Dichotomy for Symmetric Boolean PCSPs

Miron Ficak
Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland
miron.ficak@student.uj.edu.pl

Marcin Kozik
Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland
marcin.kozik@uj.edu.pl

Miroslav Olšák
Department of Algebra, Charles University, Prague, Czech Republic
mirek@olsak.net

Szymon Stankiewicz
Theoretical Computer Science Department, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland
szymon.stankiewicz@student.uj.edu.pl

Abstract

In one of the most actively studied version of Constraint Satisfaction Problem, a CSP is defined by a relational structure called a template. In the decision version of the problem the goal is to determine whether a structure given on input admits a homomorphism into this template. Two recent independent results of Bulatov [FOCS'17] and Zhuk [FOCS'17] state that each finite template defines CSP which is tractable or NP-complete.

In a recent paper Brakensiek and Guruswami [SODA'18] proposed an extension of the CSP framework. This extension, called Promise Constraint Satisfaction Problem, includes many naturally occurring computational questions, e.g. approximate coloring, that cannot be cast as CSPs. A PCSP is a combination of two CSPs defined by two similar templates; the computational question is to distinguish a YES instance of the first one from a NO instance of the second.

The computational complexity of many PCSPs remains unknown. Even the case of Boolean templates (solved for CSP by Schaefer [STOC'78]) remains wide open. The main result of Brakensiek and Guruswami [SODA'18] shows that Boolean PCSPs exhibit a dichotomy (PTIME vs. NPC) when "all the clauses are symmetric and allow for negation of variables". In this paper we remove the "allow for negation of variables" assumption from the theorem. The "symmetric" assumption means that changing the order of variables in a constraint does not change its satisfiability. The "negation of variables" means that both of the templates share a relation which can be used to effectively negate Boolean variables.

The main result of this paper establishes dichotomy for all the symmetric boolean templates. The tractability case of our theorem and the theorem of Brakensiek and Guruswami are almost identical. The main difference, and the main contribution of this work, is the new reason for hardness and the reasoning proving the split.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Theory of computation → Constraint and logic programming

Keywords and phrases promise constraint satisfaction problem, PCSP, algebraic approach

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.57

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is avaialable at `https://arxiv.org/abs/1904.12424`, [11] .

Funding Research was partially supported by National Science Centre, Poland grant no. 2014/2013/B/ST6/01812.

1 Introduction

Constraint Satisfaction Problems have been studied in computer science in many forms. In the general approach an instance of the CSP consists of variables and constraints. In the decision version of the problem the objective is to verify whether there exists an evaluation of variables that meets all the constraints.

One particular type of CSPs received a lot of attention in the past years. In this approach constraints are relations taken from a fixed, finite relational structure called a template. The interest in this particular version was driven by a conjecture of Feder and Vardi [10] postulating that each finite template defines a CSP which is tractable or NP-complete.

A great variety of decision problems independently studied by computer scientists can be cast as CSPs. To name a few: 3-SAT, k-colorability, (generalized) unreachability in directed graphs or solving systems of linear equation over a finite field, are all CSPs defined by finite templates. The class of all the computational problems falling into the scope of the conjecture is very big and its verification was a gradual and lengthy process. Nevertheless, from the start, the claim was supported by strong evidence. In this context the classical result of Schaefer [15] showing that the dichotomy holds for templates over Boolean domain, is perhaps the most important.

The dichotomy for all the finite templates was recently confirmed by two, independent results of Bulatov [6] and Zhuk [16]. Both of them use the algebraic approach [13, 7], where the complexity of a template is studied via compatible operations called polymorphisms. The algebraic approach proved very successful not only in the decision version of the CSP: a number of important results in optimization [14], approximation [2] etc. of the CSP is based on some versions of polymorphisms.

A positive resolution of the dichotomy conjecture motivates the following question: is the class of CSPs unique, or maybe a part of a larger, natural class which also exhibits a dichotomy? Note that such a class should be amenable to some sort of the algebraic approach, as no other tools offer comparable power even in the case of the CSP. In the recent paper [5] Brakensiek and Guruswami proposed a candidate for such a class.

The Constraint Satisfaction Problem defined by a fixed language can be cast as a problem of finding homomorphism from a relational structure given on input to a fixed template. The class proposed by Brakensiek and Guruswami as an extension of CSP is called Promise Constraint Satisfaction Problems. A PCSP is based on two CSPs with similar templates and the question is to distinguish YES instances of the first CSP from NO instances of the second.

To provide a few examples: the CSP defined by an undirected clique (without loops) of size k as a template is just k-colorability. Defining PCSP by two cliques, say of sizes k and l satisfying $k < l$, we get the following problem: distinguish between the graphs with chromatic number $\leq k$ and those with chromatic number $> l$. These problems are studied independently [9, 12, 3, 8], but the characterization of complexities for all pairs (k, l) is either incomplete or done under additional assumptions.

Another example is a Boolean PCSP. A single ternary relation $\{(0,0,1),(0,1,0),(1,0,0)\}$ defines a CSP which is known as Monotone-1-in-3-SAT, and similarly the relation $\{0,1\}^3 \setminus \{(0,0,0),(1,1,1)\}$ gives rise to the CSP known as Monotone-NAE-SAT. Thus the question of distinguishing between instances which are satisfiable as Monotone-1-in-3-SAT instances and not satisfiable as Monotone-NAE-SAT instances is a PCSP. Surprisingly this problem is tractable even allowing for the negation of variables [1, 5].

Further examples of problems expressible as PCSPs can be found in [5]. Promise Constraint Satisfaction Problems generalize CSPs and include many additional, natural problems. The algebraic approach to the CSP can be adjusted to work in the case of the PCSP. The first Galois correspondence between PCSPs and the polymorphisms was introduced in [5], and the more abstract algebraic approach was proposed in [8]. Despite all the interest, PCSPs lack a classification result that would play the role of Schaefer's theorem. This motivates a more systematic study of Boolean PCSPs.

