be an operation of \mathbf{P}_z (and \mathbf{D}) witnessing the absorption and n be its arity.

We define a quasiorder (that is, a reflexive and transitive relation) on the set of all pairs (A, x) such that $x \in V$ and $A \subsetneq P_x$ by

$$(A, x) \leq (B, y) \text{ iff } B = A + p \text{ for some pattern } p \text{ from } x \text{ to } y .$$

Let \mathcal{M} be a maximal component of this quasiorder greater or equal to (E, z)

Claim. *For any $x \in V$, if $(A, x), (B, x) \in \mathcal{M}$, then $A = B$.*

Proof. Since $(A, x) \leq (B, x) \leq (A, x)$ there are closed patterns p, q from x to x such that $A + p = B$ and $B + q = A$. Then $A + p + q = A$ and, by property (P2) from the previous section we get $A = A + p = B$. \square

Let X be the set of all variables which appear in a pair from \mathcal{M} . For every $x \in X$, the previous claim allows us to define $P_x^0 = P_x^1 = \dots = P_x^n$ as the unique subset of P_x such that $(P_x^i, x) \in \mathcal{M}$. Thus we have

$$\mathcal{M} = \{(P_x^i, x) : x \in X\} .$$

For $x \in V \setminus X$ we set $P_x^i = P_x$ for all $i \in \{0, \dots, n\}$.

It remains to verify properties (D1) through (D4). (D1) is satisfied by construction. (D2) and (D4) are trivial for $x \in V \setminus X$. For variables $x \in X$ we use part (3) of Lemma 7.4: As each $P_x^i, x \in X, i \in \{0, \dots, n\}$ is obtained from (E, z) by adding a pattern from z to x , P_x^i is an absorbing subuniverse of \mathbf{P}_x , and thus properties (D2) and (D4) are satisfied,

To verify (D3), we take $x \in X, y \in V, i \in \{0, \dots, n\}$, and a constraint $C \in \mathcal{C}$ whose scope contains $\{x, y\}$. If $y \in X$ then, since $(P_x^i, x) \leq (P_y^i, y) \leq (P_x^i, x)$, there exist patterns p from x to y and q from y to x such that $P_x^i + p = P_y^i$ and $P_y^i + q = P_x^i$. Now $P_x^i + (x, C, y) = P_y^i$ by Lemma 6.9. If, on the other hand, $y \in V \setminus X$, then $P_x^i + (x, C, y) = P_y = P_y^i$ since otherwise $(P_x^i + (x, C, y), y)$ would belong to the quosierordered set and then y would belong to X by maximality of \mathcal{M} .

7.2 No absorption

Throughout this subsection we assume that no \mathbf{P}_x has a proper absorbing subuniverse.

We first recall the notion of a congruence. We say that an equivalence $\sim \subseteq P \times P$ on P is a *congruence* of an algebra \mathbf{P} if $\sim \leq \mathbf{P}^2$. A congruence \sim on \mathbf{P} is *maximal* if the unique congruence properly containing \sim is the trivial congruence P^2 . Since the diagonal $\{(a, a) : a \in P\}$ is always a congruence of \mathbf{P} , every at least two-element finite algebra has a maximal congruence.

Let $z \in V$ be any variable such that $|P_z| > 1$ and let \sim be a maximal congruence of \mathbf{P}_z . From Theorem 7.2 we get an n -ary operation t of \mathbf{P}_z and elements $c_1, \dots, c_n, b \in P_z$ such that $t(a_1, \dots, a_n) = b$ whenever $a_i \in P_z$ and $|\{i : a_i \neq c_i\}| \leq 1$. We set $P_z^0 = b/\sim$ and $P_z^i = c_i/\sim$ for all $i \in [n]$.

