

Robust Satisfiability of Constraint Satisfaction Problems

Libor Barto*

Department of Mathematics and Statistics
McMaster University
and
Department of Algebra
Charles University in Prague
libor.barto@gmail.com

Marcin Kozik†

Department of Theoretical Computer Science
Jagiellonian University
marcin.kozik@tcs.uj.edu.pl

November 2, 2011

Abstract

An algorithm for a constraint satisfaction problem is called robust if it outputs an assignment satisfying at least $(1 - g(\varepsilon))$ -fraction of the constraints given a $(1 - \varepsilon)$ -satisfiable instance, where $g(\varepsilon) \rightarrow 0$ as $\varepsilon \rightarrow 0$, $g(0) = 0$. Guruswami and Zhou conjectured a characterization of constraint languages for which the corresponding constraint satisfaction problem admits an efficient robust algorithm. This paper confirms their conjecture.

*Supported by the Grant Agency of the Czech Republic, grant 201/09/P223 and by the Ministry of Education of the Czech Republic, grant MSM 0021620839.

†Supported by the Foundation for Polish Science, grant HOM/2008/7 (supported by MF EOG), and Ministry of Science and Higher Education of Poland, grant N206 357036.

1 Introduction

The constraint satisfaction problem (CSP) provides a common framework for many theoretical problems in computer science as well as for many real-life applications. An instance of the CSP consists of a number of variables and constraints imposed on them, and the objective is to efficiently find an assignment for variables with desired properties, or at least to decide whether such an assignment exists. In the decision problem for CSP we want to decide if there is an assignment satisfying all the constraints, in Max-CSP we wish to find an assignment satisfying maximum number of constraints, in the approximation version of Max-CSP we seek for an assignment which is in some sense close to the optimal one. This paper deals with an interesting special case, robust satisfiability of the CSP: Given an instance which is almost satisfiable (say $(1 - \varepsilon)$ -fraction of the constraint can be satisfied), we want to efficiently find an almost satisfying assignment (which satisfies at least $(1 - g(\varepsilon))$ -fraction of the constraints, where $\lim_{\varepsilon \rightarrow 0} g(\varepsilon) = 0$).

Most of the computational problems for the CSP are hard in general, therefore we have to put some restrictions on the instance. In this paper we restrict the constraint language, that is, all constraint relations must come from a fixed, finite set of relations on the domain. Robust satisfiability was in this setting first suggested and studied in a paper by Zwick [27]. The motivation is that in certain practical situations instances might be close to satisfiable (for example, a small fraction of constraints might have been corrupted by noise) and an algorithm that is able to satisfy most of the constraints could be useful.

Zwick [27] concentrated on Boolean CSPs. He gave a semidefinite programming (SDP) based algorithm which finds $(1 - O(\varepsilon^{1/3}))$ -satisfying assignment for $(1 - \varepsilon)$ -satisfiable instances of 2-SAT and linear programming (LP) based algorithm which finds $(1 - O(1/\log(1/\varepsilon)))$ -satisfying assignment for $(1 - \varepsilon)$ -satisfiable instances of Horn- k -Sat (the number k refers to the maximum numbers of variables in a Horn constraint). The quantitative dependence on ε was improved for 2-SAT to $(1 - O(\sqrt{\varepsilon}))$ in [9]. For CUT, a special case of 2-SAT, the Goemans-Williamson algorithm [13] also achieves $(1 - O(\sqrt{\varepsilon}))$. The same dependence was proved more generally for Unique-Games(q) [8] (where q refers to the size of the domain), which improved $(1 - O(\sqrt[5]{\varepsilon} \log^{1/2}(1/\varepsilon)))$ obtained in [20]. For Horn-2-Sat the exponential loss can be replaced by $(1 - 3\varepsilon)$ [19] and even $(1 - 2\varepsilon)$ [14]. These bounds for Horn- k -Sat ($k \geq 3$), Horn-2-Sat, 2-SAT, and Unique-Games(q) are actually essentially optimal [20, 21, 14] assuming Khot's Unique Games Conjecture [20].

On the negative side, if the decision problem for CSP is NP-complete, then given a satisfiable instance it is NP-hard to find an assignment satisfying α -fraction of the constraints for some constant $\alpha < 1$ (see [19] for the Boolean case and [17] for the general case). In particular these problems cannot admit an efficient robust satisfiability algorithm (assuming $P \neq NP$). However NP-completeness of the decision problem is not the only obstacle for robust algorithms. In [15] Håstad proved a strikingly optimal hardness result: for E3-LIN(q) (linear equations over \mathbb{Z}_q where each equation contains precisely 3 variables) it is NP-hard to find an assignment satisfying $(1/q + \varepsilon)$ -fraction of the constraints given an instance which is $(1 - \varepsilon)$ -satisfiable. Note that the trivial random algorithm achieves $1/q$ in expectation.

As observed in [27] the above results cover all Boolean CSPs, because, by Schaefer's theorem [26], E3-LIN(q), Horn- k -Sat, and 2-SAT are essentially the only CSPs with tractable decision problem. What about larger domains? A natural property which distinguishes Horn- k -Sat, 2-SAT, and Unique-Games(q) from E3-LIN(q) and NP-complete CSPs is bounded width [11]. Briefly, a CSP has bounded width if the decision problem can be solved by checking local consistency of the instance. These problems were characterized independently by the authors [1] and Bulatov [4]. It was proved that, in some sense, the only obstacle to bounded width is E3-LIN(q) – the same problem which is difficult for robust satisfiability. These facts motivated Guruswami and Zhou to

conjecture [14] that the class of bounded width CSPs coincide with the class of CSPs admitting a robust satisfiability algorithm.

A partial answer to the conjecture for width 1 problems was recently independently given by Kun, O’Donnell, Tamaki, Yoshida and Zhou [22] (where they also show that width 1 characterizes problems robustly decidable by the canonical LP relaxation), and Dalmau and Krokhin [10] (where they also consider some problems beyond width 1). This paper confirms the Guruswami and Zhou conjecture in full generality. The proof uncovers an interesting connection between the outputs of SDP (and LP) relaxations and Prague strategies – a consistency notion crucial for the bounded width characterization [1].

2 Preliminaries

Definition 2.1. An instance of the CSP is a triple $\mathcal{I} = (V, D, \mathcal{C})$ with V a finite set of variables, D a finite domain, and \mathcal{C} a finite list of constraints, where each constraint is a pair $C = (S, R)$ with S a tuple of variables of length k , called the scope of C , and R an k -ary relation on D (i.e. a subset of D^k), called the constraint relation of C .

A finite set of relations Γ on D is called a constraint language. An instance of $\text{CSP}(\Gamma)$ is an instance of the CSP such that all the constraint relations are from Γ .

An assignment for \mathcal{I} is a mapping $F : V \rightarrow D$. We say that F satisfies a constraint $C = (S, R)$ if $F(S) \in R$ (where F is applied component-wise). The value of F , $\text{Val}(F, \mathcal{I})$, is the fraction of constraints it satisfies. The optimal value of \mathcal{I} is $\text{Opt}(\mathcal{I}) = \max_{F:V \rightarrow D} \text{Val}(F, \mathcal{I})$.

The decision problem for $\text{CSP}(\Gamma)$ asks whether an input instance \mathcal{I} of $\text{CSP}(\Gamma)$ has a solution, i.e. an assignment which satisfies all the constraints. It is known [5] that if $\text{CSP}(\Gamma)$ is tractable, then there exists a polynomial algorithm for finding an assignment F with $\text{Val}(F, \mathcal{I}) = 1$.

Definition 2.2. Let Γ be a constraint language and let $\alpha, \beta \leq 1$ be real numbers. We say that an algorithm (α, β) -approximates $\text{CSP}(\Gamma)$, if it outputs an assignment F with $\text{Val}(F, \mathcal{I}) \geq \alpha$ for every instance \mathcal{I} of $\text{CSP}(\Gamma)$ such that $\text{Opt}(\mathcal{I}) \geq \beta$.

We say that $\text{CSP}(\Gamma)$ admits a robust satisfiability algorithm if there exists a function $g : [0, 1] \rightarrow [0, 1]$ such that $\lim_{\varepsilon \rightarrow 0} g(\varepsilon) = 0$, $g(0) = 0$, and a polynomial algorithm which $(1 - g(\varepsilon), 1 - \varepsilon)$ -approximates $\text{CSP}(\Gamma)$ for every $\varepsilon \in [0, 1]$.

Bounded width and the Guruswami-Zhou conjecture

A natural notion which distinguishes known CSPs which admit a robust satisfiability algorithm (like Horn- k -Sat, 2-SAT, and Unique-Games(q)) from those which do not (like E3-LIN(q), NP-complete CSPs) is *bounded width*.

Informally, $\text{CSP}(\Gamma)$ has bounded width if the decision problem for $\text{CSP}(\Gamma)$ can be solved by checking local consistency. More specifically, for fixed integers (k, l) , the (k, l) -algorithm derives the strongest constraints on k variables which can be deduced by looking at l variables at a time. During the process we may obtain a contradiction (i.e. an empty constraint relation), in this case \mathcal{I} has no solution. We say that $\text{CSP}(\Gamma)$ has *width* (k, l) if this procedure is sound, that is, an instance has a solution if and only if the (k, l) -consistency algorithm does not derive a contradiction. We say that $\text{CSP}(\Gamma)$ has *width* k , if it has width (k, l) for some l . Finally, we say that $\text{CSP}(\Gamma)$ has *bounded width* if it has width k for some k . We refer to [11, 24, 6] for formal definitions and background.

Conjecture 2.3 (Guruswami, Zhou [14]). $\text{CSP}(\Gamma)$ admits a robust satisfiability algorithm if and only if $\text{CSP}(\Gamma)$ has bounded width.

One implication of the Guruswami-Zhou conjecture follows from known results. In [1] and [4] it was proved that $\mathbf{E3-LIN}(q)$ is essentially the only obstacle for bounded width – if Γ cannot “encode linear equations”, then $\text{CSP}(\Gamma)$ has bounded width (here we do not need to assume $\text{P} \neq \text{NP}$). Therefore, if $\text{CSP}(\Gamma)$ does not have bounded width, then Γ can encode linear equations and, consequently, $\text{CSP}(\Gamma)$ admits no robust satisfiability algorithm by Håstad’s result [15] (assuming $\text{P} \neq \text{NP}$). Details will be presented in [10].

This paper proves the other implication:

Theorem 2.4. *If $\text{CSP}(\Gamma)$ has bounded width then it admits a robust satisfiability algorithm. The randomized version of this algorithm returns an assignment satisfying, in expectation, $(1 - O(\log \log(1/\varepsilon)/\log(1/\varepsilon)))$ -fraction of the constraints given a $(1 - \varepsilon)$ -satisfiable instance.*

LP and SDP relaxations

Essentially the only known way to design efficient approximation algorithms is through linear programming (LP) relaxations and semidefinite programming (SDP) relaxations. For instance, the robust satisfiability algorithm for $\text{Horn-}k\text{-Sat}$ [27] uses LP relaxation while the robust satisfiability algorithms for 2-SAT and $\text{Unique-Games}(q)$ [27, 9] are SDP-based.

Recently, robust satisfiability algorithm was devised in [22] and independently [10] for all CSPs of width 1 (this covers $\text{Horn-}k\text{-Sat}$, but not 2-SAT or $\text{Unique-Games}(q)$). The latter one uses a reduction to $\text{Horn-}k\text{-Sat}$ while the former uses an LP relaxation directly. In fact, it is shown in [22] that, in some sense, LP relaxations can be used precisely for width 1 CSPs.

Our algorithm is based on the canonical SDP relaxation [25]. We will use it only for instances with unary and binary constraints (a reduction will be provided in the full version of this article). In this case we can formulate the relaxation as follows.

Definition 2.5. *Let Γ be a constraint language over D consisting of at most binary relations and let $\mathcal{I} = (V, D, \mathcal{C})$ be an instance of $\text{CSP}(\Gamma)$ with m constraints. The goal for the canonical SDP relaxation of \mathcal{I} is to find $(|V||D|)$ -dimensional real vectors $\mathbf{x}_a, x \in V, a \in D$ maximizing*

$$\frac{1}{m} \left(\sum_{(x,R) \in \mathcal{C}} \sum_{a \in R} \|\mathbf{x}_a\|^2 + \sum_{((x,y),R) \in \mathcal{C}} \sum_{(a,b) \in R} \mathbf{x}_a \mathbf{y}_b \right) \quad (*)$$

subject to

$$\begin{aligned} (SDP1) \quad & \mathbf{x}_a \mathbf{y}_b \geq 0 && \text{for all } x, y \in V, a, b \in D, \\ (SDP2) \quad & \mathbf{x}_a \mathbf{x}_b = 0 && \text{for all } x \in V, a, b \in D, a \neq b, \text{ and} \\ (SDP3) \quad & \sum_{a \in D} \mathbf{x}_a = \sum_{a \in D} \mathbf{y}_a, \quad \|\sum_{a \in D} \mathbf{x}_a\|^2 = 1 && \text{for all } x, y \in V. \end{aligned}$$

The dot products $\mathbf{x}_a \mathbf{y}_b$ can be thought of as weights and the goal is to find vectors so that maximum weight is given to pairs (or elements) in constraint relations. It will be convenient to use the notation

$$\mathbf{x}_A = \sum_{a \in A} \mathbf{x}_a$$

for a variable $x \in V$ and a subset $A \subseteq D$, so that condition (SDP3) can be written as $\mathbf{x}_D = \mathbf{y}_D$, $\|\mathbf{x}_D\|^2 = 1$. The contribution of one constraint to (*) is by (SDP3) at most 1 and it is the greater the less weight is given to pairs (or elements) outside the constraint relation.