The main result of Brakensiek and Guruswami, Theorem 2.1 in [5], establishes dichotomy for a certain class of Boolean PCSPs. A PCSP template falls into this class if all the relations in the templates are symmetric (i.e. invariant under permutations, or equivalently, determined by Hamming weights of the tuples) and additionally the template contains a relation which can be used to negate Boolean variables in both CSP templates. As the additional relation is binary and symmetric, the result concerns all the symmetric templates containing this particular relation. In this paper we remove the additional assumption and show that all symmetric Boolean templates exhibit a dichotomy.

Let us further compare the results. The algorithms required for the original and extended result are exactly the same: Gaussian elimination or linear programming relaxation depending on the polymorphisms of the template. The list of polymorphisms implying tractability differs slightly as we need to allow additional threshold functions (Boolean functions returning 0 if and only if the number of 1's is below a threshold). Unfortunately the condition which guarantees hardness in the original paper fails when the negating relation is absent. The new hardness condition and a more involved analysis of the minion of polymorphisms are required in the proof and constitute the main contribution of this paper.

The publication is organized as follows. The next section contains basic definitions commonly used in context of an algebraic approach to the CSP or the PCSP. Section 3 contains a list of polymorphisms that guarantee tractability, statement of the main theorem and a proof of the tractability case. In section 4 we introduce notation and nomenclature. Section 5 contains the algebraic condition implying hardness of PCSP and a proof of this implication. The main part of the reasoning behind the result is focused on showing that lack of polymorphisms from the tractability list implies, in our case, the condition for hardness. Section 6 contains an overview of this proof and a complete reasoning can be found in the full version of the paper.

2 Basic definitions

This section contains basic definitions and notions relevant to CSP and PCSP. A relation $R \subseteq A^n$ is an *n-ary relation* and the set A is its *universe*. A relation is *symmetric*, if for every permutation σ of $[n]$ (where $[n]$ is defined to be $\{1, \ldots, n\}$) if $(a_1, \ldots, a_n) \in R$ then also $(a_{\sigma(1)}, \ldots, a_{\sigma(n)}) \in R$. A relation $R^m \subseteq (A^m)^n$ is a *Cartesian power* of $R \subseteq A^n$ if $(a^1, \ldots, a^n) \in R^m$ if and only if $(a^1_i, \ldots, a^n_i) \in R$ for every i (i.e. R^m is defined from R coordinate-wise).

A *relational structure* $\mathbf{A}$ is a tuple $(A; R_1, \ldots R_n)$ where each R_i is a relation on A, and we call a relational structure *symmetric* if all its relations are. Two relational structures are *similar* if they have the same sequence of arities of their relations. E.g. a relational structure $(A; R_1, \ldots R_n)$ and it's m-th power $(A^m; (R_1)^m, \ldots (R_n)^m)$ are similar. For two similar structures say $\mathbf{A} = (A; R_1, \ldots, R_n)$ and $\mathbf{B} = (B; S_1, \ldots, S_n)$ a function $h : A \to B$ is *a homomorphism* if for every i and every tuple $(a_1, \ldots, a_m) \in R_i$ the tuple $(h(a_1), \ldots, h(a_m)) \in S_i$.

The Constraint Satisfaction Problem defined by a relational structure $\mathbf{B}$ (denoted by CSP($\mathbf{B}$)) is the following decision problem:

Input:	a relational structure $\mathbf{A}$ similar to $\mathbf{B}$
Question:	does there exists a homomorphism from $\mathbf{A}$ to $\mathbf{B}$?

The relational structure $\mathbf{B}$ is called a *template* of such a problem.

The Promise Constraint Satisfaction Problem is a promise problem defined by a pair of similar relational structures $(\mathbf{B}, \mathbf{C})$ such that there exists a homomorphism from $\mathbf{B}$ to $\mathbf{C}$. The PCSP$(\mathbf{B}, \mathbf{C})$ is:

Input:	a relational structure $\mathbf{A}$ similar to $\mathbf{B}$ and $\mathbf{C}$
Output YES:	if there exists a homomorphism from $\mathbf{A}$ to $\mathbf{B}$
Output NO:	if there is no homomorphism from $\mathbf{A}$ to $\mathbf{C}$.

Just like in the case of the CSP, the pair $(\mathbf{B}, \mathbf{C})$ is called a template. Clearly PCSP$(\mathbf{B}, \mathbf{B})$ is CSP$(\mathbf{B})$ and therefore the PCSP generalizes the CSP.

Both problems exhibit a Galois correspondence i.e. instead of studying the structure of the template one can choose to analyze the structure of template's polymorphisms [13, 7, 5, 8]. A *polymorphism* of a relational structure $\mathbf{B}$ is a homomorphism from a finite Cartesian power of $\mathbf{B}$ to $\mathbf{B}$. Similarly a polymorphism of a PCSP template $(\mathbf{B}, \mathbf{C})$ is a homomorphism from a finite Cartesian power of $\mathbf{B}$ to $\mathbf{C}$. We denote the set of all polymorphisms of $\mathbf{B}$ by Pol$(\mathbf{B})$, and the set of all polymorphisms of $(\mathbf{B}, \mathbf{C})$ by Pol$(\mathbf{B}, \mathbf{C})$.

For each relational structure $\mathbf{B}$ the set Pol$(\mathbf{B})$ is *clone* i.e. it contains projections and is closed under composition. Similarly for a pair $(\mathbf{B}, \mathbf{C})$ the set Pol$(\mathbf{B}, \mathbf{C})$ is a minion. *A minion* is a set of functions closed under taking *minors* i.e. creating functions by identifying variables, permuting variables and introducing dummy variables. If $f(x_1, \ldots, x_n)$ is a function and $f'(x) = f(x, \ldots, x)$ then $f'(x)$ is the unary minor of $f(x_1, \ldots, x_n)$ and $f''(x, y) = f(x, y, \ldots, y)$ is a binary minor of $f(x_1, \ldots, x_n)$.