The set X is defined by

$$X = \{x \in V : \text{there exists a pattern } p_x \text{ from } z \text{ to } x \text{ such that } P_z^0 + p_x \subsetneq P_x\}.$$

We fix patterns p_x from the definition of X (if there are more choices we take any of them). For $x \in X$ we set $P_x^i = P_z^i + p_x$ for all $i \in \{0, \dots, n\}$, and for $x \in V \setminus X$ we set $P_x^i = P_x$.

It remains to verify properties (D1) through (D4). Property (D1) is satisfied by construction.

Properties (D2) and (D4) are trivial for $x \in V \setminus X$. Now we verify them for $x = z$.

Claim. *Properties (D2) and (D4) are satisfied for $x = z$.*

Proof. (D2) follows from idempotency of \mathbf{D} : If s is a k -ary operation of \mathbf{P}_x and $a_1, \dots, a_k \in P_z^i$, then $t(a_1, \dots, a_k) \sim t(a_1, \dots, a_1) = a_1$, so $t(a_1, \dots, a_k) \in a_1 / \sim = P_z^i$.

(D4). If $a_i \in P_z$ for all i and $a_i \in P_z^i$ for all but at most one i , say $i = 1$, then $t(a_1, \dots, a_n) \sim t(a_1, c_2, \dots, c_n) = b$, so $t(a_1, \dots, a_n) \in P_z^0$. \square

To prove (D4) for $x \in X$, we use part (2) of Lemma 7.4 with $p = p_x$, $B = P_z^0$, and $A_i = P_z^i$ for all but one i , for which we choose $A_i = P_z$. Similarly, (D2) follows from part (3) of the same lemma.

To verify (D3) we use the following consequence of Theorem 7.3.

Claim. *For every $x \in V$ and every pattern p from z to x either*

- (1) *for any $i, j \in \{0, \dots, n\}$ with $P_z^i \neq P_z^j$, the sets $P_z^i + p$ and $P_z^j + p$ are disjoint, or*
- (2) *$P_z^i + p = P_x$ for all $i \in \{0, \dots, n\}$.*

Proof. Let $S = \{(a, b) \in P_z \times P_x : p \text{ connects } a \text{ to } b\}$ and

$$R = \alpha \circ S = \{(a, b) \in P_z \times P_x : p \text{ connects some } a' \sim a \text{ to } b\} ,$$

$$R^{-1} = \{(b, a) : (a, b) \in R\} .$$

Since S (see part (1) of Lemma 7.4) and α are subpowers of \mathbf{D} , the relational composition $R = \alpha \circ S$ is, by Lemma 7.5, a subpower of \mathbf{D} . Clearly, the projection of R onto the first coordinate is equal to P_z and the projection to the second coordinate is equal to P_x .

Let β be the transitive closure of $\{(a, b) \in P_z \times P_z : (\exists c \in P_x)(a, c), (b, c) \in R\}$. This equivalence is a congruence of \mathbf{P}_z by Lemma 7.5, because $R \leq (\mathbf{P}_z)^2$ and $\beta = R \circ R^{-1} \circ R \dots$ (finitely many times).

As $\alpha \subseteq \beta$ and α is a maximal congruence, either $\beta = \alpha$, or $\beta = P_x \times P_x$. The first case is a restatement of (1). In the second case we have $R = P_z \times P_x$ by Theorem 7.3, and this is a restatement of (2). \square

For any $x \in X$, case (1) for the pattern p_x takes plays, since for the \sim -block $b / \sim = P_z^0$ we have $P_z^0 + p_x \subsetneq P_x$. Therefore, if $P_z^i \neq P_z^j$, then $P_x^i = P_z^i + p_x$ and $P_x^j = P_z^j + p_x$ are disjoint. It follows that for every $i \in \{0, \dots, n\}$ we have $P_x^i - p_x = P_z^i$.