The optimal value for the sum (*), $\text{SDPOpt}(\mathcal{I})$, is always at least $\text{Opt}(\mathcal{I})$. There are algorithms that outputs vectors with (*) $\geq \text{SDPOpt}(\mathcal{I}) - \delta$ which are polynomial in the input size and $\log(1/\delta)$.

Polymorphisms

Much of the recent progress on the complexity of the decision problem for CSP was achieved by the algebraic approach [5]. The crucial notion linking relations and operations is a polymorphism:

Definition 2.6. *An l -ary operation f on D is a polymorphism of a k -ary relation R , if*

$$(f(a_1^1, \dots, a_1^l), f(a_2^1, \dots, a_2^l), \dots, f(a_k^1, \dots, a_k^l)) \in R$$

whenever $(a_1^1, \dots, a_k^1), (a_1^2, \dots, a_k^2), \dots, (a_1^l, \dots, a_k^l) \in R$.

We say that f is a polymorphism of a constraint language Γ , if it is a polymorphism of every relation in Γ . The set of all polymorphisms of Γ will be denoted by $\text{Pol}(\Gamma)$

We say that Γ is a core, if all its unary polymorphisms are bijections.

The complexity of the decision problem for $\text{CSP}(\Gamma)$ (modulo log-space reductions) depends only on equations satisfied by operations in $\text{Pol}(\Gamma)$ (see [5, 23]). Moreover, equations also determine whether $\text{CSP}(\Gamma)$ has bounded width [24]. The following theorem [12] states one such an equational characterization:

Theorem 2.7. *Let Γ be a core constraint language. Then the following are equivalent.*

- $\text{CSP}(\Gamma)$ has bounded width.
- $\text{Pol}(\Gamma)$ contains a 3-ary operation f_1 and a 4-ary operation f_2 such that, for all $a, b \in D$,

$$f_1(a, a, b) = f_1(a, b, a) = f_1(b, a, a) = f_2(a, a, a, b) = \dots = f_2(b, a, a, a) \quad \text{and} \quad f_1(a, a, a) = a.$$

We remark that the problem of deciding whether $\text{CSP}(\Gamma)$ has bounded width, given Γ as an input, is tractable (the problem is obviously in NP).

3 Prague instances

The proof of the characterization of bounded width CSPs in [1] relies on a certain consistency notion called Prague strategy. It turned out that Prague strategies are related to outputs of canonical SDP relaxations and this connection is what made our main result possible.

The notions defined below will be used only for certain types of instances and constraint languages. Therefore, in the remainder of this section we assume that Λ is a constraint language on a domain D , Λ contains only binary relations, $\mathcal{J} = (V, D, \mathcal{C}^{\mathcal{J}})$ is an instance of $\text{CSP}(\Lambda)$ such that every pair of distinct variables is the scope of at most one constraint $((x, y), P_{x,y}^{\mathcal{J}})$, and if $((x, y), P_{x,y}^{\mathcal{J}}) \in \mathcal{C}^{\mathcal{J}}$ then $((y, x), P_{y,x}^{\mathcal{J}}) \in \mathcal{C}^{\mathcal{J}}$, where $P_{y,x}^{\mathcal{J}} = \{(b, a) : (a, b) \in P_{x,y}^{\mathcal{J}}\}$. (We sometimes omit the superscripts for $P_{x,y}$'s.)

The most basic consistency notion for CSP instances is 1-minimality.

Definition 3.1. *The instance \mathcal{J} is called 1-minimal, if there exist subsets $P_x^{\mathcal{J}} \in D$, $x \in V$ such that, for every constraint $((x, y), P_{x,y}^{\mathcal{J}})$, the constraint relation $P_{x,y}^{\mathcal{J}}$ is subdirect in $P_x^{\mathcal{J}} \times P_y^{\mathcal{J}}$, i.e. the projection of $P_{x,y}^{\mathcal{J}}$ to the first (resp. second) coordinate is equal to $P_x^{\mathcal{J}}$ (resp. $P_y^{\mathcal{J}}$).*

The subset $P_x^{\mathcal{J}}$ is uniquely determined by the instance (if x is in the scope of some constraint).

Weak Prague instance

We will work with a weakening of the notion of a Prague strategy which we call a weak Prague instance. First we need to define steps and patterns.

Definition 3.2. A step (in \mathcal{J}) is a pair of variables (x, y) which is the scope of a constraint in $\mathcal{C}^{\mathcal{J}}$. A pattern from x to y is a sequence of variables $p = (x = x_1, x_2, \dots, x_k = y)$ such that every (x_i, x_{i+1}) , $i = 1, \dots, k-1$ is a step.

For a pattern $p = (x_1, \dots, x_k)$ we define $-p = (x_k, \dots, x_1)$. If $p = (x_1, \dots, x_k)$, $q = (y_1, \dots, y_l)$, $x_k = y_1$ then the concatenation of p and q is the pattern $p + q = (x_1, x_2, \dots, x_k = y_1, y_2, \dots, y_l)$. For a pattern p from x to x and a natural number k , kp denotes the k -time concatenation of p with itself.

For a subset $A \subseteq D$ and a step $p = (x, y)$ we put $A + p$ to be the projection of the constraint relation $P_{x,y}$ onto the second coordinate after restricting the first coordinate to A , that is, $A + p = \{b \in D : (\exists a \in A) (a, b) \in P_{x,y}\}$. For a general pattern p , the set $A + p$ is defined step by step.

Definition 3.3. \mathcal{J} is a weak Prague instance if

- (P1) \mathcal{J} is 1-minimal,
- (P2) for every $A \subseteq P_x^{\mathcal{J}}$ and every pattern p from x to x , if $A + p = A$ then $A - p = A$, and
- (P3) for any patterns p_1, p_2 from x to x and every $A \subseteq P_x^{\mathcal{J}}$, if $A + p_1 + p_2 = A$ then $A + p_1 = A$.

The instance \mathcal{J} is nontrivial, if $P_x^{\mathcal{J}} \neq \emptyset$ for every $x \in V$.

To clarify the definition let us consider the following digraph: vertices are all the pairs (A, x) , where $x \in V$ and $A \subseteq P_x^{\mathcal{J}}$, and $((A, x), (B, y))$ forms an edge iff (x, y) is a step and $A + (x, y) = B$. Condition (P3) means that no strong component contains (A, x) and (A', x) with $A \neq A'$, condition (P2) is equivalent to the fact that every strong component contains only undirected edges. Also note that 1-minimality implies $A \subseteq A + p - p$ for any pattern from x .

The simplest example of an instance satisfying (P1) and (P2) but not (P3) is $V = \{x, y, z\}$, $D = \{0, 1\}$, $P_{x,y} = P_{x,z} = \{(0, 0), (1, 1)\}$, $P_{y,z} = \{(0, 1), (1, 0)\}$. We have $\{0\} + (x, y, z, x) + (x, y, z, x) = \{0\}$, but $\{0\} + (x, y, z, x) = \{1\}$.

The simplest example of an instance satisfying (P1) and (P3) but not (P2) is $V = \{x, y, z\}$, $D = \{0, 1\}$, $P_{x,y} = P_{y,z} = P_{z,x} = \{(0, 0), (1, 0), (1, 1)\}$. Here $\{0\} + (x, y, z, x) = \{0\}$, but $\{0\} - (x, y, z, x) = \{0, 1\}$.

The main result of this paper relies on the following theorem which is a slight generalization of a result in [1].

Theorem 3.4. If $\text{CSP}(\Lambda)$ has bounded width and \mathcal{J} is a nontrivial weak Prague instance of $\text{CSP}(\Lambda)$, then \mathcal{J} has a solution (and a solution can be found in polynomial time).

SDP and Prague instances

We now show that one can naturally associate a weak Prague instance to an output of the canonical SDP relaxation. This material will not be used in what follows, it is included to provide some intuition for the proof of the main theorem.

Let \mathbf{x}_a , $x \in V$, $a \in D$ be arbitrary vectors satisfying (SDP1), (SDP2) and (SDP3). (These vectors do not need to come as a result of the canonical SDP relaxation of a CSP instance.) We define a CSP instance $\mathcal{J} = (V, D, \{((x, y), P_{x,y}) : x, y \in V, x \neq y\})$ by $P_{x,y} = \{(a, b) : \mathbf{x}_a \mathbf{y}_b > 0\}$. and we show that it is a weak Prague instance.

The instance is 1-minimal with $P_x^{\mathcal{J}} = \{a \in D : \mathbf{x}_a \neq \mathbf{0}\}$. To prove this it is enough to verify that the projection of $P_{x,y}$ to the first coordinate is equal to P_x . If $(a, b) \in P_{x,y}$, then clearly \mathbf{x}_a cannot be the zero vector, therefore $a \in P_x^{\mathcal{J}}$. On the other hand, if $a \in P_x^{\mathcal{J}}$ then $0 < \|\mathbf{x}_a\|^2 = \mathbf{x}_a \mathbf{x}_D = \mathbf{x}_a \mathbf{y}_D$ and thus at least one of the dot products $\mathbf{x}_a \mathbf{y}_b$, $b \in D$ is nonzero and $(a, b) \in P_{x,y}$.

To check (P2) and (P3) we note that, for any $x, y \in V, x \neq y$ and $A \subseteq P_x^{\mathcal{J}}$, the vector $\mathbf{y}_{A+(x,y)}$ has either a strictly greater length than \mathbf{x}_A , or $\mathbf{x}_A = \mathbf{y}_{A+(x,y)}$, and the latter happens iff $A + (x, y, x) = A$ (see Claim 4.1.3, in fact, one can check that $\mathbf{y}_{A+(x,y)}$ is obtained by adding to \mathbf{x}_A an orthogonal vector whose size is greater than zero iff $A + (x, y, x) \neq A$). By induction, for any pattern p from x to y , the vector \mathbf{y}_{A+p} is either strictly longer than \mathbf{x}_A , or $\mathbf{x}_A = \mathbf{y}_{A+p}$ and $A + p - p = A$. Now (P2) follows immediately and (P3) is also easily seen: If $A + p + q = A$ then necessarily $\mathbf{x}_A = \mathbf{x}_{A+p}$ which is possible only if $A = A + p$.

Remarks. To prove property (P2) we only need to consider lengths of the vectors. In fact, this property will be satisfied when we start with the canonical linear programming relaxation (and define the instance \mathcal{J} in a similar way). This is not the case for property (P3).

We also remark that the above weak Prague instance is in fact a Prague strategy in the sense of [1]. However the following example shows that \mathcal{J} is not necessarily a (2,3)-strategy, meaning that the (2,3)-algorithm will remove some pairs from constraint relations. Consider $V = \{x, y, z\}$, $D = \{0, 1\}$ and vectors $\mathbf{x}_0 = (1/2, 1/2, 0)$, $\mathbf{x}_1 = (1/2, -1/2, 0)$, $\mathbf{y}_0 = (1/4, -1/4, \sqrt{2}/4)$, $\mathbf{y}_1 = (3/4, 1/4, -\sqrt{2}/4)$, $\mathbf{z}_0 = (1/4, 1/4, \sqrt{2}/4)$, $\mathbf{z}_1 = (3/4, -1/4, \sqrt{2}/4)$. The constraint relations are then $P_{x,y} = \{(0, 1), (1, 0), (1, 1)\}$, $P_{x,z} = \{(0, 0), (0, 1), (1, 1)\}$, $P_{y,z} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. The pair $(0, 0)$ will be deleted from $P_{y,z}$ during the (2,3)-algorithm, since there is no $a \in \{0, 1\}$ such that $(0, a) \in P_{y,x}$ and $(0, a) \in P_{z,x}$.

Finally, we note that if \mathcal{I} is an instance of the CSP with $\text{SDPOpt}(\mathcal{I}) = 1$ and we define \mathcal{J} using vectors with $(*)=1$, then a solution of \mathcal{J} is necessarily a solution to \mathcal{I} . Showing that “ $\text{SDPOpt}(\mathcal{I}) = 1$ ” implies “ \mathcal{I} has a solution” was suggested as a first step to prove the Guruswami-Zhou conjecture. The above example explains that it is not straightforward to achieve this goal using (2,3)-strategies.

4 Proof

The main result, Theorem 2.4, is a consequence of the following theorem. The reduction, derandomization and omitted details will be given in the full version of the paper.