In some cases, instead of considering a PCSP template $((A; R_1, \ldots, R_n), (B; S_1, \ldots, S_n))$ we work with an equivalent concept of *a language* i.e. a sequence of pairs $[R_1, S_1], \ldots, [R_n, S_n]$. We say that a pair $[S, T]$ is *compatible* with a minion $\mathcal{M}$, if every member of $\mathcal{M}$ maps an appropriate power of S to T (the exponent of the power is the arity of the operation).

A primitive positive formula (pp-formula) is a formula constructed using atomic formulas, conjunction and existential quantification. Such formulas play a special role in CSP and PCSP: if a relation R has a primitive positive definition in $\mathbf{B}$ then R is compatible with Pol$(\mathbf{B})$ and adding R to $\mathbf{B}$ does not change the computational complexity of the CSP$(\mathbf{B})$. Similarly, if a pair $[R, S]$ has a pp-definition in the language of $(\mathbf{A}, \mathbf{B})$ (pp-formula in $[R_i, S_i]$ defines such $[R, S]$ in the natural way) then $[R, S]$ is compatible with Pol$(\mathbf{A}, \mathbf{B})$ and adding it to the language/template does not change the complexity [5]. One more construction, called *strict relaxation*, plays an important role in the theory of PCSP: if $[R_i, S_i]$ is an element of the language $(\mathbf{B}, \mathbf{C})$ and $R \subseteq R_i$ while $S_i \subseteq S$ then $[R, S]$ is compatible with Pol$(\mathbf{A}, \mathbf{B})$ and adding it to the language/template does not change the complexity.

3 Main theorem and tractability

Focusing on the Boolean domain we present the main theorem of the paper and prove that the tractable cases are indeed solvable in P. In this part of the proof our paper does not deviate much from [5]; the polymorphisms which imply tractability are almost the same with an exception of the threshold case.

- A n-ary function is *a max* (*a min*) if it returns maximum (resp. minimum) of its arguments (in the natural order on $\{0, 1\}$).

- A function $f(x_1,\dots,x_n)$ is *an alternating threshold* if $n=2k+1$ and

$$f(x_1,\dots,x_k,x_{k+1},\dots,x_n)=\begin{cases}0 & \text{if } \sum_{i=1}^{k} x_i \geq \sum_{i=k+1}^{n} x_i,\\ 1 & \text{if } \sum_{i=1}^{k} x_i < \sum_{i=k+1}^{n} x_i,\end{cases}$$

- A function $f(x_1,\dots,x_n)$ is *an xor* if n is odd and $f(x_1,\dots,x_n)=x_1+\dots+x_n \bmod 2$.
- A function $f(x_1,\dots,x_n)$ is *a q-threshold* (where q is a rational between 0 and 1) if

$$f(x_1,\dots,x_n)=\begin{cases}0 & \text{if } \sum_{i=1}^{n} x_i < nq,\\ 1 & \text{if } \sum_{i=1}^{n} x_i > nq,\end{cases}$$

and nq is not an integer. Note that all the evaluations of the $f(x_1,\dots,x_n)$ are determined.

We denote the set of all max functions by MAX, all the min functions by MIN, all alternating thresholds by AT all xor by XOR and all q-thresholds by THR_q. For a set of functions F by $\overline{F}$ we denote $\{1-f(x_1,\dots,x_n) : f(x_1,\dots,x_n)\in F\}$. We are ready to state the main result of the paper.

▶ **Theorem 1.** *Let $(\mathbf{A},\mathbf{B})$ be a symmetric, Boolean PCSP language. If Pol$(\mathbf{A},\mathbf{B})$ contains a constant or includes at least one of the sets MAX, MIN, AT, XOR , THR_q (for some q), $\overline{\mathrm{MAX}}$, $\overline{\mathrm{MIN}}$, $\overline{\mathrm{AT}}$, $\overline{\mathrm{XOR}}$ or $\overline{\mathrm{THR}_q}$ (for some q) then* $\mathrm{PCSP}(\mathbf{A},\mathbf{B})$ *is tractable. Otherwise it is NP-complete.*

Comparing the statement of Theorem 1 and Theorem 2.1 of [5] we find two differences: the earlier paper additionally assumes that negated variables can appear in instances and it allows the authors to substitute "THR_q for some q" with $\mathrm{THR}_{1/2}$ in the list of conditions that force tractability.

In the remaining part of this section we will show the tractability case of Theorem 1. The reasoning differs very little from the one found in [5] and therefore we cover it quickly: If $\mathrm{Pol}(\mathbf{A},\mathbf{B})$ contains a constant function $\mathrm{PCSP}(\mathbf{A},\mathbf{B})$ is clearly tractable; if it includes MAX, MIN and XOR tractability follows from Lemma 3.1 of [5]. If $\mathrm{AT}\subseteq\mathrm{Pol}(\mathbf{A},\mathbf{B})$ then Claim 2 of Section 3.2 [5] implies tractability. Finally the case of THR_q is a minor generalization of the argument in Claim 1 of Section 3.2 in the same paper, or a special case of Theorem 5.2 in [4].