Now we are ready to prove (P3). Consider $x \in X$, $y \in V$, and a constraint $C \in \mathcal{C}$ whose scope contains $\{x, y\}$. If $y \in V \setminus X$, then $P_z^0 + p_x + (x, C, y) = P_y$, otherwise y would be in X according to the definition. This means that case (2) for the pattern $p = p_x + (x, C, y)$ in the claim takes

plays and thus $P_z^i + p_x + (x, C, y) = P_y$. Since $P_x^i = P_z^i + p_x$, we get $P_x^i + (x, C, y) = P_y = P_y^i$. If, on the other hand, $y \in X$, we have $P_x^i - p_x = P_z^i$ and $P_y^i - p_y = P_z^i$ by the previous paragraph. Then $P_x^i - p_x + p_y = P_y^i$ and $P_y^i - p_y + p_x = P_x^i$, so, by Lemma 6.9, $P_x^i + (x, C, y) = P_y^i$.

7.3 Smaller instance

We are now in the situation that we have an n -ary operation t of \mathbf{D} , a nonempty subset $X \subseteq V$ and subsets $P_x^i \subseteq P_x$, $x \in V$, $i \in \{0, \dots, n\}$ satisfying (D1) through (D4).

We define instances $\mathcal{I}^i = (V, D, \mathcal{C}^i)$ for each $i \in \{0, \dots, n\}$ by restricting \mathcal{I} to the subsets P_x^i . More precisely, we put $\mathcal{C}^i = \{C^i : C \in \mathcal{C}\}$, where for a constraint C with scope W

$$C^i = \{f \in C : (\forall x \in W) f(x) \in P_x^i\} .$$

Every P_x^i is, by (D2), a subuniverse of \mathbf{P}_x and therefore is a subuniverse of \mathbf{D} . It follows that each C^i is a subpower of \mathbf{D} . The instance \mathcal{I}^i is thus an instance of $\text{CSP}(\mathbf{D})$.

Claim. *For every $i \in \{0, \dots, n\}$, the instance \mathcal{I}^i is 1-minimal with $P_x^{(\mathcal{I}^i)} = P_x^i$ for each $x \in V$.*

Proof. We need to show that, for each $C \in \mathcal{C}$ and each x in the scope W of C , the projection A of C^i onto x is equal to P_x^i . The inclusion $A \subseteq P_x^i$ is trivial, so let us take $a \in P_x^i$ and prove that $a \in A$.

Assume first that $x \in X$. Let f be any element of C with $f(x) = a$. By (D3), for every $y \in W$ it holds that $P_x^i + (x, C, y) = P_y^i$, hence $f(y) \in P_y^i$. Therefore $f \in C^i$ and so $a = f(x) \in A$.

Next assume that $x \notin X$ and there exists $y \in W \setminus X$. Then, by (D3) and the second part of (D1), we get $P_y^i + (y, C, x) = P_x$. Therefore, there exists $f \in C$ such that $f(y) \in P_y^i$ and $f(x) = a$. The same argument as in the previous paragraph (using y instead of x) shows that $f \in C^i$ and then $a \in A$.

Finally, if $X \cap W = \emptyset$, then, by the second part of (D1), we have $C = C^i$ and thus $A = P_x = P_x^i$. \square

It remains to show that $\mathcal{J} = \mathcal{I}^0$ is a Prague instance. To distinguish patterns in \mathcal{I} and in the instances \mathcal{I}^i we use the following notation. If $p = (x_1, C_1, x_2, C_2, \dots)$ is a pattern in \mathcal{I} we write p^i for the corresponding pattern in \mathcal{I}^i , that is, the pattern $p^i = (x_1, C_1^i, x_2, C_2^i, \dots)$.

Let $x \in V$, $a, b \in P_x^{(\mathcal{J})} = P_x^0$, let p be a closed pattern from x to x , and assume that a and b are connected in $\llbracket p \rrbracket$ in the instance \mathcal{I} . We aim to prove that $k \times p^0$ connects a to b for some k . (To prove that \mathcal{J} is a Prague instance we could use a stronger assumption that a and b are connected in $\llbracket p \rrbracket$ in the instance \mathcal{I}^0 .)