Theorem 4.1. *Let Γ be a core constraint language over D containing at most binary relations. If $\text{CSP}(\Gamma)$ has bounded width, then there exists a randomized algorithm which given an instance \mathcal{I} of $\text{CSP}(\Gamma)$ and an output of the canonical SDP relaxation with value at least $1 - 1/n^{4n}$ (where n is a natural number) produces an assignment with value at least $1 - K/n$, where K is a constant depending on $|D|$. The running time is polynomial in m (the number of constraints) and n^n .*

Proof. Let $\mathcal{I} = (V, D, \mathcal{C})$ be an instance of $\text{CSP}(\Gamma)$ with m constraints and let \mathbf{x}_a , $x \in V$, $a \in D$ be vectors satisfying (SDP1), (SDP2), (SDP3) such that the sum $(*)$ is at least $1 - 1/n^{4n}$. Without loss of generality we assume that $n > |D|$.

Let us first briefly sketch the idea of the algorithm. The aim is to define an instance \mathcal{J} in a similar way as in the previous section (see Step 9), but instead of all pairs with nonzero weight we only include pairs of weight greater than a threshold (chosen in Step 1). This guarantees that every solution to \mathcal{J} satisfies all the constraints of \mathcal{I} which do not have large weight on pairs outside the constraint relation (the bad constraints are removed in Step 3). The instance \mathcal{J} (more precisely, its algebraic closure formed in Step 10) has a solution by Theorem 3.4 as soon as we ensure that

it is a weak Prague instance. Property (P1) is dealt with in a similar way as in [22]: We keep only constraints with a gap – all pairs have either smaller weight than the threshold, or significantly larger (Step 2). This also gives a property similar to the one in the motivating discussion in the previous section: The vector $\mathbf{y}_{A+(x,y)}$ is either significantly longer than \mathbf{x}_A or these vectors are almost the same. However, large amount of small differences can add up, so we need to continue taming the instance. In Steps 4 and 5 we divide the unit sphere into layers and remove some constraints so that almost the same vectors of the form $\mathbf{x}_A, \mathbf{y}_{A+(x,y)}$ never lie in different layers. This already guarantees property (P2). For property (P3) we use “cutting by hyperplanes” idea from [13]. We choose sufficiently many hyperplanes so that every pair $\mathbf{x}_A, \mathbf{x}_B$ of different vectors in the same layer is cut (the bad variables are removed in Step 7) and we do not allow almost the same vectors to cross the hyperplane (Step 8).

The description of the algorithm follows.

1. Choose $r \in \{1, 2, \dots, n-1\}$ uniformly at random.
2. Remove from \mathcal{C} all the unary constraints (x, R) such that $\|\mathbf{x}_a\|^2 \in [n^{-4r-4}, n^{-4r}]$ for some $a \in D$ and all the binary constraints $((x, y), R)$ such that $\mathbf{x}_a \mathbf{y}_b \in [n^{-4r-4}, n^{-4r}]$ for some $a, b \in D$.
3. Remove from \mathcal{C} all the unary constraints (x, R) such that $\|\mathbf{x}_a\|^2 \geq n^{-4r}$ for some $a \notin R$ and all the binary constraints $((x, y), R)$ such that $\mathbf{x}_a \mathbf{y}_b \geq n^{-4r}$ for some $(a, b) \notin R$.

Let $u_1 = 2|D|^2 n^{-4r-4}$ and $u_2 = n^{-4r} - u_1$. For two real numbers $\gamma, \psi \neq 0$ we denote by $\gamma \div \psi$ the greatest integer i such that $\gamma - i\psi > 0$ and this difference is denoted by $\gamma \bmod \psi$.

4. Choose $s \in [0, u_2]$ uniformly at random.
5. Remove from \mathcal{C} all the binary constraints $((x, y), R)$ such that $|\|\mathbf{x}_A\|^2 - \|\mathbf{y}_B\|^2| \leq u_1$ and $(\|\mathbf{x}_A\|^2 - s) \div u_2 \neq (\|\mathbf{y}_B\|^2 - s) \div u_2$ for some $A, B \subseteq D$.

The remaining part of the algorithm uses the following definitions. For all $x \in V$ let $P_x = \{a \in D : \|\mathbf{x}_a\|^2 \geq n^{-4r}\}$. For a vector \mathbf{w} we put $h(\mathbf{w}) = (\|\mathbf{w}\|^2 - s) \div u_2$ and $t(\mathbf{w}) = \left\lceil \pi(\log n)n^{2r} \min\{\sqrt{(h(\mathbf{w})+2)u_2}, 1\} \right\rceil$. We say that \mathbf{w}_1 and \mathbf{w}_2 are *almost the same* if $h(\mathbf{w}_1) = h(\mathbf{w}_2)$ and $\|\mathbf{w}_1 - \mathbf{w}_2\|^2 \leq u_1$.

6. Choose unit vectors $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{\lceil \pi(\log n)n^{2r} \rceil}$ independently and uniformly at random.
7. We say that a variable $x \in V$ is *uncut* if there exists $A, B \subseteq P_x, A \neq B$ such that $h(\mathbf{x}_A) = h(\mathbf{x}_B)$ and $\text{sgn } \mathbf{x}_A \mathbf{q}_i = \text{sgn } \mathbf{x}_B \mathbf{q}_i$ for every $1 \leq i \leq t(\mathbf{x}_A)$ (in words, no hyperplane determined by the first $t(\mathbf{x}_A) = t(\mathbf{x}_B)$ vectors \mathbf{q}_i cuts the vectors $\mathbf{x}_A, \mathbf{x}_B$). Remove from \mathcal{C} all the constraints whose scope contains an uncut variable.
8. Remove from \mathcal{C} all the binary constraints $((x, y), R)$ for which there exist $A \subseteq P_x, B \subseteq P_y$ such that $\mathbf{x}_A, \mathbf{y}_B$ are almost the same and $\text{sgn } \mathbf{x}_A \mathbf{q}_i \neq \text{sgn } \mathbf{y}_B \mathbf{q}_i$ for some $1 \leq i \leq t(\mathbf{x}_A)$.
9. Let \mathcal{S} denote the set of pairs which are the scope of some binary constraint of \mathcal{I} . Let

$$\mathcal{J} = (V, D, \{((x, y), P_{x,y}^{\mathcal{J}}) : (x, y) \in \mathcal{S} \cup \mathcal{S}^{-1}\}), P_{x,y}^{\mathcal{J}} = \{(a, b) : \mathbf{x}_a \mathbf{y}_b \geq n^{-4r}\}.$$

10. Form the algebraic closure \mathcal{J}' of the instance \mathcal{J} : $\mathcal{J}' = (V, D, \{((x, y), P_{x,y}^{\mathcal{J}'}) : (x, y) \in \mathcal{S} \cup \mathcal{S}^{-1}\})$:

$$P_{x,y}^{\mathcal{J}'} = \{(f(a_1, a_2, \dots), f(b_1, b_2, \dots)) : f \in \text{Pol}(\Gamma), (a_1, b_1), (a_2, b_2), \dots \in P_{x,y}^{\mathcal{J}}\}$$

11. Return a solution of \mathcal{J}' .

Claim 4.1.1. *Expected fraction of constraints removed in steps 2, 3, 5, 7 and 8 is at most K/n for some constant K .*

Proof. Step 2. For each binary constraint there are $|D|^2$ choices for $a, b \in D$ and therefore at most $|D|^2$ bad choices for r . For a unary constraint the number of bad choices is at most $|D|$. Thus the probability that a given constraint will be removed is at most $|D|^2/(n-1)$ and it follows that the expected fraction of removed constraints is at most $|D|^2/(n-1)$.

Step 3. The contribution of every removed constraint to the sum $(*)$ is at most $1 - n^{-4r} \leq 1 - n^{-4n+4}$. If more than γ -fraction of the constraints is removed than the sum is at most $1/m((1 - \gamma)m + \gamma m(1 - n^{-4n+4})) = 1 - \gamma n^{-4n+4}$. Since $(*) \geq 1 - 1/n^{4n}$, we have $\gamma \leq 1/n^4$.

Step 5. For every constraint $((x, y), R)$ and every $A, B \subseteq D$ such that $|\|\mathbf{x}_A\|^2 - \|\mathbf{y}_B\|^2| \leq u_1$, $\|\mathbf{x}_A\| \leq \|\mathbf{y}_B\|$, the inequality $(\|\mathbf{x}_A\|^2 - s) \div u_2 \neq (\|\mathbf{y}_B\|^2 - s) \div u_2$ is satisfied only if $s \in (l - iu_2, l + u_1 - iu_2]$ for some integer i , where $l = \|\mathbf{x}_A\|^2 \pmod{u_2}$. These bad choices for s cover at most (u_1/u_2) -fraction of the interval $[0, u_2]$. As $u_1/u_2 < K_1/n^4$ (for a suitable constant K_1 depending on $|D|$), the probability of a bad choice is at most K_1/n^4 . There are $4^{|D|}$ pairs of subsets $A, B \subseteq D$, therefore the probability that the constraint is removed is less than $K_1 4^{|D|}/n^4$ and so is the expected fraction of removed constraints.

Before analyzing Steps 7 and 8 let us observe that, for any vector \mathbf{w} such that $1 \geq \|\mathbf{w}\| \geq n^{-4r}$,

$$\pi(\log n)n^{2r} \|\mathbf{w}\| \leq t(\mathbf{w}) \leq 2\pi(\log n)n^{2r+1} \|\mathbf{w}\| + 1.$$

The first inequality follows from

$$\sqrt{(h(\mathbf{w}) + 2)u_2} = \sqrt{u_2((\|\mathbf{w}\|^2 + 2u_2 - s) \div u_2)} \geq \sqrt{u_2 \frac{\|\mathbf{w}\|^2 + u_2 - s}{u_2}} \geq \|\mathbf{w}\|$$

and the second inequality follows from

$$\sqrt{(h(\mathbf{w}) + 2)u_2} \leq \sqrt{u_2 \frac{(\|\mathbf{w}\|^2 + 2u_2 - s)}{u_2}} \leq \sqrt{\|\mathbf{w}\|^2 + 2u_2} \leq \sqrt{\|\mathbf{w}\|^2 + 2\|\mathbf{w}\|^2} < 2\|\mathbf{w}\|.$$

Step 7. Consider two different subsets A, B of P_x such that $h(\mathbf{x}_A) = h(\mathbf{x}_B)$. Suppose that $A \setminus B \neq \emptyset$, the other case is symmetric. Let θ be the angle between \mathbf{x}_A and \mathbf{x}_B . As $\mathbf{x}_A - \mathbf{x}_{A \cap B} (= \mathbf{x}_{A \setminus B})$, $\mathbf{x}_B - \mathbf{x}_{A \cap B}$ and $\mathbf{x}_{A \cap B}$ are pairwise orthogonal, the angle θ is greater than or equal to the angle θ_A between \mathbf{x}_A and $\mathbf{x}_{A \cap B}$. We have $\sin \theta_A = \|\mathbf{x}_{A \setminus B}\| / \|\mathbf{x}_A\|$. Since $A \subseteq P_x$, we get $\|\mathbf{x}_{A \setminus B}\| \geq \sqrt{n^{-4r}} = n^{-2r}$ and then $\sin \theta_A = \|\mathbf{x}_{A \setminus B}\| / \|\mathbf{x}_A\| \geq n^{-2r} / \|\mathbf{x}_A\|$, so $\theta \geq \theta_A \geq n^{-2r} / \|\mathbf{x}_A\|$.

The probability that \mathbf{q}_i does not cut \mathbf{x}_A and \mathbf{x}_B is thus at most $1 - n^{-2r}/\pi \|\mathbf{x}_A\|$ and the probability that none of the vectors $\mathbf{q}_1, \dots, \mathbf{q}_{t(\mathbf{x}_A)}$ cut them is at most

$$\left(1 - \frac{n^{-2r}}{\pi \|\mathbf{x}_A\|}\right)^{t(\mathbf{x}_A)} \leq \left[\left(1 - \frac{1}{\pi n^{2r} \|\mathbf{x}_A\|}\right)^{\pi n^{2r} \|\mathbf{x}_A\|}\right]^{\log n} \leq \left(\frac{1}{2}\right)^{\log n} = \frac{1}{n}.$$

The first inequality uses the fact that $t(\mathbf{x}_A) \geq (\log n)n^{2r} \|\mathbf{x}_A\|$ observed above. In the second inequality we have used that $(1 - 1/\eta)^\eta \leq 1/2$ whenever $\eta \geq 2$.

For a single variable there are at most $4^{|D|}$ choices for $A, B \subseteq P_x$, therefore the probability that x is uncut is at most $4^{|D|}/n$. The scope of every constraint contains at most 2 variables, hence the probability that a constraint is removed is at most $2 \cdot 4^{|D|}/n$ and the expected fraction of the constraints removed in this step has the same upper bound.