The remaining cases reduce, just like in [5], to the ones from the previous paragraph: let relational structure $\mathbf{B}'$ be obtained from $\mathbf{B}$ by exchanging the roles of 0 and 1 (that is, in every relation in $\mathbf{B}$, in every tuple of this relation and at every position in this tuple we change x to $1-x$). The YES instances of $\mathrm{PCSP}(\mathbf{A},\mathbf{B}')$ and $\mathrm{PCSP}(\mathbf{A},\mathbf{B})$ are trivially the same and so are the NO instances. If $\overline{\mathrm{MIN}}\subseteq\mathrm{Pol}(\mathbf{A},\mathbf{B})$ then $\mathrm{MIN}\subseteq\mathrm{Pol}(\mathbf{A},\mathbf{B}')$ and, by the cases already established, $\mathrm{PCSP}(\mathbf{A},\mathbf{B}')$ is tractable. Clearly $\mathrm{PCSP}(\mathbf{A},\mathbf{B})$ is tractable as well and all the remaining tractable cases can be dealt with the same way.

4 The notation for symmetric Boolean PCSPs

In order to show NP-hardness in the remaining case of Theorem 1, we require a few definitions which allow us to work with symmetric Boolean relations and Boolean function concisely.

Every symmetric relation $R\subseteq\{0,1\}^m$ is uniquely determined by the set $I\subseteq\{0,\dots,m\}$ consisting of the Hamming weights of its elements. This fact allows us to use R and I interchangeably. Let $(\mathbf{B},\mathbf{C})$ be a symmetric, Boolean PCSP template with language $[R_1,S_1],\dots,[R_n,S_n]$ where the arities of the relations are $a_1,\dots,a_n$. We will denote such a language by $[\,I_1 \mid J_1\,]_{a_1},\dots,[\,I_n \mid J_n\,]_{a_n}$ where I_i (J_i) is a set of Hamming weights of elements of R_i (S_i respectively). We will often use a flattened form of this notation: we will denote $[\{1\} \mid \{1,2\}]_2$ by $[1 \mid 1,2]_2$ and so on as well as $[n] = \{1,\dots,n\}$.

Focusing on compatibility; an operation $f(x_1,\dots,x_n)$ is compatible with $[0 \mid 0]_1$ if and only if $f(0,\dots,0) = 0$ and compatible with $[1 \mid 1]_1$ if and only if $f(1,\dots,1) = 1$. The pair $[1 \mid 1]_2$ defines negation in $\mathbf{A}$ and $\mathbf{B}$ and therefore the main result of [5] is a special case of Theorem 1; the additional assumption states that $[1 \mid 1]_2$ is in the language of PCSP.

We proceed to illustrate a number of pp-definitions and strict relaxations that appear repeatedly in the proofs. Using $[I \mid J]_n$ and $[0 \mid 0]_1$ one can define $[I \setminus \{n\} \mid J \setminus \{n\}]_{n-1}$ using the following pp-formula:

$$\exists x_1 \ [0 \mid 0]_1(x_1) \wedge [I \mid J]_n(x_1,\dots,x_n).$$

Similarly

$$\exists x_1 \ [1 \mid 1]_1(x_1) \wedge [I \mid J]_n(x_1,\dots,x_n)$$

defines $[I' \mid J']_{n-1}$ where $I' = \{i-1 : i \in I \text{ and } i \neq 0\}$ and $J' = \{j-1 : j \in J \text{ and } j \neq 0\}$. The strict relaxations we use are straightforward: take $[I \mid J]_n$ with $i \in I$ while $j \notin J$ then, for example, $[i \mid \{0,\dots,n\} \setminus \{j\}]_n$ is a strict relaxation of $[I \mid J]_n$.

In the proof of tractability for $(\mathbf{B},\mathbf{C})$ (at the end of Section 3) we swapped the role of 0 and 1 in $\mathbf{C}$. In the new notation we change $[I_1 \mid J_1]_{a_1},\dots,[I_n \mid J_n]_{a_n}$ to $[I_1 \mid J_1']_{a_1},\dots,[I_n \mid J_n']_{a_n}$ where $J_k' = \{a_k - j : j \in J_k\}$. In some of the proofs we reuse this construction, although we usually swap for both $\mathbf{B}$ and $\mathbf{C}$ at the same time.

We define notation for Boolean functions next. A Boolean function $f(x_1,\dots,x_n)$ is *idempotent* if $f(0,\dots,0) = 0$ and $f(1,\dots,1) = 1$. By the discussion above a minion is idempotent (i.e. contains idempotent functions only) if it is compatible with $[0 \mid 0]_1$ and $[1 \mid 1]_1$. Moreover the idempotent part of $\mathrm{Pol}(\mathbf{B},\mathbf{C})$ can be obtained by adding these pairs to the language.

For a Boolean function $f(x_1,\dots,x_n)$ and a set $U \subseteq [n]$ the value $f(U)$ is defined as $f(x_1,\dots,x_n)$ where $\{i : x_i = 1\} = U$. When n is clear from the context we can write $\overline{U}$ instead of $[n] \setminus U$. Let $f(x_1,\dots,x_n)$ be a Boolean function $U \subseteq [n]$ then U is

- a 1-SET if $f(U) = 1$,
- a 0-SET if $f(\overline{U}) = 0$,
- a 1-FIXING-SET (0-FIXING-SET) if every $V \supseteq U$ is a 1-SET (resp. 0-SET).

Moreover we say that a minion has *small fixing sets*, if there exists a constant N such that every function from the minion has a 1-FIXING-SET smaller than N, or every function from the minion has a 0-FIXING-SET smaller than N. Finally we say that a minion has *bounded antichains*, if there exist a constant M such that no function in the minion has M pairwise disjoint 1-SETs, and no function in the minion has M pairwise disjoint 0-SETs.

5 The hardness proof

In order to satisfy the assumptions of Lemma 5.1, we need some structural properties of the minion $\mathrm{Pol}(\mathbf{A},\mathbf{B})$. The following theorem collects these properties and is a cornerstone of our classification.