Let m be an integer provided by Lemma 6.8. In particular, if $a', b' \in P_x^0$ are connected in $\llbracket p \rrbracket$ then $m \times p$ connects a' to b' .

For simplicity, let us assume $x \in V \setminus X$ (the other case is an easy consequence and is handled afterwards). We show that $nm \times p^0$ connects a to b by applying the operation t to an n -tuple of realizations

$$\mathbf{f}^i = (f_{1,1}^i, \dots, f_{1,ml}^i, f_{2,1}^i, \dots, f_{2,ml}^i, \dots, \dots, f_{n,ml}^i), \quad i \in [n]$$

of the pattern $nm \times p$, where l denotes the length of p . We construct these realizations so that

- for every $i \in [n]$, the realization \mathbf{f}^i starts at a and ends at b , that is, $f_{1,1}^i(x) = a$ and $f_{n,ml}^i(x) = b$.
- for every $i, j \in [n]$ with $i \neq j$, the j -th fragment $(f_{j,1}^i, \dots, f_{j,ml}^i)$ of \mathbf{f}^i is a realization of $m \times p^i$.

The i -th realization \mathbf{f}^i is found as follows. Since $a \in P_x = P_x^i$ (recall that $x \in V \setminus X$) there exists a realization $(f_{1,1}^i, \dots, f_{i-1,ml}^i)$ of the pattern $(i-1)m \times p^i$ with $f_{1,1}^i(x) = a$. Similarly, there exists a realization $(f_{i+1,1}^i, \dots, f_{n,ml}^i)$ of the pattern $(n-i)m \times p^i$ with $f_{n,ml}^i(x) = b$. It remains to fill in the i -th segment. Let $a' = f_{i-1,ml}^i(x)$, $b' = f_{i+1,1}^i(x)$ (for $i = 1$ we set $a' = a$, for $i = n$ we set $b' = b$). The elements a' and b' are connected in $\llbracket p \rrbracket$ in the instance \mathcal{I} since the pattern $-(i-1)m \times p$ connects a' to a , the elements a and b are connected in $\llbracket p \rrbracket$, and $(n-i)m \times p$ connects b to b' . Therefore $m \times p$ connects a' to b' . We take any realization $(f_{i,1}^i, \dots, f_{i,ml}^i)$ witnessing it. This finishes the construction of \mathbf{f}^i .

Now we define $f_{i,j} = t(f_{i,j}^1, \dots, f_{i,j}^n)$ for every $i \in [n]$, $j \in [ml]$. For every x in the domain of $f_{i,j}$ and for every $i' \in [n]$, $i' \neq i$ we have $f_{i,j}^{i'}(x) \in P_x^{i'}$, therefore $f_{i,j}(x) \in P_x^0$ by (D4). It follows that $\mathbf{f} = (f_{1,1}, \dots, f_{n,ml})$ is a realization of p^0 . Moreover, since t is idempotent, $f_{1,1}(x) = t(f_{1,1}^1, \dots, f_{1,1}^n) = t(a, \dots, a) = a$ and $f_{n,ml}(x) = b$. Thus \mathbf{f} witness that $nm \times p^0$ connects a and b , as promised.

It remains to deal with the case $x \in X$. If every variable in $\llbracket p \rrbracket$ is in X , then, by (D3), every realization of $m \times p$ witnessing that this pattern connects

a to b is in fact a realization of $m \times p^0$. We can thus assume that some $y \in \llbracket p \rrbracket$ does not belong to X . We can write $p = p_1 + p_2$, where p_2 stars at y . Clearly, we can find $a', b' \in P_y^0$ such that p_1 connects a to a' and p_2 connects b' to b . The previous case used for the pattern $p_2 + p_1$ and elements a', b' (note that these elements are connected in $\llbracket p_2 + p_1 \rrbracket$) gives us that $mn \times (p_2 + p_1)^0$ connects a' to b' . Then the pattern $(p_1 + mn \times (p_2 + p_1) + p_2)^0 = (mn + 1) \times p^0$ connects a to b . The proof of Theorem 6.4 is concluded.