Step 8. Assume that $((x, y), R)$ is a binary constraint and $A \subseteq P_x, B \subseteq P_y$ are such that \mathbf{x}_A and \mathbf{y}_B are almost the same. Let θ be the angle between \mathbf{x}_A and \mathbf{y}_B and θ_A be the angle between \mathbf{y}_B and $\mathbf{y}_B - \mathbf{x}_A$. By the law of sines we have $\|\mathbf{x}_A\|/(\sin \theta_A) = \|\mathbf{y}_B - \mathbf{x}_A\|/(\sin \theta)$, and

$$\theta \leq 2 \sin \theta = \frac{\|\mathbf{y}_B - \mathbf{x}_A\|}{\|\mathbf{x}_A\|} \sin(\theta_A) \leq \frac{\|\mathbf{y}_B - \mathbf{x}_A\|}{\|\mathbf{x}_A\|} \leq \frac{\sqrt{u_1}}{\|\mathbf{x}_A\|}.$$

Therefore, the probability that vectors \mathbf{x}_A and \mathbf{y}_B are cut by some of the vectors \mathbf{q}_i , $1 \leq i \leq t(\mathbf{x}_A)$ is at most

$$t(\mathbf{x}_A) \frac{\sqrt{u_1}}{\|\mathbf{x}_A\|} \leq (2\pi(\log n)n^{2r} \|\mathbf{x}_A\| + 1) \frac{\sqrt{2|D|^2 n^{-4r-4}}}{\|\mathbf{x}_A\|} \leq K_2(\log n)n^{-2} \leq \frac{K_2}{n},$$

where K_2 is a constant. There are at most $4^{|D|}$ choices for A, B , so the probability that our constraint will be removed is less than $K_2 4^{|D|}/n$. \square

We now proceed to show that \mathcal{J} is a weak Prague instance. First we check condition (P1):

Claim 4.1.2. *The instance \mathcal{J} is 1-minimal and $P_x^{\mathcal{J}} = P_x$.*

Proof. Let $(x, y) \in \mathcal{S}$ and take an arbitrary constraint $((x, y), R)$ which remained in \mathcal{C} .

First we prove that $P_{x,y} \subseteq P_x \times P_y$ for every $a, b \in D$. Indeed, if $(a, b) \in P_{x,y}$ then $\mathbf{x}_a \mathbf{y}_b \geq n^{-4r}$, therefore $\|\mathbf{x}_a\|^2 = \mathbf{x}_a \mathbf{x}_D = \mathbf{x}_a \mathbf{y}_D \geq n^{-4r}$, so $a \in P_x$. Similarly, $b \in P_y$.

On the other hand, if $a \in P_x$ then $n^{-4r} \leq \|\mathbf{x}_a\|^2 = \mathbf{x}_a \mathbf{y}_D$, thus there exist $b \in D$ such that $\mathbf{x}_a \mathbf{y}_b \geq n^{-4r}/|D| \geq n^{-4r-4}$ (we have used $n^4 \geq |D|$). But then $\mathbf{x}_a \mathbf{y}_b \geq n^{-4r}$, otherwise the constraint $((x, y), R)$ would be removed in Step 2. This implies that $(a, b) \in P_{x,y}$. We have shown that the projection of $P_{x,y}$ to the first coordinate contains P_x . Similarly, the second projection contains P_y , so $P_{x,y}$ is subdirect in $P_x \times P_y$. \square

For verification of properties (P2) and (P3) the following observation will be useful.

Claim 4.1.3. *Let $(x, y) \in \mathcal{S} \cup \mathcal{S}^{-1}$, $A \subseteq P_x$, $B = A + (x, y)$. If $A = B + (y, x)$, then the vectors \mathbf{x}_A and \mathbf{y}_B are almost the same. In the other case, i.e. if $A \subsetneq B + (y, x)$, then $h(\mathbf{y}_B) > h(\mathbf{x}_A)$.*

Proof. The number $\|\mathbf{y}_B - \mathbf{x}_A\|^2$ is equal to

$$\mathbf{y}_B \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_B + \mathbf{x}_A \mathbf{x}_A = \mathbf{x}_D \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_B + \mathbf{x}_A \mathbf{y}_D = \mathbf{x}_{D \setminus A} \mathbf{y}_B + \mathbf{x}_A \mathbf{y}_{D \setminus B}.$$

No pair (a, b) , with $a \in A$ and $b \in D \setminus B$, is in $P_{x,y}^{\mathcal{J}}$ so the dot product $\mathbf{x}_a \mathbf{y}_b$ is smaller than n^{-4r} . Then in fact $\mathbf{x}_a \mathbf{y}_b < n^{-4r-4}$ otherwise all the constraints with scope (x, y) would be removed in Step 2. It follows that the second summand is always at most $|D|^2 n^{-4r-4}$ and the first summand has the same upper bound in the case $B + (y, x) = A$.

Moreover, $\|\mathbf{y}_B\|^2 - \|\mathbf{x}_A\|^2$ is equal to

$$\mathbf{y}_B \mathbf{y}_B - \mathbf{x}_A \mathbf{x}_A = \mathbf{x}_D \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_D = \mathbf{x}_D \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_{D \setminus B} = \mathbf{x}_{D \setminus A} \mathbf{y}_B - \mathbf{x}_A \mathbf{y}_{D \setminus B}.$$

If $B + (y, x) = A$ then we have a difference of two nonnegative numbers less than or equal $|D|^2 n^{-4r-4}$, therefore the absolute value of this expression is at most u_1 . But then $h(\mathbf{x}_A) = h(\mathbf{y}_B)$, otherwise all constraint with scope (x, y) or (y, x) would be removed in Step 5. Using the previous paragraph, it follows that \mathbf{x}_A and \mathbf{x}_B are almost the same.

If $B + (y, x)$ properly contains A then the first summand is greater than or equal to n^{-4r} , so the whole expression is at least $n^{-4r} - |D|^2 n^{-4r-4} > u_2$ and thus $h(\mathbf{y}_B) > h(\mathbf{x}_A)$. \square

Claim 4.1.4. \mathcal{J} is a weak Prague instance.

Proof. (P2). Let $A \subseteq P_x$ and let $p = (x_1, \dots, x_i)$ be a pattern in \mathcal{J} from x to x (i.e. $x_1 = x_i = x$). By the previous claim $h(\mathbf{x}_A) = h((\mathbf{x}_i)_{A+(x_1, \dots, x_i)}) \geq h((\mathbf{x}_{i-1})_{A+(x_1, \dots, x_{i-1})}) \geq \dots \geq h((\mathbf{x}_2)_{A+(x_1, x_2)}) \geq h(\mathbf{x}_A)$. It follows that all these inequalities must in fact be equalities and, by applying the claim again, the vectors $(\mathbf{x}_j)_{A+(x_1, x_2, \dots, x_j)}$ and $(\mathbf{x}_{j+1})_{A+(x_1, x_2, \dots, x_{j+1})}$ are almost the same and $A + (x_1, x_2, \dots, x_{j+1}) + (x_{j+1}, x_j) = A + (x_1, x_2, \dots, x_j)$ for every $1 \leq j < i$. Therefore $A + p - p = A$ as required.

(P3). Let $A \subseteq P_x$, let $p_1 = (x_1, \dots, x_i), p_2$ be two patterns from x to x such that $A + p_1 + p_2 = A$ and let $B = A + p_1$. For contradiction assume $A \neq B$. The same argument as above proves that the vectors $(\mathbf{x}_j)_{A+(x_1, x_2, \dots, x_j)}$ and $(\mathbf{x}_{j+1})_{A+(x_1, x_2, \dots, x_{j+1})}$ are almost the same for every $1 \leq j < i$, and then $h(\mathbf{x}_A) = h(\mathbf{x}_B)$. There exists $i \leq t(\mathbf{x}_A)$ such that $\text{sgn } \mathbf{x}_A \mathbf{q}_i \neq \text{sgn } \mathbf{x}_B \mathbf{q}_i$, otherwise x is uncut and all constraints whose scope contains x would be removed in Step 7. But this leads to contradiction, since $\text{sgn } (\mathbf{x}_j)_{A+(x_1, \dots, x_j)} \mathbf{q}_i = \text{sgn } (\mathbf{x}_{j+1})_{A+(x_1, \dots, x_{j+1})} \mathbf{q}_i$ for all $1 \leq j < i$, otherwise the constraints with scope (x_j, x_{j+1}) would be removed in Step 8. \square

Observe that every solution F to \mathcal{J} is a solution to the torso of \mathcal{I} : For every remaining unary constraint (x, R) we have $P_x \subseteq R$ (from Step 3) and for every remaining binary constraint $((x, y), R)$ we have $P_{x,y} \subseteq R$. Since we have removed at most (K/n) -fraction of the constraints from \mathcal{C} , the mapping F is an assignment for the original instance \mathcal{I} of value at least K/n . Also, the instance \mathcal{J} is nontrivial because, for each $x \in V$, there exists at least one $a \in D$ with $\|\mathbf{x}_a\|^2 > 1/n^4$ (recall that we assume $n > |D|$).

The only problem is that the CSP over the constraint language of \mathcal{J} (consisting of $P_{x,y}^{\mathcal{J}}$'s) does not necessarily have bounded width. This is why we are forming the algebraic closure \mathcal{J}' in Step 10. The new instance still has the property that $P_x^{\mathcal{J}'} = \{f(a_1, a_2, \dots) : f \in \text{Pol}(\Gamma), a_1, a_2, \dots \in P_x\} \subseteq R$ for every unary constraint (x, R) , and $P_{x,y}^{\mathcal{J}'} \subseteq R$ for every binary constraint $((x, y), R)$, since the constraint relations are preserved by every polymorphism of Γ . Moreover, every polymorphism of Γ is a polymorphism of the constraint language Λ' of \mathcal{J}' , therefore $\text{CSP}(\Lambda')$ has bounded width (see Theorem 2.7 for instance; technically, Λ' does not need to be a core, but we can simply add all the singleton unary relations). As the algebraic closure of a weak Prague instance is a weak Prague instance, we can apply Theorem 3.4 to get a solution to \mathcal{J}' . \square

5 Open Problems

The quantitative dependence of g on ε is not very far from the (UGC-) optimal bound for **Horn- k -Sat**. Is it possible to get rid of the extra $\log \log(1/\varepsilon)$?

A straightforward derandomization using a result from [18] has $g(\varepsilon) = O(\log \log(1/\varepsilon) / \sqrt{\log(1/\varepsilon)})$. How to improve it to match the randomized version?

It was observed by Andrei Krokhin that the quantitative dependence is, at least to a large extent, also controlled by the polymorphisms of the constraint language. The problems **2-SAT**, **Unique-Games**(q) suggest that majority or, more generally, near-unanimity polymorphisms could be responsible for polynomial behavior.

The simplest example of polymorphism which does not imply any known stronger property for decision CSPs other than bounded width is the 2-semilattice operation f on a three element domain $D = \{0, 1, 2\}$ defined by $f(0, 0) = f(0, 1) = f(1, 0) = 0$, $f(1, 1) = f(1, 2) = f(2, 1) = 1$, $f(2, 2) = f(2, 0), f(0, 2) = 2$. This might be a source for possible hardness results.

Finally, we believe that the connection between SDP, LP and consistency notions deserves further investigation.

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. New conditions for taylor varieties and CSP. In *Proceedings of the 2010 25th Annual IEEE Symposium on Logic in Computer Science, LICS '10*, pages 100–109, Washington, DC, USA, 2010. IEEE Computer Society.
- [3] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. 2011. in preparation.
- [4] Andrei Bulatov. Bounded relational width. 2009. manuscript.
- [5] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the complexity of constraints using finite algebras. *SIAM J. Comput.*, 34:720–742, March 2005.
- [6] 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.
- [7] Stanley N. Burris and H. P. Sankappanavar. *A course in universal algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1981.
- [8] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, STOC '06*, pages 205–214, New York, NY, USA, 2006. ACM.
- [9] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. *ACM Trans. Algorithms*, 5:32:1–32:14, July 2009.
- [10] Victor Dalmau and Andrei Krokhin. Robust satisfiability for CSPs: algorithmic and hardness results. 2011. in preparation.
- [11] Tomás Feder and Moshe Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28:57–104, February 1999.
- [12] Ralph Freese, Marcin Kozik, Andrei Krokhin, Miklós Maróti, Ralph McKenzie, and Ross Willard. On Maltsev conditions associated with omitting certain types of local structures. 2011. in preparation.
- [13] M. X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.
- [14] Venkatesan Guruswami and Yuan Zhou. Tight bounds on the approximability of almost-satisfiable Horn SAT and exact hitting set. In Dana Randall, editor, *SODA*, pages 1574–1589. SIAM, 2011.
- [15] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48:798–859, July 2001.