▶ **Theorem 2.** *Let $\mathbf{A},\mathbf{B}$ be a symmetric PCSP language such that Pol$(\mathbf{A},\mathbf{B})$ is idempotent. If Pol$(\mathbf{A},\mathbf{B})$ does not include MAX, MIN, AT, XOR and THR$_q$ (for any q), then Pol$(\mathbf{A},\mathbf{B})$ has small fixing sets and bounded antichains.*

The Brakensiek and Guruswami [5] version of Theorem 2 requires that $(\mathbf{A}, \mathbf{B})$ contains $[\,1 \mid 1\,]_2$ and concludes that there exists a constant M such that every member of $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ has a set of size at most M which is a 1-FIXING-SET and a 0-FIXING-SET at the same time. The following example illustrates that their condition fails in our case.

▶ **Example 3.** Consider PCSP defined by a language consisting of $[\,0 \mid 0\,]_1$, $[\,1 \mid 1\,]_1$, $[\,1 \mid 1, 2\,]_3$ and $[\,1 \mid 1, 2\,]_4$. It is easy to verify that it falls into the hardness case of Theorem 1. On the other hand for each odd n the function $f(x_1, \ldots, x_n)$ defined as maximum of x_1 and n-ary element of $\mathrm{THR}_{1/2}$ is compatible with all the relational pairs. These functions have no uniform bound on the size of minimal 0-FIXING-SETs.

In the reminder of this section we use Theorem 2 to finish the proof of Theorem 1. We begin by introducing the machinery developed in [8] (a direct proof is possible, but involves a bit more technical considerations). The paper [8] defines *minor identity* as a formal expression of the form

$$f(x_1, \ldots, x_n) \approx g(x_{\pi(1)}, \ldots, x_{\pi(m)})$$

where f and g are function symbols (of arity n and m, respectively), $x_1, \ldots, x_n$ are variables, and $\pi : [m] \to [n]$. A minor identity is satisfied in a minion $\mathcal{M}$ (of functions from A to B) if there exists an interpretation of the function symbols f and g in $\mathcal{M}$, say ζ, satisfying

$$\zeta(f)(a_1, \ldots, a_n) = \zeta(g)(a_{\pi(1)}, ..., a_{\pi(m)})$$

for all $a_1, \ldots, a_n \in A$.

A bipartite minor condition is a finite set of minor identities in which function symbols used on the right- and left-hand sides are disjoint. A minor condition is *satisfied in a minion*, if there exists an interpretation simultaneously satisfying all the identities. A minor condition is *trivial* if it is satisfied in every minion, in particular, in the minion consisting of all projections on a set A that contains at least two elements. Finally, still following [8], a bipartite minor condition Σ is ε-robust (for some $\varepsilon > 0$) if no ε-fraction of identities from Σ is trivial.

▶ **Lemma 5.1** (Corollary 5.8 from [8]). *If there exists an $\varepsilon > 0$ such that $Pol(\mathbf{A}, \mathbf{B})$ does not satisfy any ε-robust bipartite minor condition, then $PCSP(\mathbf{A}, \mathbf{B})$ is NP-hard.*

In order to apply Lemma 5.1 to $\mathrm{PCSP}(\mathbf{A}, \mathbf{B})$ we need to ensure that $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ does not satisfy any ε-robust bipartite minor condition. Our first step is to prove it in the idempotent case.

▶ **Proposition 5.2.** *Let $\mathcal{M}$ be an idempotent minion with small fixing sets, and bounded antichains. Then $\mathcal{M}$ does not satisfy any ε-robust bipartite minor condition.*

Proof. The proof follows the same pattern as the proofs of Propositions 5.10 and 5.12 in [8] so we will use the notation from those Propositions in this proof. All we need to do is to find $\varepsilon > 0$ and a mapping assigning to each member of $\mathcal{M}$ a probability distribution on its variables. The probability distribution needs to satisfy the following condition: if $f, g \in \mathcal{M}$ and $f(x_1, \ldots, x_n) \approx g(x_{\pi(1)}, \ldots, x_{\pi(m)})$ then

- choosing a variable from the LHS according to the distribution for f and
- choosing a variable from the RHS according to the distribution for g,

with probability greater than ε we will choose the same variable.

In order to find such ε and the mapping for $\mathcal{M}$ we assume without loss of generality that small fixing sets in $\mathcal{M}$ are 1-FIXING-SETs and their size as well as a size of an antichain is bounded by constant M. We choose $\varepsilon < 1/M^4$ and define the probability distribution as follows: fix $f \in \mathcal{M}$ and from the collection of 1-FIXING-SETs smaller than M choose a maximal subset of pairwise disjoint 1-FIXING-SETs. Let U_f be the set of numbers appearing in this subset and the probability distribution for f is the uniform distribution on U_f.

Take an identity as above; as $|U_f| \leq M^2$ and $|U_g| \leq M^2$ in order to prove the claim it suffices to show that $\pi(U_g) \cap U_f \neq \emptyset$. Let U be one of the 1-FIXING-SETs which defined U_g. The set $\pi(U)$ is a 1-FIXING-SET of f and its size is bounded by M. The maximality of the subset defining U_f implies that U_f and $\pi(U)$ intersect, which concludes the proof. ◀

We are now ready to finish the proof of Theorem 1 (modulo Theorem 2) following a reasoning similar to the one used in [5]. Let $(\mathbf{B}, \mathbf{C})$ be a PCSP language such that $\mathrm{Pol}(\mathbf{B}, \mathbf{C})$ doesn't contain constant functions and do not include any of MAX, MIN, AT, XOR, THR_q, $\overline{\mathrm{MAX}}$, $\overline{\mathrm{MIN}}$, $\overline{\mathrm{AT}}$, $\overline{\mathrm{XOR}}$, $\overline{\mathrm{THR}_q}$. Let $(\mathbf{B}_+, \mathbf{C}_+)$ be $(\mathbf{B}, \mathbf{C})$ with $[\,1 \mid 1\,]_1$ and $[\,0 \mid 0\,]_1$ added. By Theorem 2 and Proposition 5.2 $\mathrm{Pol}(\mathbf{B}_+, \mathbf{C}_+)$ does not satisfy any ε-robust minor condition (for some fixed ε). Note that $\mathrm{Pol}(\mathbf{B}_+, \mathbf{C}_+)$ consists of these elements of $\mathrm{Pol}(\mathbf{B}, \mathbf{C})$ which have identity as the unary minor. Thus $\mathrm{Pol}(\mathbf{B}, \mathbf{C}) \setminus \mathrm{Pol}(\mathbf{B}_+, \mathbf{C}_+)$ consists of elements of $\mathrm{Pol}(\mathbf{B}, \mathbf{C})$ which have $x \mapsto 1 - x$ as the unary minor.