8 Consequences

8.1 The intersection property

An immediate consequence of Corollary 6.5 is a positive answer to Conjecture 1 in [20] about the following intersection property of subuniverses.

Definition 8.1. *For $k > 0$ and $B, C \subseteq D^n$, we say that B and C are k -equal if for every subset I of $[n]$ of size at most k , the projection of B and C onto the coordinates I are equal.*

We say that a clone \mathbf{D} (with a finite universe D) has the k -intersection property if for every $n > 0$ and every $B \leq \mathbf{D}^n$

$$\bigcap \{ C \leq \mathbf{D}^n : C \text{ and } B \text{ are } k\text{-equal} \} \neq \emptyset .$$

Valeriote proved [20] that if an idempotent clone \mathbf{D} is not $\text{SD}(\wedge)$, then \mathbf{D} fails the k -intersection property for every $k > 0$, and conjectured the converse. From our results it follows that the converse is true even for $k = 2$.

Corollary 8.2. *Every idempotent $\text{SD}(\wedge)$ clone \mathbf{D} on a finite set satisfies the 2-intersection property.*

Proof. Let $n > 0$, $B \leq \mathbf{D}^n$ and $R_1, \dots, R_k \leq \mathbf{D}^n$ be a list of all subpowers which are 2-equal to B . We need to show that $\bigcap_{i=1}^k R_i \neq \emptyset$. The claim is trivial for $n \leq 2$, so assume $n \geq 3$. We can look at the relations R_i as constraints of an instance $\mathcal{I} = (D, V, \mathcal{C})$ with $V = [n]$. The instance \mathcal{I} is $(2, 3)$ -minimal since all R_i 's are 2-equal to B , therefore it has a solution by Corollary 6.5. A solution gives us an element of the intersection $R_1 \cap \dots \cap R_k$. \square

8.2 Deciding bounded width

A deeper consequence of Corollary 6.5 is the following characterization of $\text{SD}(\wedge)$ clones.

Theorem 8.3. [17] *An idempotent clone \mathbf{D} with finite universe D is $\text{SD}(\wedge)$ iff \mathbf{D} has weak near-unanimity operations t_1 and t_2 of arity 3 and 4, respectively, such that, for any $a, b \in D$,*

$$t_1(a, a, b) = t_2(a, a, a, b) .$$

For a constraint language Γ with all singletons, the existence of polymorphisms t_1, t_2 satisfying the properties in Theorem 8.3 can be viewed as an instance of $\text{CSP}(\Gamma)$. This idea can be used to obtain an efficient decision algorithm for such polymorphisms. The following result was observed by Marcin Kozik, Miklós Maróti and the author (and possibly others). A different proof is by Bulatov [7].

Corollary 8.4. *There exists a polynomial time algorithm that decides whether the clone of polymorphism \mathbf{D} of a given finite constraint language Γ with all singletons is $\text{SD}(\wedge)$. (If \mathbf{D} is $\text{SD}(\wedge)$, the algorithm can return a certificate – pair of operations (t_1, t_2) of \mathbf{D} as in Theorem 8.3.)*

Proof. We form an instance $\mathcal{I} = (V, D, \mathcal{C})$ of $\text{CSP}(\Gamma)$. The set V of variables will be the disjoint union of D^3 and D^4 . A function $V \rightarrow D$ can be identified with a pair of mappings $(t_1 : D^3 \rightarrow D, t_2 : D^4 \rightarrow D)$.