- [16] David Hobby and Ralph McKenzie. *The structure of finite algebras*, volume 76 of *Contemporary Mathematics*. American Mathematical Society, Providence, RI, 1988.
- [17] Peter Jonsson, Andrei Krokhin, and Fredrik Kuivinen. Hard constraint satisfaction problems have hard gaps at location 1. *Theor. Comput. Sci.*, 410:3856–3874, September 2009.
- [18] Zohar Shay Karnin, Yuval Rabani, and Amir Shpilka. Explicit dimension reduction and its applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:121, 2009.
- [19] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000.
- [20] Subhash Khot. On the power of unique 2-prover 1-round games. In *In Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 767–775. ACM Press, 2002.
- [21] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37:319–357, April 2007.
- [22] Gabor Kun, Ryan O’Donnell, Suguru Tamaki, Yuichi Yoshida, and Yuan Zhou. Linear programming, width-1 CSPs and robust satisfaction. 2011. manuscript.
- [23] Benoît Larose and Pascal Tesson. Universal algebra and hardness results for constraint satisfaction problems. *Theor. Comput. Sci.*, 410:1629–1647, April 2009.
- [24] Benoit Larose and László Zádori. Bounded width problems and algebras. *Algebra Universalis*, 56(3-4):439–466, 2007.
- [25] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC’08*, pages 245–254, 2008.
- [26] Thomas J. Schaefer. The complexity of satisfiability problems. In *Conference Record of the Tenth Annual ACM Symposium on Theory of Computing (San Diego, Calif., 1978)*, pages 216–226. ACM, New York, 1978.
- [27] Uri Zwick. Finding almost-satisfying assignments. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC ’98, pages 551–560, New York, NY, USA, 1998. ACM.

Appendix

Omitted details concerning Theorem 2.4

Reduction to core constraint languages with unary and binary relations

The reduction is given in the following proposition.

Proposition 5.1. *Let Γ be a constraint language on the domain D which contains relations of maximum arity l and such that $\text{CSP}(\Gamma)$ has bounded width. Then there exists a core constraint language Γ' on D' containing only at most binary relations such that $\text{CSP}(\Gamma')$ has bounded width and such that the following holds: If $\text{CSP}(\Gamma')$ admits a robust satisfiability algorithm which is $(1 - g(\varepsilon), 1 - \varepsilon)$ -approximating (for every ε), then $\text{CSP}(\Gamma)$ admits a robust satisfiability algorithm which is $(1 - (l + 1)g(\varepsilon), 1 - \varepsilon)$ -approximating.*

Proof. First we form the core of Γ : We take a unary polymorphism $f \in \text{Pol}(\Gamma)$ with minimal image (with respect to inclusion) and put $\Gamma^c = \{R^c = R \cap f(D)^{\text{arity}(R)} : R \in \Gamma\}$, $D^c = f(D)$. Then Γ^c is a core constraint language. It is known that $\text{CSP}(\Gamma)$ has bounded width iff $\text{CSP}(\Gamma^c)$ does (see [24]), therefore $\text{CSP}(\Gamma^c)$ has bounded width.

Next we define the constraint language Γ' . The domain is $D' = (D^c)^l$. For every relation $R^c \in \Gamma^c$ of arity k we add to Γ' the unary relation R' defined by

$$(a_1, \dots, a_l) \in R' \quad \text{iff} \quad (a_1, \dots, a_k) \in R^c,$$

for every $k \leq l$ we add the binary relation

$$E_k = \{((a_1, \dots, a_l), (b_1, \dots, b_l)) : a_1 = b_k\},$$

and for every $(a_1, \dots, a_l) \in D'$ we add the singleton unary relation $\{(a_1, \dots, a_l)\}$. The singletons ensure that Γ' is a core. That $\text{CSP}(\Gamma')$ has bounded width can be seen, for instance, from Theorem 2.7: If f_1^c, f_2^c are polymorphisms of Γ^c from this theorem, then the corresponding operations f_1', f_2' acting coordinate-wise on D' satisfy the same equations and it is straightforward to check that f_1', f_2' are polymorphisms of Γ' .

Now, let $\mathcal{I} = (V, D, \mathcal{C})$ be an instance of $\text{CSP}(\Gamma)$ with $\text{Opt}(\mathcal{I}) = \varepsilon$. We transform \mathcal{I} to an instance \mathcal{I}' of $\text{CSP}(\Gamma')$ as follows. We keep the original variables and for every constraint $C = ((x_1, \dots, x_k), R)$ in \mathcal{C} we introduce a new variable x_C and add $k + 1$ constraints

$$((x_C), R'), ((x_1, x_C), E_1), ((x_2, x_C), E_2), \dots, ((x_k, x_C), E_k). \quad (\dagger)$$

If $F : V \rightarrow D$ is an assignment for \mathcal{I} of value $1 - \varepsilon$ then $F^c = fF$ has at least the same value (as f preserves the constraint relations), and the assignment F' for \mathcal{I}' defined by

$$\begin{aligned} F'(x) &= (F^c(x), ?, \dots, ?) \quad \text{for } x \in V \\ F'(x_C) &= (F^c(x_1), \dots, F^c(x_k), ?, \dots, ?) \quad \text{for } C = ((x_1, \dots, x_k), R) \end{aligned}$$

(where $?$ stands for an arbitrary element of A) has value at least $1 - \varepsilon$, since all the binary constraints in \mathcal{I}' are satisfied, and the constraint (x_C, R') is satisfied whenever F satisfies C .

We run the robust algorithm for $\text{CSP}(\Gamma')$ to get an assignment G' for \mathcal{I}' with value at least $1 - g(\varepsilon)$, and we define $G(x)$, $x \in V$ to be the first coordinate of $G'(x)$. Note that, for any constraint C of \mathcal{I} , if G' satisfies all the constraints (\dagger) then G satisfies C . Therefore the value of G is at least $1 - (l + 1)g(\varepsilon)$. \square

Proof of Theorem 2.4 using Theorem 4.1

Let Γ be a core constraint language with at most binary relations (which we can assume by Proposition 5.1) such that $\text{CSP}(\Gamma)$ has bounded width. Let \mathcal{I} be an instance of $\text{CSP}(\Gamma)$ with m constraints and let $1 - \varepsilon = \text{Opt}(\mathcal{I})$.

We first check whether \mathcal{I} has a solution. This can be done in polynomial time since $\text{CSP}(\Gamma)$ has bounded width. If a solution exists we can find it in polynomial time (see the note after Definition 2.1).

In the other case we know that $\varepsilon \geq 1/m$. We run the SDP relaxation with precision $\delta = 1/m$ and obtain vectors with the sum (*) equal to $v \geq \text{SDPOpt}(\mathcal{I}) - 1/m$. Finally, we execute the algorithm provided in Theorem 4.1 with the following choice of n .

$$n = \left\lfloor \frac{\log \omega}{4 \log \log \omega} \right\rfloor, \quad \text{where } \omega = \min \left\{ \frac{1}{1-v}, m \right\}.$$

The assumption is satisfied, because $v \geq 1 - 1/n^{4n}$ is equivalent to $n^{4n} \leq 1/(1-v)$ and

$$n^{4n} = 2^{4n \log n} \leq 2^{4 \frac{\log \omega}{4 \log \log \omega} \log \frac{\log \omega}{4 \log \log \omega}} < 2^{\frac{\log \omega}{\log \log \omega} \log \log \omega} = \omega \leq \frac{1}{1-v}.$$

The algorithm runs in time polynomial in m as $n^n < n^{4n} \leq \omega \leq m$. To estimate the fraction of satisfied constraints, observe that $v \geq \text{Opt}(\mathcal{I}) - 1/m = 1 - \varepsilon - 1/m \geq 1 - 2\varepsilon$, so $1/(1-v) \geq 1/2\varepsilon$, and also $m \geq 1/\varepsilon$, therefore $\omega \geq 1/2\varepsilon$. The fraction of satisfied constraints is at least $1 - K/n$ and

$$\frac{n}{K} \geq \frac{1}{K} \left(\frac{\log \omega}{4 \log \log \omega} - 1 \right) \geq K_3 \frac{\log(1/2\varepsilon)}{\log \log(1/2\varepsilon)} \geq K_4 \frac{\log(1/\varepsilon)}{\log \log(1/\varepsilon)},$$

where K_3, K_4 are suitable constants. Therefore the fraction of satisfied constraints is at least

$$1 - O\left(\frac{\log \log(1/\varepsilon)}{\log(1/\varepsilon)}\right).$$

Derandomization

We start by describing the changes in Theorem 4.1. The statement remains the same except the algorithm will be polynomial in m and $2^{n^2 \log^2 n}$.

The random choices in Step 1 and Step 4 can be easily avoided: In Step 1 we can try all $(n-1)$ possible choices for r and in Step 4 we can try all choices for s from some sufficiently dense finite set, for instance $\{0, u_2/n^4, 2u_2/n^4, \dots\}$. The only difference is that bad choices for s could cover a slightly bigger part of the interval than u_1/u_2 and we would get a slightly worse constant K_1 .

For derandomization of Step 6 we first slightly change the constant in the definition of $t(\mathbf{w})$, say $t(\mathbf{w}) = \lceil 4(\log n) \dots \rceil$. Next we use Theorem 1.3. from [18] from which it follows that we can efficiently find a set Q of unit vectors such that

$$|Q| = (|V||D|)^{1+o(1)} 2^{O(\log^2(1/\kappa))}$$

and such that, for any vectors \mathbf{v}, \mathbf{w} with angle θ between them, the probability that a randomly chosen vector from Q cuts \mathbf{v} and \mathbf{w} differs from θ/π by at most κ . We choose $\kappa = 1/n^{2n} = 1/2^{2n \log n}$, therefore

$$|Q| \leq K_5 m^{K_6} 2^{n^2 \log^2 n},$$

where we have used $|V| = O(m)$ which is true whenever every variable is in the scope of some constraint (we can clearly assume this without loss of generality).

Now if we choose $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_{\lceil 4(\log n)n^{2n} \rceil}$ uniformly at random from Q , the estimates derived in Steps 8 and 9 remain almost unchanged: The probability that \mathbf{q}_i does not cut \mathbf{x}_A and \mathbf{x}_B in Step 8 is at most $1 - n^{-2r}/\pi \|\mathbf{x}_A\| + \kappa \leq 1 - n^{-2r}/4 \|\mathbf{x}_A\|$ (for a sufficiently large n), and the probability that vectors \mathbf{x}_A and \mathbf{y}_B are cut by some \mathbf{q}_i in Step 9 is at most K'_2/n (for any $K'_2 > K_2$).

Of course we cannot try all possible $\lceil 4(\log n)n^{2n} \rceil$ -tuples of vectors from Q as there are too many. However, we can apply the method of conditional expectations – we choose the vectors one by one keeping an estimate of the expected number of constraints removed below K/n .

Finally, the proof of the deterministic version of Theorem 2.4 remains almost the same except we need to ensure that $2^{n^2 \log^2 n}$ is polynomial in m . Therefore we need to choose a smaller value for n , say

$$n = \left\lfloor \frac{\sqrt{\log \omega}}{\log \log \omega} \right\rfloor,$$

then the algorithm outputs an assignment satisfying at least $\left(1 - O\left(\frac{\log \log(1/\varepsilon)}{\sqrt{\log(1/\varepsilon)}}\right)\right)$ -fraction of the constraints.

Algebraic closure of a weak Prague instance

Proposition 5.4 below justifies the last sentence in the proof of Theorem 4.1. But first we collect some useful facts about Prague instances.

It will be convenient to replace (P2) with an alternative condition:

Lemma 5.2. *Let \mathcal{J} be a 1-minimal instance. Then (P2) is equivalent to the following condition.*

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

Proof. (P2*) \Rightarrow (P2). If $p = (x = x_1, x_2, \dots, x_k = x)$ is a pattern from x to x such that $A + p = A$, then repeated application of (P2*) gives us

$$\begin{aligned} A + p - p &= [A + (x_1, x_2, \dots, x_{k-1})] + (x_{k-1}, x_k, x_{k-1}) + (x_{k-1}, x_{k-2}, \dots, x_1) \\ &= A + (x_1, x_2, \dots, x_{k-1}) + (x_{k-1}, x_{k-2}, \dots, x_1) \\ &= [A + (x_1, x_2, \dots, x_{k-2})] + (x_{k-2}, x_{k-1}, x_{k-2}) + (x_{k-2}, x_{k-3}, \dots, x_1) \\ &= A + (x_1, x_2, \dots, x_{k-2}) + (x_{k-2}, x_{k-3}, \dots, x_1) \\ &= \dots \\ &= A. \end{aligned}$$

(P2) \Rightarrow (P2*). By applying (P2) to the pattern $(x, y) + p$ we get $A + (x, y) + p - p + (y, x) = A$. From 1-minimality it follows that $A + (x, y) \subseteq A + (x, y) + p - p$, hence $A + (x, y, x) = (A + (x, y)) + (y, x) \subseteq (A + (x, y) + p - p) + (y, x) = A$. The other inclusion follows again from 1-minimality. \square

The next lemma shows that when we start with an element and keep adding a pattern from x to x , the process will stabilize.

Lemma 5.3. *Let \mathcal{J} be a weak Prague instance, $x \in V$, $a \in P_x$, and let p be a pattern from x to x . Then there exists a natural number l such that the set $[a]_p := \{a\} + lp$ satisfies $[a]_p + p = [a]_p$ and $a \in [a]_p$.*

Proof. Because the domain is finite there exist positive integers l and l' such that $\{a\} + lp + l'p = a + l'p$. As $[a]_p + p + (l' - 1)p = [a]_p$ it follows from (P3) that $[a]_p + p = [a]_p$. By 1-minimality, a is in $\{a\} + lp - lp$ which is equal to $[a]_p$ by (P2). \square

Proposition 5.4. *Let $\mathcal{J} = (V, D, \{P_{x,y} : (x, y) \in \mathcal{S}\})$ be a weak Prague instance and let \mathcal{F} be a set of operations on D . Then $\mathcal{J}' = (V, D, \{P'_{x,y} : (x, y) \in \mathcal{S}\})$, where*

$$P'_{x,y} = \{(f(a_1, a_2, \dots), f(b_1, b_2, \dots)) : f \in \mathcal{F}, (a_1, b_1), (a_2, b_2), \dots \in P_{x,y}\},$$

is a weak Prague instance.