Consider the set $\overline{\mathrm{Pol}(\mathbf{B}, \mathbf{C}) \setminus \mathrm{Pol}(\mathbf{B}_+, \mathbf{C}_+)}$. It is a minion and it is equal to $\mathrm{Pol}(\mathbf{B}_-, \mathbf{C}_-)$, where $(\mathbf{B}_-, \mathbf{C}_-)$ is obtained from $(\mathbf{B}, \mathbf{C})$ in two steps: first the roles of 0 and 1 are swapped in $\mathbf{C}$ (just like in the tractability proof) and then $[\,1 \mid 1\,]_1, [\,0 \mid 0\,]_1$ are added to the language. Applying Proposition 5.2 to $(\mathbf{B}_-, \mathbf{C}_-)$ we conclude that $\overline{\mathrm{Pol}(\mathbf{B}, \mathbf{C}) \setminus \mathrm{Pol}(\mathbf{B}_+, \mathbf{C}_+)}$ does not satisfy any ε-robust minor condition (for some ε). The same holds for $\mathrm{Pol}(\mathbf{B}, \mathbf{C}) \setminus \mathrm{Pol}(\mathbf{B}_+, \mathbf{C}_+)$ and therefore $\mathrm{Pol}(\mathbf{B}, \mathbf{C})$ is a disjoint union of two minions which, for some ε, do not satisfy any ε-robust minor conditions. It follows that $\mathrm{Pol}(\mathbf{B}, \mathbf{C})$ does not satisfy any ε-robust minor condition and by Lemma 5.1 the $\mathrm{PCSP}(\mathbf{B}, \mathbf{C})$ is NP-hard.

6 Proof overview

Our proof of Theorem 2 consists of the following four propositions.

▶ **Proposition 6.1.** *Let $(\mathbf{A}, \mathbf{B})$ be a symmetric language such that $\mathcal{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ is idempotent. If $\mathcal{M}$ does not include neither MAX nor MIN, then it is compatible with some relational pair $[\,a \mid 1, \ldots, a+1\,]_{a+1}$ and some relational pair $[\,1 \mid 0, \ldots, c\,]_{c+1}$.*

For the next proposition we need to specialize the notion of bounded antichains. We say that a minion has *bounded antichains of 1-SETs (0-SETs)* if there exists a uniform bound on the number of pairwise disjoint 1-SETs (0-SETs respectively) an element of the minion can have.

▶ **Proposition 6.2.** *Let $\mathcal{M}$ be a minion compatible with $[\,a \mid 1, \ldots, a+1\,]_{a+1}$ and $[\,1 \mid 0, \ldots, c\,]_{c+1}$. Then $\mathcal{M}$ has bounded antichains of 1-SETs if and only if $\mathcal{M}$ has bounded antichains of 0-SETs.*

▶ **Proposition 6.3.** *Let $(\mathbf{A}, \mathbf{B})$ be a symmetric language such that $\mathcal{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ is idempotent. If $\mathcal{M}$ is compatible with some $[\,a \mid 1, \ldots, a+1\,]_{a+1}$, some $[\,1 \mid 0, \ldots, c\,]_{c+1}$ and does not have bounded antichains then $\mathcal{M}$ includes XOR or AT.*

▶ **Proposition 6.4.** *Let $(\mathbf{A}, \mathbf{B})$ be a symmetric language such that $\mathcal{M} = \mathrm{Pol}(\mathbf{A}, \mathbf{B})$ is idempotent. If $\mathcal{M}$ has bounded antichains and does not include any of THR_q then it has small fixing sets.*

The structure of the proof is as follows: if $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ has MIN or MAX we are in a tractable case. Otherwise we split the reasoning in two cases: either $\mathrm{Pol}(\mathbf{A}, \mathbf{B})$ fails the bounded antichain condition and by Proposition 6.3 we are tractable due to AT or XOR, or we have bounded antichains and by Proposition 6.4 we are either tractable due to THR_q or have small fixing sets which implies hardness (by Proposition 5.2). Proposition 6.2 allows us to "flip" the template if necessary.

In this section, we prove Propositions 6.1 and 6.2. We also provide proof sketches of Propositions 6.3 and 6.4. Detailed proofs can be found in the full version of the paper.

Proof of Proposition 6.1. The proof splits into two parts:

- $\mathcal{M}$ does not have MIN then $\mathcal{M}$ is compatible with $[\,a \mid 1,\dots,a+1\,]_{a+1}$
- $\mathcal{M}$ does not have MAX then $\mathcal{M}$ is compatible with $[\,1 \mid 0,\dots,c\,]_{c+1}$

Proof of both cases is analogous, so we will only prove the first part. Let us assume that $\mathcal{M} = \mathrm{Pol}(\mathbf{A},\mathbf{B})$ and $\mathcal{M}$ does not have MIN. So there must be $[\,I \mid J\,]_n$ in the language of $(\mathbf{A},\mathbf{B})$ such that MIN is not compatible with it. This implies that there exists $b < a < n$ such that $a \in I$ and $b \notin J$. Now, using pp-definitions and strict relaxations from Section 4, we will show that $\mathcal{M}$ is compatible with $[\,a-b \mid 1,\dots,a-b+1\,]_{a-b+1}$:

- use strict relaxation of $[\,I \mid J\,]_n$ to obtain $[\,a \mid 0,\dots,b-1,b+1,\dots,n\,]_n$;
- from the last pair pp-define, using $[\,0 \mid 0\,]_1$, the pair $[\,a \mid 0,\dots,b-1,b+1,\dots,a+1\,]_{a+1}$,
- finally from the previous pair pp-define, this time using $[\,1 \mid 1\,]_1$, the required pair $[\,a-b \mid 1,\dots,a-b+1\,]_{a-b+1}$. ◀

The following lemma is used in the proof of Proposition 6.2.