The first type of constraints will ensure that t_1 of every solution (t_1, t_2) is a polymorphism of Γ : For every relation R in Γ of arity k and all triples of k -tuples $\mathbf{a}_1 = (a_{11}, \dots, a_{1k})$, $\mathbf{a}_2 = (a_{21}, \dots, a_{2k})$, $\mathbf{a}_3 = (a_{31}, \dots, a_{3k})$ in R , we add to \mathcal{C} the constraint

$$C_{R, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3} = \{f \in D^{\{(a_{11}, a_{21}, a_{31}), \dots, (a_{1k}, a_{2k}, a_{3k})\}} : \\ (f(a_{11}, a_{21}, a_{31}), \dots, f(a_{1k}, a_{2k}, a_{3k})) \in R\} .$$

Similarly, we ensure that t_2 in every solution (t_1, t_2) is a polymorphism of Γ . Lastly, we add constraints to ensure that, in every solution (t_1, t_2) , both operations are weak near-unanimity operations and $t_1(a, a, b) = t_2(a, a, a, b)$. For instance, to take care of the last identity we add constraints

$$C_{a,b} = \{f \in D^{\{(a,a,b), (a,a,a,b)\}} : f(a, a, b) = f(a, a, a, b)\}$$

for every $a, b \in D$. (Alternatively, we could have identified some variables to force all the identities but idempotency.)

Clearly, the size of \mathcal{I} is polynomial in the size of the input and (t_1, t_2) is a solution if and only if t_1, t_2 are weak near-unanimity polymorphisms of Γ satisfying $t_1(a, a, b) = t_2(a, a, a, b)$. By Theorem 8.3, \mathcal{I} has a solution if and only if \mathbf{D} is $\text{SD}(\wedge)$.

The algorithm will try to find a solution $f : V \rightarrow D$ of \mathcal{I} by gradually fixing values for variables in some order $V = \{x_1, x_2, \dots\}$. In the k -th step we run the $(2, 3)$ -minimality algorithm $|D|$ -times, for every $d \in D$, on the instance \mathcal{I} with an extra constraint enforcing $f(x_k) = d$. If the $(2, 3)$ -minimality returns a nontrivial instance for some $d \in D$, we add to \mathcal{I} the constraint $f(x_k) = d$ and move to the next step. In the other case, when $(2, 3)$ -minimality returns a trivial instance for every $d \in D$, the algorithm terminates with a negative answer.

Now, if the algorithm succeeds, then the solution $f = (t_1, t_2)$ certifies (by Theorem 8.3) that \mathbf{D} is $\text{SD}(\wedge)$. Moreover, if \mathbf{D} is $\text{SD}(\wedge)$ then $(2, 3)$ -minimality is a correct procedure for deciding whether a solution exists (see Corollary 6.6), therefore the algorithm succeeds. This shows that the algorithm succeeds (and returns a correct certificate) if and only if \mathbf{D} is $\text{SD}(\wedge)$, as required. \square

Corollary 8.5. *There exists a polynomial time algorithm that decides whether a finite core constraint language Γ has bounded (relational) width.*

Proof. First we add all singleton unary relations to Γ and call this constraint language Γ' . The algebra \mathbf{D} is $\text{SD}(\wedge)$ if and only if the algebra of polymorphisms \mathbf{D}' of the constraint language Γ' is $\text{SD}(\wedge)$ since the operations t_1, t_2 from Theorem 8.3 are idempotent. Also, by part (2) of Lemma 5.1, Γ' has bounded width iff Γ has bounded width. The claim now follows from Theorem 8.3 and Corollary 6.6. \square

It is hard to find the core of a constraint language since it is NP-complete to even test whether a graph is a core [15]. Therefore it seems unlikely that testing for bounded width can be done efficiently in general. Is this problem NP-complete?

9 Conclusion and open problems

We have proved that a constraint satisfaction problem over a fixed language (on a finite domain) that can be solved by checking local consistency has relational width $(2, 3)$. Modulo previously known results, this amounts to showing that every $(2, 3)$ -minimal over a language which cannot simulate q -LIN has a solution.