Proof. It is apparent that \mathcal{J}' is 1-minimal with

$$P_x^{\mathcal{J}'} = P'_x := \{f(a_1, a_2, \dots) : f \in \mathcal{F}, a_1, a_2, \dots \in P_x\}.$$

In what follows, by $A +' p$ we mean the addition computed in the instance \mathcal{J}' while $A + p$ is computed in \mathcal{J} .

Before proving (P2*) and (P3) we make a simple observation.

Claim 5.4.1. *If $f \in \mathcal{F}$ is an operation of arity k , $x \in V$, p is a pattern from x , and $A_1, \dots, A_k \subseteq P_x$, $B \subseteq P'_x$ are such that $f(A_1, A_2, \dots, A_k) \subseteq B$, then $f(A_1 + p, A_2 + p, \dots, A_k + p) \subseteq B +' p$.*

Remark. By $f(A_1, \dots, A_k)$ we mean the set

$$f(A_1, A_2, \dots, A_k) = \{f(a_1, \dots, a_k) : a_1 \in A_1, a_2 \in A_2, \dots, a_k \in A_k\}.$$

Proof. It is enough to prove the claim for a single step $p = (x, y)$. The rest follows by induction. If $b \in f(A_1 + (x, y), \dots, A_k + (x, y))$ then there exist elements $b_1 \in A_1 + (x, y), \dots, b_k \in A_k + (x, y)$ so that $f(b_1, b_2, \dots, b_k) = b$. As $b_i \in A_i + (x, y)$ there are elements $a_i \in A_i$ such that $(a_i, b_i) \in P_{x,y}$ for all $1 \leq i \leq k$. But then $(f(a_1, a_2, \dots, a_k), f(b_1, b_2, \dots, b_k))$ is in $P'_{x,y}$ and $f(a_1, a_2, \dots, a_k) \in f(A_1, A_2, \dots, A_k) \subseteq B$, therefore $b = f(b_1, b_2, \dots, b_k) \in A +' (x, y)$. \square

To prove (P2*) for \mathcal{J}' let (x, y) be a step, $A \subseteq P'_x$, let p be a pattern from y to x such that $A +' (x, y) +' p = A$, and let a be an arbitrary element of $A +' (x, y, x)$. As $A +' (x, y, x) = (A +' (x, y)) +' (y, x)$, there exist $b \in A +' (x, y)$ such that $(a, b) \in P'_{x,y}$. By definition of $P'_{x,y}$, we can find $f \in \mathcal{F}$ (say, of arity k), elements a_1, a_2, \dots, a_k in P_x , and b_1, \dots, b_k in P_y so that $(f(a_1, a_2, \dots, a_k), f(b_1, b_2, \dots, b_k)) = (a, b)$ and $(a_i, b_i) \in P_{x,y}$ for all $1 \leq i \leq k$.

We consider the sets $[b_1]_q, [b_2]_q, \dots, [b_k]_q$ from Lemma 5.3 for the pattern $q = p + (x, y)$. We take l to be the maximum of the numbers for b_1, \dots, b_k from this lemma, so $[b_i]_q = b_i + lq$. We get

$$a_i \in \{b_i\} + (y, x) \subseteq [b_i]_q + (y, x) = [b_i]_q + p + (x, y) + (y, x) = [b_i]_q + p,$$

where the first step follows from $(a_i, b_i) \in P_{x,y}$, the inclusion and the first equality from Lemma 5.3, and the second equality from (P2*) for the instance \mathcal{J} (as $([b_i]_q + p) + (x, y) + p = [b_i]_q + p$). Thus $a = f(a_1, a_2, \dots, a_k)$ is an element of

$$f([b_1]_q + p, [b_2]_q + p, \dots, [b_k]_q + p) = f(\{b_1\} + lq + p, \dots, \{b_k\} + lq + p)$$

and this set is contained in $(A +' (x, y)) +' lq +' p = A +' (x, y) +' l(p + (x, y)) +' p = A$ by Claim 5.4.1 applied with $A_i = \{b_i\}$ and the pattern $lq + p$. We have shown that every element a of $A +' (x, y, x)$ lies in A . The other inclusion follows from 1-minimality.

To prove (P3) let $x \in V$, $A \subseteq P'_x$ and let p, q be patterns such that $A +' p +' q = A$. We first show that $A \subseteq A +' p$. Let $a \in P'_x$, take $f \in \mathcal{F}$, $a_1, a_2, \dots, a_k \in P_x$ such that $f(a_1, \dots, a_k) = a$, and find l

so that $[a_i]_{p+q} = a_i + l(p+q)$. From (P3) for \mathcal{J} and Lemma 5.3 it follows that $[a_i]_{p+q} + p = [a_i]_{p+q}$. By Claim 5.4.1, $a \in f([a_1]_{p+q}, [a_2]_{p+q}, \dots, [a_k]_{p+q}) = f([a_1]_{p+q} + p, [a_2]_{p+q} + p, \dots, [a_k]_{p+q} + p) \subseteq A + l(p+q) + p = A + p$. The same argument used for $A + p$ instead of A and the patterns $q + p, q$ instead of $p + q, p$ proves $A + p \subseteq A + p + q = A$. \square

Proof of Theorem 3.4

Theorem 3.4 is a generalization of a result in [1] and can be proved in a similar way. We present an alternative proof which will appear in [3].

Preliminaries

In this section we assume basic knowledge of universal algebra. Notions which are not defined here, like algebra, subuniverse, product, subdirect product, congruence, factor algebra, simple algebra, term, variety, etc. can be found in [7].

We use boldface letter for algebras, the same letter in the standard font denotes the universe of the algebra. The fact that B is a subuniverse of an algebra \mathbf{A} (or \mathbf{B} is a subalgebra of \mathbf{A}) will be denoted by $B \leq \mathbf{A}$ (or $\mathbf{B} \leq \mathbf{A}$). A subuniverse B of \mathbf{A} is proper if $\emptyset \neq B \subsetneq A$. If B is a subdirect subuniverse of $\mathbf{A}_0 \times \mathbf{A}_1 \times \dots \times \mathbf{A}_k$ we write $B \leq_S \mathbf{A}_0 \times \mathbf{A}_1 \times \dots \times \mathbf{A}_k$. In this situation, the projection kernels are denoted by π_0, π_1, \dots . By a *subpower* of \mathbf{A} we mean a subuniverse (or a subalgebra) of a power of \mathbf{A} .

We say that an instance \mathcal{I} of the CSP is *in a variety* \mathcal{V} if the domain of \mathcal{I} is the universe of some algebra \mathbf{D} in \mathcal{V} and all the constraint relations are subpowers of \mathbf{D} . A core constraint language cannot encode linear equations iff the variety generated by the algebra of polymorphisms is in a congruence meet semi-distributive ($SD(\wedge)$) variety (this follows from results in Chapter 9 of [16]). Therefore Theorem 3.4 is a consequence of the following result.

Theorem 5.5. *Every nontrivial weak Prague instance in an $SD(\wedge)$ variety has a solution.*

All instances considered in this section have the set of variables V , domain D , all constraints have binary constraint relations which are subuniverses of an algebra \mathbf{D} in a meet semi-distributive variety. The set of scopes is denoted by \mathcal{S} , we assume $\mathcal{S} = \mathcal{S}^{-1}$, $\mathcal{S} \cap \mathcal{S}^{-1} = \emptyset$, and for every $(x, y) \in \mathcal{S}$ there is exactly one constraint with this scope, its constraint relation is denoted by $P_{x,y}$ (with a superscript indicating the instance, when necessary). If the instance is 1-minimal then every $P_x^{\mathcal{J}}$ is a subuniverse of \mathbf{D} and we denote the subalgebra of \mathbf{D} with universe $P_x^{\mathcal{J}}$ by $\mathbf{P}_x^{\mathcal{J}}$ (the superscript is often omitted).

Absorption

One of the main tools for proving Theorem 5.5 is the absorption.

Definition 5.6. *We say that B is an absorbing subuniverse of an algebra \mathbf{A} , denoted by $B \triangleleft \mathbf{A}$, if $B \leq \mathbf{A}$ and there exists a term t of \mathbf{A} such that $t(B, B, \dots, B, A, B, B, \dots, B) \subseteq B$ for any position of A .*

The next lemma is a straightforward consequence of the definitions.

Lemma 5.7. *Let $R \leq_S \mathbf{A} \times \mathbf{B}$ and let C be an (absorbing) subuniverse of \mathbf{A} . Then*

$$\{d \in B : \exists c \in C (c, d) \in R\}$$

is an (absorbing) subuniverse of \mathbf{B} . The absorption is realized by the same term.

The Absorption Theorem [2] concerns linked subdirect products of algebras from Taylor varieties. We formulate it only in the special case of $SD(\wedge)$ varieties.

Definition 5.8. A subdirect subalgebra R of $\mathbf{A} \times \mathbf{B}$ is called linked, if $\pi_0 \vee \pi_1 = 1_{\mathbf{R}}$.

Theorem 5.9 (Absorption Theorem). If \mathbf{A} and \mathbf{B} are algebras in an $SD(\wedge)$ variety, $R \leq_S \mathbf{A} \times \mathbf{B}$ is linked and $R \neq A \times B$, then \mathbf{A} or \mathbf{B} has a proper absorbing subuniverse.

We also require the following corollary.

Lemma 5.10. Let \mathbf{A} and \mathbf{B} be algebras in an $SD(\wedge)$ variety such that neither \mathbf{A} nor \mathbf{B} has a proper absorbing subuniverse, let $R \leq_S \mathbf{A} \times \mathbf{B}$ and let α be a maximal congruence of \mathbf{A} . Then

(case1) either $(a, b), (a', b) \in R$ implies that $(a, a') \in \alpha$ for all $a, a' \in A$, or

(case2) for every $a \in A$ and $b \in B$ there exists $a' \in A$ such that $(a, a') \in \alpha$ and $(a', b) \in R$.

Proof. Consider the subdirect product $R' = \{(a/\alpha, b) : (a, b) \in R\} \leq_S \mathbf{A}/\alpha \times \mathbf{B}$. The algebra \mathbf{A}/α has no absorbing subuniverse (since the preimage of an absorbing subuniverse of \mathbf{A}/α is an absorbing subuniverse of \mathbf{A}) and is simple (as α is maximal). If (case1) is not satisfied then the projection of $\pi_0 \vee \pi_1$ to the coordinate \mathbf{A}/α is not the equality congruence, therefore, since \mathbf{A}/α is simple, $\pi_0 \vee \pi_1 = 1_{\mathbf{R}'}$, so R' is linked. By the Absorption theorem, $R' = A/\alpha \times B$ which is a restatement of (case2). \square

Pointed terms

The second algebraic tool are pointed terms.

Definition 5.11. Let \mathbf{A} be an algebra. A term $t(x_0, \dots, x_{n-1})$ of \mathbf{A} points to a if there exist $a_0, \dots, a_{n-1} \in A$ such that

$$t(b_0, \dots, b_{n-1}) = a \text{ whenever } b_i \in A \text{ and } |\{i : a_i \neq b_i\}| \leq 1.$$

The existence of pointed terms is proven in a sequence of lemmata.

Lemma 5.12. Let $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_n$ be algebras and let $\mathbf{R} \not\leq \prod_{i=0}^n \mathbf{A}_i$. If $\mathbf{R} \triangleleft \prod_{i=0}^n \mathbf{A}_i$ then some \mathbf{A}_i contains a proper absorbing subalgebra.

Proof. Suppose, for a contradiction, that the lemma holds for $n-1$ and fails for $\mathbf{R} \not\leq \prod_{i=0}^n \mathbf{A}_i$. The projection $\pi_0(\mathbf{R})$ is an absorbing subuniverse of \mathbf{A}_0 , so $\pi_0(\mathbf{R}) = \mathbf{A}_0$. Therefore there exists $a \in A_0$ such that

$$\emptyset \neq R' = \{(b_1, \dots, b_{n-1}) \mid (a, b_1, \dots, b_1)\} \neq \prod_{i=1}^n \mathbf{A}_i.$$

But then, since $R' \triangleleft \prod_{i=1}^n \mathbf{A}_i$, we get a proper absorbing subuniverse of one of the \mathbf{A}_i 's. \square

Lemma 5.13. Let $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_n$ be simple algebras with no proper absorbing subuniverses lying in an $SD(\wedge)$ variety. If $\mathbf{R} \leq_S \prod_{i=0}^n \mathbf{A}_i$ and $\pi_i \vee \pi_j = 1_{\mathbf{R}}$ for every $i \neq j$, then $\mathbf{R} = \prod_{i=0}^n \mathbf{A}_i$.