▶ **Lemma 6.5.** *Let $\mathcal{M}$ be a minion. Then:*

- *if $\mathcal{M}$ is compatible with some $[\,a \mid 1,\dots,a+1\,]_{a+1}$, then for each f in $\mathcal{M}$ a union of a-many pairwise disjoint 0-SETs is a 1-SET.*
- *if $\mathcal{M}$ is compatible with some $[\,1 \mid 0,\dots,c\,]_{c+1}$, then for each f in $\mathcal{M}$ a union of c-many pairwise disjoint 1-SETs is a 0-SET.*

Proof. The proofs of the two cases are analogous, so we will only prove the second one. Let $U_1,\dots,U_c$ be disjoint 1-SETs of the n-ary function $f \in \mathcal{M}$ and $U = \bigcup_{i=1}^{c} U_i$. Since every coordinate i occurs in exactly one set of $U_1,\dots,U_c,\overline{U}$ and f is compatible with $[\,1 \mid 0,\dots,c\,]_{c+1}$, the tuple $(f(U_1),\dots,f(U_c),f(\overline{U}))$ cannot evaluate to $(1,\dots,1)$. Therefore $f(\overline{U}) = 0$ and U is a 0-SET. See Figure 1 for example.

1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0

Figure 1 Example of c disjoint 1-SETs creating a 0-SET with $c = 3$. The yellow column represents the result of an evaluation of function f on tuples represented by other columns. The columns are in $[\,1 \mid 0,1,2,3\,]_4$ and the grey cells are $U_1,\dots,U_c$ while the red cells are U. ◀

Proof of Proposition 6.2. By using Lemma 6.5 we conclude that:

- if f contains an antichain of 1-SETs of size n then it also contains an antichain of 0-SETs of size at least $\lfloor \frac{n}{c} \rfloor$
- if f contains an antichain of 0-SETs of size n then it also contains an antichain of 1-SETs of size at least $\lfloor \frac{n}{a} \rfloor$

so if one of the antichains of 0-SETs or 1-SETs is bounded then the other one has to be bounded as well. ◀

Proof sketch of Proposition 6.3. Since $\mathcal{M}$ has unbounded antichains, we can take a function from $\mathcal{M}$ with a arbitrarily large antichain of 1-SETs. By taking its minor, we obtain f satisfying

$$f(1,0,\ldots,0)=f(0,1,0,\ldots,0)=\cdots=f(0,\ldots,0,1,0)=1.$$

Notice that the last coordinate is exceptional, it does not have to form a 1-SET. By taking further minors of f we either get g, of arbitrarily large arity, that satisfies

$$g(1,0,\ldots,0)=g(0,1,0,\ldots,0)=\cdots=g(0,\ldots,0,1)=1,$$

or compatibility with AT (see the full version of the paper). We are left with the case when g's, of arbitrarily large arity, are in $\mathcal{M}$.

If $\mathcal{M}$ does not include AT, it is compatible (after possibly changing ones to zeros and zeros to ones) with $[\,1 \mid 0,\ldots,n-2,n\,]_n$ or $[\,0,d \mid 0,\ldots,n-1\,]_n$ for some $d<n$. We use these relational pairs for forcing further behavior of g, and finally obtain an xor of an arbitrarily large arity. This implies that XOR is a subset of $\mathcal{M}$. ◀

Proof sketch of Proposition 6.4. If a minion $\mathcal{M}$ has bounded antichains and does not have q-threshold for any q, we can find (skipping an easy case discussed in the full version of the paper) positive integers a,b,c,d such that $c/d<a/b<1$ such that $\mathcal{M}$ is compatible with relational pairs

$$[\,a \mid 0,\ldots,b-1\,]_b, \quad [\,c \mid 1,\ldots,d\,]_d. \tag{1}$$

Notice that the converse, i.e. that these relational pairs prevent threshold, is clear since (1) disallows any q-threshold such that $q<a/b$ and any q-threshold such that $q>c/d$. It can be shown that these relational pairs are the general obstacle to a threshold polymorphism. We prove the proposition by induction on $a+b+c+d$.

For the reminder of the proof to work we are forced to work with weaker assumptions – instead of $\mathcal{M}$ being compatible with (1) we assume that $\mathcal{M}$ is "almost compatible" with the relational pairs. Nevertheless, the "almost compatibility" notion is rather technical, and we ignore it in this sketch. For a formal proof, see the full version of the paper. Here, let us simply assume that $\mathcal{M}$ is compatible with (1).