A stronger theorem has been obtained – every Prague instance over such a language has a solution. Is 1-minimality together with properties (P2) and (P3) from Section 6 sufficient to guarantee a solution? Or even just 1-minimality with (P3)?

A different weakening of $(2, 3)$ -minimality is singleton arc consistency (see [12]). An instance is singleton arc consistent if it is 1-minimal and the 1-minimality algorithm succeeds even when we add any constraint of the form “ $x = a$ ”, where x is a variable and $a \in P_x$. The authors in [12] ask whether any singleton arc consistent instance over a bounded width language has a solution. A positive answer would give a more efficient algorithm for solving CSPs over bounded width languages.

Funding

This work was supported by the Grant Agency of the Czech Republic [13-01832S].

Acknowledgement

I would like to thank Marcin Kozik for many discussions on this article.

References

- [1] Libor Barto and Marcin Kozik. Constraint satisfaction problems of bounded width. In *FOCS'09: Proceedings of the 50th Symposium on Foundations of Computer Science*, pages 595–603, 2009.
- [2] Libor Barto and Marcin Kozik. Absorbing subalgebras, cyclic terms, and the constraint satisfaction problem. *Logical Methods in Computer Science*, 8(1), 2012.
- [3] Libor Barto and Marcin Kozik. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44th symposium on Theory of Computing, STOC '12*, pages 931–940, New York, NY, USA, 2012. ACM.
- [4] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, January 2014.
- [5] Libor Barto, Marcin Kozik, and David Stanovský. Mal'tsev conditions, lack of absorption, and solvability. submitted, 2013.
- [6] Manuel Bodirsky. Constraint satisfaction problems with infinite templates. In Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer,

- editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 196–228. Springer, 2008.
- [7] Andrei Bulatov. Bounded relational width. manuscript, 2009.
- [8] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34:720–742, March 2005.
- [9] Andrei A. Bulatov. Combinatorial problems raised from 2-semilattices. *J. Algebra*, 298(2):321–339, 2006.
- [10] Andrei A. Bulatov, Andrei Krokhin, and Benoit Larose. Complexity of constraints. chapter Dualities for Constraint Satisfaction Problems, pages 93–124. Springer-Verlag, Berlin, Heidelberg, 2008.
- [11] Andrei A. Bulatov and Matthew Valeriote. Recent results on the algebraic approach to the CSP. In Nadia Creignou, Phokion G. Kolaitis, and Heribert Vollmer, editors, *Complexity of Constraints*, volume 5250 of *Lecture Notes in Computer Science*, pages 68–92. Springer, 2008.
- [12] Hubie Chen, Victor Dalmau, and Berit Grøtten. Arc consistency and friends. *J. Log. and Comput.*, 23(1):87–108, February 2013.
- [13] Víctor Dalmau. There are no pure relational width 2 constraint satisfaction problems. *Inf. Process. Lett.*, 109(4):213–218, January 2009.
- [14] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic snc and constraint satisfaction: A study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.
- [15] Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discrete Mathematics*, 109(13):117 – 126, 1992.
- [16] Pawel Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. In *Proceedings of the Twenty-Second Annual IEEE Symposium on Logic in Computer Science (LICS 2007)*, pages 213–222. IEEE Computer Society Press, July 2007.
- [17] Marcin Kozik, Andrei Krokhin, Matthew Valeriote, and Ross Willard. Characterizations of several Maltsev conditions. submitted, 2014.

- [18] Benoit Larose and László Zádori. Bounded width problems and algebras. *Algebra Universalis*, 56(3-4):439–466, 2007.
- [19] Miklós Maróti and Ralph McKenzie. Existence theorems for weakly symmetric operations. *Algebra Universalis*, 59(3-4):463–489, 2008.
- [20] Matthew A. Valeriote. A subalgebra intersection property for congruence distributive varieties. *Canad. J. Math.*, 61(2):451–464, 2009.