Proof. First we prove that there exists j such that $\bigwedge_{i \neq j} \pi_i \neq 0_{\mathbf{R}}$. Indeed, otherwise the $SD(\wedge)$ congruence identity gives $\bigwedge_{i \neq j} (\pi_i \vee \pi_j) = 0_{\mathbf{R}}$ which is a contradiction.

The projection of $\pi_0 \vee \bigwedge_{i \neq 0} \pi_i$ to the 0-th coordinate is not the equality congruence, and, since \mathbf{A}_0 is simple, $\pi_0 \vee \bigwedge_{i \neq 0} \pi_i = 1_{\mathbf{R}}$.

Suppose, for a contradiction, that the lemma holds for $n - 1$ and fails for $\mathbf{R} \leq_S \prod_{i=0}^n \mathbf{A}_i$. By this assumption, the projection of \mathbf{R} to the coordinates $1, 2, \dots, n$ is equal to $\prod_{i=1}^n \mathbf{A}_i$ and \mathbf{R} is a subdirect product of \mathbf{A}_0 and $\prod_{i=1}^n \mathbf{A}_i$. By the previous paragraph, R is linked and therefore, by the Absorption Theorem, R is either equal to $A_0 \times \prod_{i=1}^n A_i$, or \mathbf{A}_0 contains a proper absorbing subuniverse – which contradicts the assumption, or $\prod_{i=1}^n \mathbf{A}_i$ contains a proper absorbing subuniverse – which contradicts Lemma 5.12. \square

Lemma 5.14. *Let \mathbf{A} be a simple algebra in an $SD(\wedge)$ variety with no proper absorbing subuniverse. Let n be an arbitrary positive integer, and let $\mathcal{Z} \subseteq A^n$ be such that:*

$$\forall a \neq b \in \mathcal{Z} \exists i, j (a_i = a_j \wedge b_i \neq b_j) \vee (a_i \neq a_j \wedge b_i = b_j)$$

and

$$\forall a \in \mathcal{Z} \text{Sg}_{\mathbf{A}}(\{a_0, \dots, a_{n-1}\}) = \mathbf{A}.$$

Then every function from \mathcal{Z} to A is a restriction of some n -ary term operation of \mathbf{A} .

Proof. Let $\mathbf{F} \leq \mathbf{A}^{A^n}$ be the free algebra on n -generators (the universe consists of the n -ary term operations of \mathbf{A}).

Choose a, b to be two arbitrary tuples from \mathcal{Z} and let \mathbf{R} be the projection of \mathbf{F} to the coordinates a, b . Since \mathbf{F} is generated by projections, we have $(a_i, b_i) \in \mathbf{R}$ for every $i < n$. By the second condition on \mathcal{Z} the algebra \mathbf{R} is subdirect in $\mathbf{A} \times \mathbf{A}$. The first condition provides i, j with, say, $a_i = a_j$ and $b_i \neq b_j$ (the other case is similar). Thus the projection of $\pi_0 \vee \pi_1$ (computed in \mathbf{R}) to the second coordinate is not the equality congruence, and, by simplicity of \mathbf{A} , it is the full congruence. That means that \mathbf{R} is linked, so $\pi_a \vee \pi_b = 1_{\mathbf{F}}$.

By the previous paragraph, the restriction of \mathbf{F} to \mathcal{Z} satisfies the assumptions of Lemma 5.13, and therefore is the full relation. In other words, every function from \mathcal{Z} to \mathbf{A} extends to a term operation of \mathbf{A} . \square

Corollary 5.15. *Let \mathbf{A} be a simple algebra with no proper absorbing subuniverse in an $SD(\wedge)$ variety. Then for every $a \in A$ there exists a term of \mathbf{A} which points to a .*

Proof. Let $A = \{a_0, \dots, a_{n-1}\}$ and $b = (a_0, a_0, a_1, a_1, \dots, a_{n-1}, a_{n-1})$. Putting

$$\mathcal{Z} = \{c \in \mathbf{A}^{2n} : |\{i : c_i \neq b_i\}| \leq 1\}$$

we obtain a set satisfying the assumption of Lemma 5.14 and therefore the constant function mapping \mathcal{Z} to a extends to a term operation – the term points to a . \square

Pointed decomposition

The proof of Theorem 5.5 will proceed by reducing a given weak Prague instance to a smaller induced subinstances.

Definition 5.16. *Let \mathcal{J} be an instance. Then \mathcal{J}' is an induced subinstance of \mathcal{J} with potatoes $P'_x \leq \mathbf{D}$, $x \in V$ if $P'_{x,y} = P_{x,y} \cap (P'_x \times P'_y)$.*

In order to succeed with such a reduction we need to decompose the instance first.

Definition 5.17. *A decomposition of a 1-minimal instance \mathcal{J} consists of induced subinstances $\mathcal{J}^1, \dots, \mathcal{J}^l$ of \mathcal{J} with potatoes $P_x^k \leq \mathbf{P}_x$, $x \in V$, $k \leq l$, and a subset X of V such that*

- if $x \notin X$ then $P_x^k = P_x$ for all $k \leq l$,

- if $x \in X$ then
 - $P_x^k \cap P_x^{k'} = \emptyset$ for all $k \neq k'$, and
 - for any step (x, y) either $P_x^k + (x, y) = P_y^k$ for all $k \leq l$, or $P_x^k + (x, y) = P_y$ for all $k \leq l$.

We say that the decomposition is pointed, if there exists a term t of \mathbf{D} of arity m and indices k_1, \dots, k_m such that

$$\forall o \leq k \forall x \in V \ t(P_x^{k_1}, \dots, P_x^{k_{o-1}}, P_x, P_x^{k_{o+1}}, \dots, P_x^{k_m}) \subseteq P_x^1.$$

The decomposition is proper if \mathcal{J}^1 is a nontrivial instance and $P_x^1 \subsetneq P_x$ for some $x \in V$.

In the following two subsections we prove:

Theorem 5.18. *Every nontrivial weak Prague instance \mathcal{J} with $|P_x| > 1$ for some $x \in V$ has a proper pointed decomposition.*

In the last subsection we show that \mathcal{J}^1 is a weak Prague instance. These two facts imply Theorem 5.5: A minimal induced subinstance \mathcal{I} of any weak Prague instance has $|P_x^{\mathcal{I}}| = 1$, $x \in V$ and such an instance trivially has a solution (which sends x to the unique element of $P_x^{\mathcal{I}}$).

Pointed decomposition in absorption case

We prove Theorem 5.18 in the case that some algebra \mathbf{P}_x has a proper absorbing subuniverse A .

We define a preorder on the set of all pairs (C, y) , $\emptyset \neq C \subsetneq P_y$ by $(C, y) \sqsubseteq (D, z)$ if there exists a pattern p from y to z such that $C + p = D$. Among the equivalence classes of this preorder which are greater than A we choose a maximal and denote it by \mathcal{M} .

Let X denote the set of all $y \in V$ for which there exists some C such that $(C, y) \in \mathcal{M}$. The set C is uniquely determined by y : If $(C, y), (D, y) \in \mathcal{M}$ then by the fact that (C, y) and (D, y) are in the same equivalence class of \sqsubseteq we get $C + p = D$ and $D + q = C$ for some patterns p and q from y to y , and, by (P3), $C = D$. We put P_y^1 to be the unique set with $(P_y^1, y) \in \mathcal{M}$. For $y \notin X$ we put $P_y^1 = P_y$.

We show that the induced subinstance \mathcal{J}^1 of \mathcal{J} with potatoes P_y^1 , $y \in V$ is a pointed decomposition of \mathcal{J} .

By definition, $P_y^1 = A + p$ for some pattern p from x to y . Lemma 5.7 applied step by step shows that P_y^1 is an absorbing subuniverse of P_y and the absorbing term is the same as for A and \mathbf{P}_x . This term satisfies the condition in Definition 5.17 with $k_i = 1$ for all i .

Let $y \in X$ be arbitrary and let (y, z) be any step. If $(P_y^1 + (y, z), z) \in \mathcal{M}$ then $P_y^1 + (y, z) = P_z^1$ as required. On the other hand if $(P_y^1 + (y, z), z) \notin \mathcal{M}$ then this pair is outside of the preorder, therefore $P_y^1 + (y, z) = P_z$.

Pointed decomposition in no absorption case

Now we prove Theorem 5.18 in the case that none of the algebras \mathbf{P}_x , $x \in V$ has a proper absorbing subuniverse.

Lemma 5.19. *Let α be a maximal congruence on some \mathbf{P}_x . Then for every step (x, y)*

(case1) either $(a/\alpha + (x, y)) \cap (b/\alpha + (x, y)) \neq \emptyset$ implies $(a, b) \in \alpha$ for all $a, b \in P_x$,

(case2) or $a/\alpha + (x, y) = P_y$ for all $a \in P_x$.

Proof. The lemma is a restatement of Lemma 5.10. □

If \mathbf{P}_x and α are as in the previous lemma then

$$\{a/\alpha + (x, y) : a \in P_x\}$$

is a partition of P_y and the corresponding equivalence relation, denoted by $\alpha + (x, y)$, is easily seen to be a congruence of \mathbf{P}_y . Moreover in (case1) the function

$$\{(a/\alpha, b/(\alpha + (x, y))) : (a, b) \in C\}$$

provides an isomorphism between \mathbf{P}_x/α and $\mathbf{P}_y/(\alpha + (x, y))$. Therefore, in this case, $\alpha + (x, y)$ is a maximal congruence of \mathbf{P}_y and $(\alpha + (x, y)) + (y, x) = \alpha$.

Lemma 5.20. *Let α be a maximal congruence on \mathbf{P}_x , $a \in P_x$, and let p be a pattern from x to y . If $a/\alpha + p \not\subseteq P_y$ then for any $a' \in P_x$ we have $a'/\alpha + p - p = a'/\alpha$.*

Proof. The proof is by induction on the length of p . Let $p = (x, z) + p'$. Then $a/\alpha + (x, z) \not\subseteq P_z$ and (case1) of Lemma 5.19 (used for the step (x, z)) applies. Therefore $\alpha + (x, z)$ is a maximal congruence on \mathbf{P}_z , both $a/\alpha + (x, z)$ and $a'/\alpha + (x, z)$ are its congruence classes and $a'/\alpha + (x, z) + (z, x) = a'/\alpha$. By inductive assumption $(a'/\alpha + (x, z)) + p' - p' = (a'/\alpha + (x, z))$ and the lemma is proved. □

If \mathbf{P}_x and α are as in the previous lemma, $a/\alpha + p \not\subseteq P_y$, and q is another pattern from x to y such that $a/\alpha + q \not\subseteq P_y$ then, by the previous lemma, $(a/\alpha + p) - p + q = a/\alpha + q$ and $(a/\alpha + q) - q + p = a/\alpha + p$, and using (P3) we get $a/\alpha + p = a/\alpha + q$.

Now we are ready to define the decomposition. We assume that P_x has at least two elements and put α to be a maximal congruence on \mathbf{P}_x . We denote by P_x^1, \dots, P_x^l the equivalence classes of α and will decompose \mathcal{J} into $\mathcal{J}^1, \dots, \mathcal{J}^l$. We include y into X if there is a pattern p from x to y such that $P_x^1 + p \not\subseteq P_y$ and, in this case, we set $P_y^k = P_x^k + p$ (by the discussion after the lemma, the definition is independent on the choice of p). If $y \notin X$ let $P_y^k = P_y$.

Let y be an arbitrary variable in X . By Lemma 5.20, $P_y^k \cap P_y^{k'} = \emptyset$ for any $k \neq k'$. Choose any step (y, z) , and suppose that $P_y^k + (y, z) \not\subseteq P_z$ for some k . Then, by Lemma 5.20 again, $P_y^k + (y, z) \not\subseteq P_z$ for all k and $P_y^k + (y, z) = P_z^k$ as required. Therefore the induced subinstances $\mathcal{J}^1, \dots, \mathcal{J}^l$ with potatoes $P_x^k, x \in V$ form a decomposition.

Clearly, for any k , P_x^k is a subuniverse of \mathbf{P}_x and therefore, by Lemma 5.7, P_y^k is a subuniverse of \mathbf{P}_y . In order to find a pointed term for this decomposition consider \mathbf{P}_x/α . This algebra is simple, has no proper absorbing subuniverses (since \mathbf{P}_x does not) and lies in an $\text{SD}(\wedge)$ variety. Therefore, by Corollary 5.15, there exists a term $t(x_1, \dots, x_m)$ in \mathbf{P}_x/α pointing to P_x^1/α .

This term clearly satisfies the required condition for x and for $y \notin X$. On the other hand if $y \in X$ then $\mathbf{P}_y/(\alpha + p)$ is isomorphic to \mathbf{P}_x/α via an isomorphism sending \mathbf{P}_x^1 to \mathbf{P}_y^1 and therefore $t(x_1, \dots, x_m)$ satisfies the condition for \mathbf{P}_y as well. This finishes the case when there is no absorption in the Prague instance.