It turns out that the only interesting case is $c/d<a/b<1/2$. All the other cases can be either resolved directly or reduced to this one. Now, consider a minimal (ordered by inclusion) 0-SET U and let f_U denote $|U|$-ary operation obtained from f by plugging zeros to every coordinate not contained in U. Since f is compatible with $[\,a \mid 0,\ldots,b-1\,]_b$ and U is a 0-SET, f_U is compatible with $[\,c \mid 1,\ldots,d-c\,]_{d-c}$. Every 1-SET in f_U is also a 1-SET in f, so f_U has bounded antichains of 1-SETs. (bounded across every $f\in\mathcal{M}$ and every U). Moreover, since U is minimal, the complement $\overline{U}$ of U is "almost" a 1-SET (every strict superset is). If $\overline{U}$ was a 1-SET, f_U would be compatible with $[\,a \mid 0,1,\ldots,b-a-1\,]_{b-a}$ since f is compatible with $[\,a \mid 0,1,\ldots,b-1\,]_b$. This is where the weaker notion of compatibility (the star-compatibility) is necessary in the full proof. However for the sake of simplicity, assume that f_U is compatible with $[\,a \mid 0,1,\ldots,b-a-1\,]_{b-a}$. Since f_U has bounded antichains of 1-SETs and it is compatible with relational pairs

$$[\,a \mid 0,1,\ldots,b-a-1\,]_{b-a}, \quad [\,c \mid 1,\ldots,d-c\,]_{d-c}$$

where $c/(d-c)<a/(b-a)$, it has also bounded antichains of 0-SETs. Therefore, we can apply the induction hypothesis and obtain a small (bounded across every $f\in\mathcal{M}$ and every U) 1-FIXING-SET or 0-FIXING-SET V in f_U. For our purposes, we don't need to know

that the set is fixing, it suffices that it is a 0-SET or a 1-SET. Let $\mathcal{L}_f$ denote the set of all possible sets V above across all minimal 0-SETs U. From the induction hypothesis, we also get that either every $V \in \mathcal{L}_f$ is a 1-SET in the appropriate f_U, or every $V \in \mathcal{L}_f$ is a 0-SET in the appropriate f_U.

▷ Claim 4. The size of pairiwise disjoint subsystems of $\mathcal{L}_f$ is bounded by a number independent of the chosen $f \in \mathcal{M}$.

If every $V \in \mathcal{L}_f$ is a 1-SET in the appropriate f_U, then V is a 1-SET in f and the claim follows from $\mathcal{M}$ having bounded antichains. Let us prove the claim if every $V \in \mathcal{L}_f$ is a 0-SET in the appropriate f_U. Consider c disjoint elements $V_1, \ldots, V_c \in \mathcal{L}$, and let $U_1, \ldots, U_c$ be the appropriate minimal 0-SETs. Thus also every $\overline{U_i} \cup V_i$ is a 0-SET. Since

$$U_1, U_2, \ldots, U_c, \overline{U_1} \cup V_1, \overline{U_2} \cup V_2, \ldots, \overline{U_c} \cup V_c$$

are 0-SETs, $V_1 \cup \ldots \cup V_c$ is a 1-SET by compatibility with $[\, c \mid 1, 2, \ldots 2c+1\,]_{2c+1}$. Let M be the bound on antichains of 1-SETs in $\mathcal{M}$, the size of antichains in $\mathcal{L}$ is bounded by cM.

Finally, we use the claim to find a small 1-FIXING-SET in f. Consider any maximal sequence $V_1, \ldots, V_n \in \mathcal{L}$ of disjoint sets and let

$$W = V_1 \cup V_2 \cup \ldots \cup V_n,$$

Every 0-SET contains a minimal 0-SET, every minimal 0-SET contains some $V \in \mathcal{L}$ and every $V \in \mathcal{L}$ intersects W. Therefore every 0-SET intersects W, so W is the desired 1-FIXING-SET. ◀

References

1. P. Austrin, J. Håstad, and V. Guruswami. (2 + epsilon)-Sat Is NP-Hard. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 1–10, October 2014. doi:10.1109/FOCS.2014.9.
2. Libor Barto and Marcin Kozik. Robust Satisfiability of Constraint Satisfaction Problems. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 931–940, New York, NY, USA, 2012. ACM. doi:10.1145/2213977.2214061.
3. Joshua Brakensiek and Venkatesan Guruswami. New Hardness Results for Graph and Hypergraph Colorings. In *Proceedings of the 31st Conference on Computational Complexity*, CCC '16, pages 14:1–14:27, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2016.14.
4. Joshua Brakensiek and Venkatesan Guruswami. An Algorithmic Blend of LPs and Ring Equations for Promise CSPs. *CoRR*, abs/1807.05194, 2018. arXiv:1807.05194.
5. Joshua Brakensiek and Venkatesan Guruswami. *Promise Constraint Satisfaction: Structure Theory and a Symmetric Boolean Dichotomy*, pages 1782–1801. SIAM, 2018. doi:10.1137/1.9781611975031.117.
6. A. A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, volume 00, pages 319–330, October 2017. doi:10.1109/FOCS.2017.37.
7. Andrei A. Bulatov, Andrei A. Krokhin, and Peter Jeavons. Constraint satisfaction problems and finite algebras. In *Automata, languages and programming (Geneva, 2000)*, volume 1853 of *Lecture Notes in Comput. Sci.*, pages 272–282. Springer, Berlin, 2000.
8. Jakub Bulín, Andrei A. Krokhin, and Jakub Oprsal. Algebraic approach to promise constraint satisfaction. *CoRR*, abs/1811.00970, 2018. arXiv:1811.00970.
9. Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional Hardness for Approximate Coloring. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 344–353, New York, NY, USA, 2006. ACM. doi:10.1145/1132516.1132567.

10 Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM Journal on Computing*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

11 Miron Ficak, Marcin Kozik, Miroslav Olsak, and Szymon Stankiewicz. Dichotomy for symmetric Boolean PCSPs, 2019. arXiv:1904.12424.

12 Sangxia Huang. Improved Hardness of Approximating Chromatic Number. In Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 233–243, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

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

14 Vladimir Kolmogorov, Andrei Krokhin, and Michal Rolinek. The Complexity of General-Valued CSPs. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 1246–1258, Washington, DC, USA, 2015. IEEE Computer Society. doi:10.1109/FOCS.2015.80.

15 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.

16 D. Zhuk. A Proof of CSP Dichotomy Conjecture. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, volume 00, pages 331–342, October 2017. doi:10.1109/FOCS.2017.38.