Reducing to the point

The following theorem finishes the proof of the main result.

Theorem 5.21. *Let \mathcal{J} be a weak Prague instance with a pointed decomposition $\mathcal{J}^1, \dots, \mathcal{J}^l$ with potatoes $P_x^k, x \in V, k \leq l$. Then \mathcal{J}^1 is a weak Prague instance.*

We start with a definition of relativized addition of patterns. Throught the proof we need to compute steps (patterns) in subinstances $\mathcal{J}^1, \dots, \mathcal{J}^l$.

For $A \subseteq P_x^k$ we denote by $A +^k(x, y)$ the addition computed in \mathcal{J}^k (i.e. $A +^k(x, y) = (A + (x, y)) \cap P_y^k$). The definition naturally extends to patterns. Sometimes, working in instance \mathcal{J} , we need to cross-over from \mathcal{J}^k to $\mathcal{J}^{k'}$. To do that we introduce $A +^{k,k'}(x, y) = (A + (x, y)) \cap P_y^{k'}$.

Theorem 5.21 is a consequence on the following lemma:

Lemma 5.22. *Let $x \in V$, $A \subseteq P_x^1$ and let p be a pattern from x to x . If $A +^1 p = A$ then $A = (A + p) \cap P_x^1$.*

Using this lemma we prove Theorem 5.21.

Proof of Theorem 5.21. It follows from the definition of decomposition that \mathcal{J}^1 (as well as $\mathcal{J}^2, \dots, \mathcal{J}^l$) is 1-minimal.

(P3). Let $A \subseteq P_x^1$, let p, q be patterns from x to x such that $A +^1 p +^1 q = A$ and let $B = A +^1 p$. Since the domain is finite, there is j such that $A + j(p + q) = A' = A' + j(p + q)$ for some $A' \subseteq P_x$. Putting $B' = A' + p$ we obtain $B' + q + (j - 1)(p + q) = A'$ and, by (P3), $A' = B'$. Then, by Lemma 5.22,

$$A = (A + j(p + q)) \cap P_x^1 = A' \cap P_x^1 = B' \cap P_x^1 \supseteq B \cap P_x^1 = B.$$

Similarly $B \supseteq A$ and finally $A = B$ as required.

(P2). If $A +^1 p = A$, where $A \subseteq P_x$ and p is a pattern from x to x , then there is j and $A' \subseteq P_x$ such that $A + jp = A' = A' + jp$ and, by (P2), $A' - jp = A'$. By Lemma 5.22, $A = A' \cap P_x^1$ and we have $A -^1 jp \subseteq (A' - jp) \cap P_x^1 = A' \cap P_x^1 = A$. On the other hand, since $A +^1 jp = A$ we get $A \subseteq A -^1 jp$. Thus $A = A -^1 jp$ as required. \square

The remaining part of this section is devoted to the proof of Lemma 5.22. Let $\mathcal{J}, \mathcal{J}^1, \dots, \mathcal{J}^l$ as well as x, A, p be as in the statement of the lemma.

Note that, as $A +^1 p = A$, we have $A \subseteq A + p \subseteq A + 2p \subseteq \dots$ and therefore it suffices to prove the claim for some pattern jp in place of p . Since the domain is finite, we can find j and $A' \subseteq P_x$ such that $A' = A + jp$, $A' + jp = A'$ and therefore, by (P2), $A' - jp = A'$. By enlarging j further we can guarantee that for any $B \subseteq P_x$ and any j' we have

- $B + j'p = A'$ if and only if $B + jp = A'$ and
- $B - j'p = A'$ if and only if $B - jp = A'$.

We substitute p with jp and get $A + p = A' = A' + p = A' - p$ and, for any j' , $B + j'p = A'$ ($B - j'p = A'$) if and only if $B + p = A'$ ($B - p = A'$ respectively). We put A'' to denote $A' \cap P_x^1$ and since, clearly, $A \subseteq (A + p) \cap P_x^1 = A''$ we aim to prove the inverse containment.

Let $t(x_1, \dots, x_m)$ be the term from the definition of pointed decomposition. By induction on $0 \leq o \leq m$ we prove the following lemma.

Lemma 5.23. *There exist $A_{o+1}, \dots, A_m \subseteq A''$ such that*

$$t(\underbrace{A'', \dots, A''}_o, A_{o+1}, \dots, A_m) \subseteq A$$

and $A_i + p = A'$ for every $i > o$.

Note that the lemma holds for $o = 0$ by putting $A_i = A$ for all i , and if it holds for $o = m$ then $A'' \subseteq t(A'', \dots, A'') \subseteq A$ and Lemma 5.22 is proved.

The remainder of this section is devoted to the proof of Lemma 5.23. The proof requires two technical claims.

Claim 5.23.1. *For every $B \subseteq A'$ we have $B + p = A'$ if and only if $B - p = A'$.*

Proof. Let $B \subseteq A'$ and $B + p = A'$. Since the instance is finite, there exists j such that $B' := B - jp = B' - jp$, but then, by (P2), $B' + jp = B'$. Since \mathcal{J} is 1-minimal, we have $B - jp + jp \supseteq B$, hence $B' = B' + 2jp \supseteq B + jp = A'$. As $B \subseteq A'$ we get $B' = B - jp \subseteq A' - jp = A'$ and therefore $B - jp = A'$. By the properties of A' and B stated above, we have $B - p = A'$ and one of the implications of the claim holds. The proof of the inverse implication is analogical. \square

Next we split the pattern p into parts. Let $p = q + (y, z) + r + (v, w) + s$ where q is a maximal initial fragment of p consisting of steps (y, z) such that $P_y^1 + (y, z) = P_y^1 + (y, z)$ and s a maximal terminal fragment of p consisting of steps (y, z) such that $P_z^1 - (y, z) = P_y^1$ (note that some of the patterns q, r, s may be empty). If such a decomposition is impossible then it is easy to see that following the pattern p from A step by step we obtain A'' contained in A and Lemma 5.22 is proven without the help of Lemma 5.23 – this is a trivial case.

Throughout the proof we will be using variations of the pattern p evaluated in subinstance \mathcal{J}^1 except for the middle part which is evaluated in \mathcal{J}^k . That is for $B \subseteq P_x^1$ we will be evaluating

$$B +^1 q +^{1,k} (y, z) +^k r +^{k,1} (v, w) +^1 s.$$

The following claim deals with such evaluations and their inverses.

Claim 5.23.2. *We show that $P_x^1 = P_x^1 +^1 q +^{1,k} (y, z) +^k r +^{k,1} (v, w) +^1 s$ and $P_x^1 = P_x^1 -^1 s -^{k,1} (v, w) -^k r -^{1,k} (y, z) -^1 q$ for any k .*

Proof. Since \mathcal{J}^1 is 1-minimal $P_x^1 +^1 q = P_y^1$. If $k = 1$ we get $P_y^1 +^{1,k} (y, z) = P_z^k$ from 1-minimality of \mathcal{J}^1 . If $k \neq 1$ then, by the choice of the partition of p , we get $P_y^1 + (y, z) \neq P_z^1$ and, by definition of decomposition, $P_y^1 + (y, z) = P_z$. Thus $P_y^1 +^{1,k} (y, z) = P_z^k$. Since \mathcal{J}^k is 1-minimal, using previous facts, we get $P_x^1 +^1 q +^{1,k} (y, z) +^k r = P_v^k$.

If $k = 1$ we get $P_v^1 +^{k,1} (v, w) = P_z^1$ from 1-minimality of \mathcal{J}^1 . If $k \neq 1$ then, by the choice of the partition of p , we get $P_w^1 - (v, w) = P_w^1 + (w, v) \neq P_z^1$ and, by definition of decomposition, $P_w^1 + (w, v) = P_v$. Thus $(P_v^k + (v, w)) \cap P_w^1 \neq \emptyset$ and, by the definition of decomposition $P_v^k + (v, w) = P_w$ and further $P_v^k +^{k,1} (v, w) = P_w^1$. For the pattern s we use, once again, 1-minimality of \mathcal{J}^1 . All these facts together imply that $P_x^1 = P_x^1 +^1 q +^{1,k} (y, z) +^k r +^{k,1} (v, w) +^1 s$.

The reasoning for the inverse pattern is an alphabetical variant of the proof above. \square

Note that the second equality of Claim 5.23.2 implies that, for any $B \subseteq P_x^1$ we have

$$B \subseteq B +^1 q +^{1,k} (y, z) +^k r +^{k,1} (v, w) +^1 s - p.$$

Proof of Lemma 5.23. Let $0 \leq o < m$ and $A_{o+1}, \dots, A_m \subseteq A''$ satisfy Lemma 5.23 with a term t and indices k_1, \dots, k_m such that

$$t(P_y^{k_1}, \dots, P_y^{k_{o'-1}}, P_y, P_y^{k_{o'+1}}, \dots, P_y^{k_m}) \subseteq P_y^1, \forall y, o'.$$

The proof proceeds by evaluating certain propagation of sets A'' and A_j along the pattern p . The set appearing on the j -th coordinate ($j \neq o + 1$) of $t(x_1, \dots, x_m)$ will be propagated along p

within \mathcal{J}^1 except that middle part of the pattern will be propagated in \mathcal{J}^{k_j} . On the $(o+1)$ -th coordinate we follow p with no restrictions.

Thus $A'_j = A_j + {}^1q + {}^{1,k_j}(y, z) + {}^{k_j}r + {}^{k_j,1}(v, w) + {}^1s$ for $j > o+1$, for $j < o+1$ we have $A'' = A'' + {}^1q + {}^{1,k_j}(y, z) + {}^{k_j}r + {}^{k_j,1}(v, w) + {}^1s$ (by Claim 5.23.2) and on the coordinate $o+1$ we put $A'_{o+1} = A_{o+1} + p$.

Let j be arbitrary. Since $A_j + p = A'$ then, by Claim 5.23.1, $A_j - p = A'$. By definition of A'_j , using the remark after Claim 5.23.2, $A'_j - 2p \supseteq A_j - p = A'$. On the other hand $A'_j - 2p \subseteq A' - 2p = A'$. Thus $A'_j - 2p = A'$ and, by the choice of p, A' and Claim 5.23.1, $A_j + p = A'$. This proves a part of Lemma 5.23, it remains to prove that $t(A'', \dots, A'', A'_{o+2}, \dots, A'_m) \subseteq A$.

We prove that $t(A'', \dots, A'_m) \subseteq A$ step by step. First we show that

$$t(A'' + {}^1q, \dots, A_{o+1} + q, \dots, A_m + {}^1q) \subseteq A + {}^1q.$$

Since every constraint relation is closed under $t(x_1, \dots, x_m)$, we have $t(A'' + q, \dots, A_m + q) \subseteq A + q$ and since, by the choice of q , we have $A_j + {}^1q = A_j + q$ and $A'' + {}^1q = A'' + q$, we get $t(A'' + {}^1q, \dots, A_{o+1} + q, \dots, A_m + {}^1q) \subseteq A + {}^1q$ as required.

Let (s, u) be an arbitrary step from $\dots + (y, z) + r + (v, w)$. Denote the sets obtained by the initial part of our pattern p (excluding (s, u)) by B_1, \dots, B_m (for $i \neq o+1$ the set B_i is obtained by following p in \mathcal{J}^1 with middle part in \mathcal{J}^{k_i} , for B_{o+1} we just follow p). By the construction $t(B_1, \dots, B_m) \subseteq B$ (where B is obtained by following the same initial part in \mathcal{J}^1) and we want to extend this inclusion by the step (s, u) .

Let $B'_{o+1} = B_{o+1} + (s, u)$ and let $B'_i = B_i + {}^{1,k_i}(s, u)$ in case of the first step, $B'_i = B_i + {}^{k_i}(s, u)$ in the middle step and $B'_i = B_i + {}^{k_i,1}(s, u)$ in case of the last step, for $i \neq o+1$. For any $b \in t(B'_1, \dots, B'_m)$ we find a_i, b_i such that $t(b_1, \dots, b_m) = b$, $a_i \in B_i$, $b_i \in B'_i$ and $(a_i, b_i) \in P_{s,u}$. Now $a = t(a_1, \dots, a_m) \in B$ (as $t(B_1, \dots, B_m) \subseteq B$) and $(a, b) = (t(a_1, \dots, a_m), t(b_1, \dots, b_m)) \in P_{s,u}$ (as t is a polymorphism of $P_{s,u}$), therefore $t(b_1, \dots, b_m) \in B + (s, u)$. But also $t(b_1, \dots, b_m) \in P_u^1$ since either all the arguments are in P_u^1 , or $b_i \in P_u^{k_i}$ for $i \neq o+1$ and we can use the property $t(P_u^{k_1}, \dots, P_u^{k_o}, P_u, P_u^{k_{o+2}}, \dots, P_u^{k_m}) \subseteq P_u^1$. Put together, $b = t(b_1, \dots, b_m) \in B + {}^1(s, u)$, which extends the inclusion to the next step.

The computations for the pattern s are analogical to those for q and finally we get $t(A'', \dots, A'_m) \subseteq A + {}^1p = A$. This finishes the proof of the lemma. \square